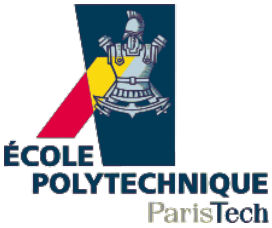
Optimisation et jeux appliqués à l'analyse statique de programmes par interprétation abstraite

Assalé Adje



Thèse présentée pour obtenir le grade de

DOCTEUR DE L'ÉCOLE POLYTECHNIQUE

Spécialité : Mathématiques Appliquées

par

Assalé Adje

Optimisation et jeux appliqués à l'analyse statique de programmes par interprétation abstraite

Soutenue le 29 Avril 2011 devant le jury composé de :

Sylvain Sorin	Université Pierre et Marie Curie, Paris	*Président du jury*
Eric Feron	Georgia Institute of Technology, Atlanta	*Rapporteur*
Michel Kieffer	École Supérieure d'Électricité, Gif-sur-Yvette	*Rapporteur*
Bertrand Jeannet	INRIA Rhône-Alpes, Montbonnot	*Examinateur*
Jérôme Leroux	LABRI-CNRS, Talence	*Examinateur*
Stéphane Gaubert	INRIA Saclay	*Directeur de thèse*
Eric Goubault	École Polytechnique, Palaiseau	*Co-Directeur de thèse*

Remerciements

Maintenant place à la traditionnelle page de remerciements pour tous ceux qui m'ont aidé, poussé, encouragé...

Tout d'abord, mes directeurs de thèse : Stéphane Gaubert et Eric Goubault qui m'ont beaucoup appris scientifiquement et humainement. Ils m'ont dirigé vers un domaine très riche et stimulant qui m'était inconnu à la sortie de mon cursus universitaire. Malgré leur charge de travail très conséquente, ils ont toujours été disponibles pour m'écouter et m'orienter vers de bonnes idées.

Je remercie sincèrement à Eric Feron et Michel Kieffer d'avoir rapporté cette thèse dont mon côté "Bourbakiste" n'a pas dû rendre la lecture facile. Leurs commentaires très constructifs ont été très bénéfiques pour améliorer ce manuscrit.

Je remercie également les membres du jury en commençant par Bertrand Jeannet et Jérôme Leroux. Je tiens à remercier Sylvain Sorin pour l'avoir présidé. Il m'a aussi initié à l'optimisation et à la théorie des jeux dès ma première année de master, deux pré-requis sans lesquels je n'aurais pas pu effectuer cette thèse. Je tiens également à remercier Jérôme Bolte qui a été mon chargé de TD en analyse convexe cette même année qui m'a énormément appris.

Je remercie tous les membres de l'équipe MeASI (ou ex-membres de l'équipe) pour les 36 premiers mois de thèse (et plus en comptant le stage), plus précisément : Alexandre, Emmanuel, Franck, Karim, Khalil, Michel, Olivier, Samuel, Sylvie et Xavier, pour leur chaleureux accueil et la disponibilité de chacun pour mes (trop) nombreuses questions sur la programmation, sur l'utilisation de Linux ainsi que de LaTEX.

Je remercie tous les membres du CMAP (aussi bien la partie administrative que la partie recherche) qui m'ont accueilli pour les quatre derniers mois de ma thèse et avec qui, malgré la courte durée de ma visite, j'ai eu d'excellents rapports.

Je remercie mes parents et mon frère pour leur soutien durant toutes mes années d'études. Je remercie également mes amis avec qui j'ai pu décompresser quand j'en avais besoin.

Enfin, une dernière pensée pour ma *cane canne* Blandine qui a été mon plus grand soutien durant ces deux dernières années de thèse malgré mes « t'inquiète pas, j'ai bientôt fini ! ».

Résumé

L'interprétation abstraite est une méthode générale qui permet de déterminer de manière automatique des invariants de programmes. Cette méthode conduit à résoudre un problème de point fixe non linéaire de grande taille mais qui possède des propriétés de monotonie. Ainsi, déterminer des bornes sur les valeurs prises par une variable au cours de l'exécution d'un programme, est un problème de point fixe équivalent à un problème de jeu à deux joueurs, à somme nulle et avec options d'arrêt. Cette dernière observation explique la mise en oeuvre d'algorithmes d'itérations sur les politiques.

Dans un premier temps, nous avons généralisé les domaines numériques polyédriques par un domaine numérique abstrait permettant de représenter des invariants non-linéaires. Nous avons défini une fonction sémantique abstraite sur ce domaine à partir d'une correspondance de Galois. Cependant, l'évaluation de celle-ci est aussi difficile qu'un problème d'optimisation globale non-convexe. Cela nous a amené à définir une fonction sémantique relâchée, construite à partir de la théorie de la dualité, qui sur-approxime de la fonction sémantique abstraite. La théorie de la dualité a également motivé une construction d'une itération sur les politiques dynamique pour calculer des invariants numériques. En pratique pour des programmes écrits en arithmétique affine, nous avons combiné la relaxation de Shor et l'information des fonctions de Lyapunov quadratique pour évaluer la fonction sémantique relâchée et ainsi générer des invariants numériques sous forme d'ellipsoïdes tronquées.

Le deuxième travail concerne l'itération sur les politiques et le calcul du plus petit point fixe qui fournit l'invariant le plus précis. Nous avons raffiné l'itération sur les politiques afin de produire le plus petit point fixe dans le cas des jeux stochastiques. Ce raffinement repose sur des techniques de théorie de Perron-Frobenius non-linéaire. En effet, la fonction sémantique abstraite sur les intervalles peut être vue comme un opérateur de Shapley en information parfaite : elle est semidifférentiable. L'approche conjointe de la semidifférentielle et des rayons spectraux non linéaires nous a permis, dans le cas des contractions au sens large de caractériser le plus petit point fixe. Cette approche mène à un critère d'arrêt pour l'itération sur politique dans le cas des fonctions affines par morceaux contractantes au sens large. Quand le point fixe est non minimal, le problème consiste à exhiber un point fixe négatif non nul de la semidifférentielle. Ce vecteur conduit à une nouvelle politique qui fournit un point fixe strictement plus petit que le point fixe courant. Cette approche a été appliquée à quelques exemples de jeux stochastiques à paiements positifs et de vérification de programmes.

Mots-clés : *Interprétation abstraite ; itération sur les politiques ; optimisation convexe ; convexité abstraite ; Perron-Frobenius non-linéaire ; théorie des jeux*

Abstract

The abstract interpretation is a general method to compute automatically program invariants. This method leads to solve a non-linear fixed point equation involving a monotone function. Determining numerical invariants in order, for instance, to give bounds on the values taken by the variables of a program, turns out to be equivalent to solving a two-player zero-sum game with stopping options. This observation allows one to transport algorithms from game theory, like policy iteration, to abstract interpretation.

The first contribution of this thesis is the generalisation of these abstract polyhedraic numerical domains. We construct a general abstract numerical domain which encompasses all the classical ones. We define an abstract semantic function in terms of a Galois connection. However, evaluating the abstract semantic function is as hard as solving a non-convex global optimization problem. Hence, we define a second semantic function called relaxed semantics constructed from duality theory, which provides a safe overapproximation of the abstract semantic function. The duality theory also motivates the construction of a dynamical policy iteration algorithm to compute numerical invariants. In practice for programs written in affine arithmetic, we combine Shor's relaxation scheme and Lyapunov functions to evaluate the relaxed semantic function and so generate numerical invariants which are the form of truncated ellipsoids.

The second contribution concerns policy iteration and computation of the smallest fixed point problem which provides the more precise invariant. We refined the policy iteration algorithm in order to compute the smallest fixed point, in the case of stochastic games. The refinement is based on non-linear Perron-Frobenius theory. However, since the abstract semantic function in the case of interval domain can be interpreted as a Shapley operator in perfect information, we use a weaker notion of differentiability : it is semidifferentiable. The approach by the semiderivatives combined by non-linear spectral radius allows us to characterise the fixed points in the non-expansive case. In the case of non-expansive and piecewise affine (Shapley) operators, the characterisation leads to a termination criteria for policy iteration. When the fixed point found by policy iteration is not minimal, the problem is reduced to finding a non-positive fixed point for a semiderivative map. This vector provides a descent direction which leads to a new policy and then to a strictly smaller fixed point. This approach has also been applied to typical examples arising from game or program verification problems.

Keywords : *Abstract interpretation ; policy iteration ; convex optimisation ; abstract convexity ; non-linear Perron-Frobenius ; game theory*

TABLE DES MATIÈRES

quelques notations

$\emptyset$	Ensemble vide
$\wp(E)$	Ensemble des parties d'un ensemble E
A^c	Complémentaire d'un ensemble
$\subseteq$	Inclusion ensembliste classique
$\cup$	Union ensembliste classique
$\cap$	Intersection ensembliste classique
$\mathrm{fix}(f)$	Ensemble des points fixes d'une application f
$\overline{\mathbb{Z}}$	$\mathbb{Z} \cup \{+\infty\} \cup \{-\infty\}$
$\overline{\mathbb{R}}$	$\mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$
$\mathrm{lfp}(f)$	Plus petit point fixe d'une application f
$\mathrm{gfp}(f)$	Plus grand point fixe d'une application f
$\mathbb{R}^X$	Ensemble des fonctions de X dans $\mathbb{R}$
$\overline{\mathbb{R}}^X$	Ensemble des fonctions de X dans $\overline{\mathbb{R}}$
$\mathbb{R}_+^X$	Ensemble des fonctions de X dans $\mathbb{R}_+$
$\overline{\mathbb{R}}_+^X$	Ensemble des fonctions de X dans $\mathbb{R}_+ \cup \{+\infty\}$
$\mathbb{R}_\infty^X$	Ensemble des fonctions de X dans $\mathbb{R} \cup \{+\infty\}$
dom	Domaine d'une fonction
Id	Fonction identité
$\mathbf{e}_x$	Fonctions d'évaluations au point x
$x \cdot y$	Produit scalaire usuel de $\mathbb{R}^d$ i.e $\sum_i^d x_i y_i$
A^t	Transposée d'une matrice
b^t	Transposé d'un vecteur i.e vecteur ligne
$\mathrm{Val}(P)$	Valeur d'un problème d'optimisation
$\mathrm{Sol}(P)$	Ensemble des solutions d'un problème d'optimisation
$\mathcal{C}(E, F)$	Ensemble d'applications continues de E dans F (E, F espaces topologiques)
$\mathrm{card}(C)$	Cardinal d'un ensemble C
$g \equiv \alpha$	$g(x) = \alpha,\ \forall\, x$
$B(a, r)$	Boule de centre a et de rayon r
$[a, b]$	Intervalle fermé de borne inférieure a et de borne supérieure b

CHAPITRE 1

INTRODUCTION

En 1962, Philippe Dreyfus, à l'époque ingénieur chez Bull, inventa le mot *informatique* pour traduire en français le terme anglais *computer science*. Il construit ce mot en contractant les mots *infor*mation et auto*matique*[1]. Ainsi, peut-on définir la science informatique comme l'automatisation du traitement de l'information. Pour nous, la notion importante ici est l'automatisation qui signifie sans intervention humaine. L'automatisation est donc à la base des programmes informatiques : ces programmes exécutent de manière automatique des calculs numériques ou logiques voire simulent des expériences. Les programmes consistent en une suite d'opérations écrites dans un certain langage de programmation qu'un humain veut exécuter. La complexité des programmes actuels et leur longueur font que les programmes sont aussi difficiles à vérifier qu'à écrire, pour illustrer nos propos : le logiciel embarqué d'une automobile récente peut contenir plus de 35 millions de lignes de codes[2]. Une relecture du code ne peut donc suffire pour garantir la correction d'un programme. Avant d'utiliser de tels logiciels, il faut pourtant vérifier leur bon fonctionnement, vérifier que le programme répond bien aux spécifications attendues, vérifier que le programme ne contient pas d'erreurs.

1.1 Vérification de programmes

1.1.1 Quelques dysfonctionnements catastrophiques

Nous soulignons dans cette sous-section l'importance des méthodes de vérifications avant l'utilisation d'un programme. Nous citons trois dysfonctionnements désastreux bien connus causés par des erreurs informatiques. On peut citer l'explosion de la fusée Ariane 5 [Boa96] en 1996, 37 secondes après son décollage. Ce dysfonctionnement était causé par la reprise d'un logiciel conçu pour Ariane 4 qui n'était plus valide sur la nouvelle fusée Ariane 5. Un dysfonctionnement plus dramatique humainement : entre 1985 et 1987, un logiciel de contrôle de la machine *Therac-25* de radiothérapie a provoqué l'administration de doses massives de radiation. Trois patients furent grièvement blessés et trois autres décédèrent [LT93]. Enfin, citons le dysfonctionnement de l'antimissile Patriot [Ske92, BBO92] qui aurait dû intercepter

1. Michel Volle relate cette anecdote dans [Vol06, p 201-202]
2. source :http://www.mathworks.com/company/pressroom/newsletter/sept06/body_electronics.html

le missile Scud lors de l'opération *Desert Storm*. Une erreur dans le logiciel de contrôle de l'antimissile empêcha l'interception et causa la mort de 28 soldats américains.

Ces dysfonctionnements exemplaires font partie d'une longue liste d'incidents liés à l'informatique. Ils expliquent la longue et complexe activité de tests et de vérifications en amont de l'utilisation et la commercialisation d'un logiciel.

1.1.2 Insuffisance du test

D'après Myers [MBT+04], tester un programme consiste à exécuter le programme afin d'y trouver des anomalies ou des défauts. Ainsi tester un programme semble la méthode la plus naturelle pour vérifier la correction d'un programme. La finalité du test est de répondre aux deux questions suivantes : « le programme réalise-t-il les spécifications attendues ? » et « le programme s'exécute-t-il normalement ? » ; plus précisément le test dit de *vérification* est censé répondre à la première question tandis que le test dit de *validation* est censé répondre à la seconde question. De plus, on peut différentier plusieurs types de tests à différents niveaux du *cycle de développement* : le test *unitaire* qui vérifie chaque module du programme individuellement, le test *d'intégration* qui vérifie le bon fonctionnement des compositions des modules et enfin le test de *conformité* qui vérifie la conformité par rapport à la spécification prédéfinie. Pour un test unitaire, des valeurs d'entrées sont tirées aléatoirement, les modules du logiciel sont ensuite exécutés avec pour entrée ces valeurs aléatoires. Même répétés en grand nombre, ces tests aléatoires assurent le bon fonctionnement des modules avec une probabilité $p < 1$ ce qui n'est pas suffisant pour des systèmes critiques où le moindre évènement imprévu peut être catastrophique. Par ailleurs, le nombre de valeurs d'entrées peut être élevé, par exemple un logiciel avec 5 entrées analogiques sur 8 bits possède 2^{40} valeurs d'entrées possibles. Le nombre des valeurs d'entrée à tester rend donc impossible un test exhaustif et l'automatisation des tests difficile. On peut résumer ce paragraphe sur le test par cette phrase due à Dijkstra dans [Dij70] : « Tester un programme peut démontrer la présence de bugs mais jamais leur absence ». Par conséquent, il faut utiliser une approche plus globale pour prouver que le programme satisfait bien les spécifications demandées et chercher toutes les erreurs contenues dans un programme.

1.1.3 Analyse statique de programmes

L'approche globale passe par des méthodes de vérification formelles comme l'analyse statique de programme. L'analyse est dite statique car l'extraction des propriétés se fait sans exécuter le programme. L'analyse statique de programme se divise en trois grandes catégories :

- Les techniques de preuve à la Hoare [Hoa69, Win93] : prouver une propriété sur un programme en traduisant le programme en une suite de formules logiques. Ces formules logiques sont ensuite prouvées à l'aide d'assistants de preuve comme Coq [3] ou Isabelle [4].
- Le *model-checking* introduit parallèlement par Clarke et Emerson [CE82] et Queille et Sifakis [QS82] prouve des propriétés sur un programme en regardant tous les états atteignables durant l'exécution de celui-ci. Le problème global est réduit à vérifier la propriété à démontrer sur cet ensemble fini d'états.
- L'*interprétation abstraite* introduite par Patrick et Radhia Cousot [CC77], une méthode qui propose de prouver les propriétés en considérant une abstraction du programme par

3. http ://inria.coq.fr
4. http ://isabelle.in.tum.de

une fonction construite à partir d'une sémantique. Nous développerons plus tard les principes de base de l'interprétation abstraite.

L'analyse statique de programmes permet de prouver l'absence de fautes dans les programmes. L'analyse statique possède d'autres applications pour la compilation notamment : l'optimisation de codes, la parallélisation à la compilation... La méthode consiste à traduire un programme en un système d'équations qui peut être résolu automatiquement. Il n'est, cependant, pas possible de construire un analyseur statique général capable de prouver ou non qu'un programme donné satisfait une certaine propriété quelque soit le programme et la propriété. Ceci est une conséquence du théorème d'indécidabilité de Rice [Ric53]. L'interprétation abstraite remédie à cette indécidabalité au prix d'une approximation. La méthode consiste à écrire un système d'équations dont toute solution détermine une sur-approximation de l'ensemble des comportements possibles du programme. Il peut arriver que cette solution ne soit pas assez précise et ne permette pas de conclure que le programme satisfait une propriété ou non. Le point important de l'interprétation abstraite est le caractère automatique du calcul d'une solution. Les mondes industriel et académique ont développé plusieurs analyseurs statiques basés sur l'interprétation abstraite ; citons en particulier : **Astrée**[5] développé à l'E.N.S Ulm détecte les erreurs à l'exécution ; **aiT**[6] de la société Absint calcule le pire temps d'exécution ; **Fluctuat**[7] développé au C.E.A permet d'analyser les erreurs entre les valeurs réelles d'un programme numérique et son implémentation machine sur les flottants ; **Penjili**[8] développé à E.A.D.S prouve automatiquement les failles dues notamment aux tableaux non initialisés, dépassement d'indices... ; **Polyspace Verifier**[9] de The MathWorks détecte les erreurs à l'exécution.

1.2 Problèmes de point fixe en interprétation abstraite

1.2.1 Domaines numériques abstraits

Domaines relationnels et formes linéaires

Une méthode intuitive pour trouver des bornes sur les valeurs des variables d'un programme en chaque point de contrôle est d'utiliser les intervalles. On cherche ainsi à déterminer en chaque point de contrôle de programme, pour chaque variable un intervalle contenant toutes les valeurs que cette variable pourra prendre durant l'exécution. Le domaine des intervalles a été le premier domaine considéré par Cousot et Cousot [CC77] pour calculer des invariants numériques. L'inconvénient d'une analyse statique par intervalles est la perte de l'information sur les relations entre les variables. L'introduction du domaine des polyèdres convexes [CH78] a été une première approche pour garder les relations linéaires entre les variables. On considère ainsi toutes les relations linéaires (ou plutôt affines) possibles. Lorsque l'on fait tendre le nombre de relations vers l'infini, on parvient ainsi à décrire un ensemble convexe fermé quelconque, car tout convexe fermé est intersection des demi-espaces fermés qui le contiennent. L'analyse statique sur le domaine des polyèdres utilise à la fois la représentation des polyèdres sous forme de contraintes et celle sous forme de générateurs et rayons.

5. http://www.astree.ens.fr
6. http://www.absint.com/ait/
7. http://www-list.cea.fr/labos/fr/LSL/fluctuat/index.html
8. http://penjili.org/penjili.html
9. http://www.mathworks.com/products/polyspace/

Or, le passage du polyèdre sous la forme de contraintes à la forme en générateurs et rayons est exponentiel puisqu'il faut 2^n générateurs pour un système à n contraintes dans $\mathbb{R}^n$. En bornant le nombre de relations linéaires au cours de l'analyse du programme, on peut résoudre le problème d'explosion combinatoire. Une approche consiste à travailler dans une sous-classe de polyèdres. Parmi celle-ci, une sous-classe remarquable est celle des zones introduites par Miné [Min01a] : une zone est un ensemble formé de vecteurs $x \in \mathbb{R}^d$, d étant le nombres de variables, définis par des contraintes de type potentiel, $x_i - x_j \leq C_{ij}$ où C_{ij} est une matrice donnée, à coefficients dans $\mathbb{R} \cup \{+\infty\}$. Les zones permettent ainsi de borner les différences entre les variables du programme. En incorporant une variable x_0 supplémentaire identiquement nulle, on exprime en particulier dans ce formalisme les bornes portant uniquement sur une variable. Autrement dit, les zones sont plus expressives que les intervalles. Par la suite, Miné [Min01b] a considéré les formes linéaires du type $\pm x_i \pm x_j$, Miné a appelé ce domaine, le domaines des octogones. Le principe du calcul des invariants dans les octogones est le même que celui des zones. Sankaranarayanan, Sipma et Manna [SSM05] ont proposé une généralisation du travail de Miné en fixant *avant* l'analyse, une famille finie $(p_k)_{k\in\{0,\ldots,K\}}$ de formes linéaires qui définissent les contraintes d'un polyèdre. L'invariant cherché est par conséquent de la forme $p_k(x) \leq C_k$ et les inconnues sont les bornes C_k. En prenant $d(d+1)$ formes linéaires de la forme $p(x) = x_i - x_j$, avec $0 \leq i, j \leq d$ et $i \neq j$, on retrouve évidemment le domaine des zones.

Fonctions de Lyapunov et invariants numériques

Une autre approche qui sert de source d'inspiration à ce travail est issue de la théorie du contrôle. Un programme informatique peut être en effet considéré comme un système dynamique à temps discret. En théorie du contrôle, l'existence d'une fonction de Lyapunov suffit à montrer la stabilité du système. Celle-ci fournit en outre une collection infinie d'ensembles invariants par la dynamique (les ensembles de sous-niveau de la fonction de Lyapunov), qui sont dans une certaine mesure analogues aux ensembles invariants construits en interprétation abstraite. Cependant, peu d'analyses de programmes utilisent des fonctions de Lyapunov ou d'autres critères de stabilité pour générer les invariants numériques. Feret [Fer04] a proposé de calculer une fonction de Lyapunov par interprétation abstraite pour les systèmes linéaires d'ordre 2. Feron [FA08a, FA08b] propose de calculer séquentiellement des ellipsoïdes à chaque point de contrôle d'un programme. On peut citer aussi Martel [Mar02] qui a utilisé la théorie de stabilité de Lyapunov en analyse statique. Martel a proposé une analyse statique pour prouver la stabilité des calculs sur les flottants dans une boucle grâce aux exposants de Lyapunov.

1.2.2 Itération de Kleene

En interprétation abstraite, l'algorithme classique utilisé pour calculer le plus petit point fixe d'une application croissante sur un treillis complet est l'itération de Kleene [Bou93, Mar98]. La vitesse de convergence de l'itération de Kleene dépend des propriétés de convergence (ou de divergence) du système dynamique sous-jacent. L'itération peut donc être très lente. La convergence en temps fini est assurée par des techniques d'accélération, introduites par Cousot [CC92b], qui ne garantissent plus la précision de l'invariant calculé.

Lorsque les opérations du programme sont uniquement affines (pas de produits ou divisions de variables), les coordonnées de la fonction sémantique abstraite, dans le cas des domaines

numériques abstraits des intervalles, zones, octogones ou gabarits de Sankaranarayanan sont des infima ou des suprema de fonctions affines (éventuellement juste des fonctions affines). Ce dernier constat montre une certaine analogie avec l'opérateur de Shapley des jeux stochastiques à information parfaite. L'itération de Kleene peut être ainsi considérée comme l'analogue de l'itération sur les valeurs de la théorie des jeux. En théorie des jeux, une alternative à l'itération sur les valeurs est proposée par l'itération sur les politiques. Initialement introduite par Howard [How60] pour les problèmes de décision markoviens à un joueur, elle a été généralisée par Hoffman et Karp [HK66] au cas des jeux satisfaisant des hypothèses d'ergodicité, et généralisée ensuite au cas de jeux combinatoires pour lesquelles de telles hypothèses ne sont plus satisfaites (jeux à paiement moyen, jeu de parité) [ZP96, CTGG99, VJ00]. Elle a été adaptée par Costan et al dans [CGG+05] à l'analyse statique.

Lorsque l'espace des politiques est fini, l'itération sur les politiques converge toujours vers un point fixe. En particulier, lorsque le point fixe est unique, on obtient a fortiori le plus petit point fixe, correspondant à l'invariant de précision maximale. Les diverses variantes de l'itération sur les politiques que l'on rencontre dans la littérature ont toutes la propriétés de produire une suite de points fixes de fonctions associées aux politiques qui est strictement décroissante pour un ordre convenable. On ne parcourt donc jamais deux fois la même politique et l'algorithme termine donc en un temps au plus égal au nombre de politiques, qui est en général exponentiel en la taille de l'instance. Un contre exemple récent de O. Friedmann [Fri09] montre que l'itération sur les politiques peut prendre un temps exponentiel dans le pire des cas. Cependant, ce contre exemple à une structure très particulière, et sur la plupart des instances (aussi bien cas réels que problèmes aléatoires), l'itération sur les politiques pour les jeux converge très rapidement [DG06, CTCG+98].

Il y a cependant une différence technique importante entre l'itération sur les politiques de la théorie des jeux et celle de l'analyse statique. Dans le premier cas, on se place typiquement dans une situation où le point fixe est unique, l'unicité étant due à la présence d'un taux d'escompte, d'une possibilité de mort du processus sous-jacent (présence d'états absorbants), et, dans des cas de non unicité, on rajoute en général des conditions qui permettent de sélectionner un unique point fixe bien défini (par exemple : sélection du point fixe "Blackwell optimal", obtenu en faisant tendre le taux d'escompte vers 0). En interprétation abstraite, on ne peut faire de même, car la fonctionnelle d'analyse statique possède en général plusieurs point fixes, et l'artifice du taux d'escompte évanescent ne peut être employé. Néanmoins la décroissance des termes de la suite générée par l'itération sur les politiques vers un point fixe implique que tout terme fournit un invariant, parfois peu précis mais valide.

Citons aussi l'approche de Gawlitza et Seidl [GS07a, GS07b] qui définissent une itération sur les « politiques » ascendantes. Gawlitza et Seidl initialisent leur algorithme par rapport à la valeur $\bot$ (le plus petit élément du domaine numérique abstrait) et itèrent en sélectionnant à chaque étape, la « politique » optimale. Le critère d'arrêt de l'algorithme demeure l'atteinte d'un point fixe. Cette approche du plus petit point fixe par le bas entraîne un gain de précision. Cependant aucun terme de la suite générée par l'algorithme ne fournit une information valide.

1.3 Contributions

1.3.1 Domaines numériques abstraits

Nous avons défini un domaine numérique abstrait basé sur la convexité abstraite [Mor70, Sin97, Rub00]. En analyse convexe (classique), on peut caractériser les convexes fermés et

les fonctions convexes semi-continues inférieurement par des théorèmes de séparation par des hyperplans (Hahn-Banach version géométrique). La convexité abstraite consiste à fixer une famille de fonctions non nécessairement linéaires et de définir une notion de convexité à partir de séparation par des lignes de niveaux des éléments de la famille. A la manière de Sankaranarayanan et al, nous nous sommes intéressés à des invariants numériques de la forme $\{x \in \mathbb{R}^d \mid p(x) \leq \alpha(p),\ \forall\, p \in \mathbb{P}\}$ où $\mathbb{P}$ est une famille de fonctions non nécessairement linéaires et non nécessairement finie. Ces invariants numériques définissent des ensembles convexes abstraits par rapport à $\mathbb{P}$.

Par la suite, nous avons défini une fonction sémantique abstraite par rapport à une correspondance de Galois entre les convexes abstraits et les fonctions convexes abstraites. En raison d'un problème de calculabilité, nous avons défini une deuxième sémantique *relâchée* construite à partir de la théorie de la dualité. En outre, cette fonction fournit une sur-approximation de la fonction sémantique abstraite. La théorie de la dualité a également motivé une construction d'une itération sur les politiques dynamique pour calculer des invariants sur notre domaine numérique abstrait.

En pratique pour des programmes écrits en arithmétique affine, nous avons proposé la relaxation de Shor pour évaluer la fonction sémantique relâchée et ainsi générer des invariants numériques sous forme d'ellipsoïdes tronquées. Dans les exemples décrits dans cette thèse, nous utilisons des fonctions de Lyapunov quadratiques pour des systèmes discrets induits par les programmes.

La méthode pour les calculs d'ellipsoïdes tronquées a été prototypée sous la forme de fichiers Matlab. La totalité des fichiers représente 3700 lignes de codes.

1.3.2 Itérations sur les politiques et problème du plus petit point fixe

Nous avons montré que l'itération sur les politiques peut être raffinée de manière à produire toujours le plus petit point fixe dans le cas des jeux stochastiques. Ce raffinement repose sur des techniques de la théorie de Perron-Frobenius non-linéaire. De plus nous avons interprété le problème de minimalité comme un problème d'optimisation. Nous avons donc extrait une condition de premier ordre pour caractériser le plus petit point fixe. Toutefois, comme la fonction sémantique abstraite sur les intervalles peut être vue comme un opérateur de Shapley en information parfaite, elle est non différentiable et ne possède pas de propriétés de convexité. Nous avons utilisé une notion plus faible de différentiablité : la *semidifférentiabilité*, cette notion consiste à remplacer l'approximation locale affine classique de la fonction par une approximation locale par une fonction homogène continue. L'approche par la semidifférentielle combinée aux *rayons spectraux non-linéaires* nous a permis de caractériser les points fixes localement minimaux. Dans le cas des contractions au sens large comme les opérateurs de Shapley, la caractérisation concerne uniquement le plus petit point fixe. Cette caractérisation a conduit à un nouveau critère d'arrêt pour l'itération sur politique dans le cas des fonctions croissantes contractantes au sens large et affines par morceaux. Lorsque le point fixe est non minimal, le problème est réduit à exhiber un vecteur propre non-linéaire de la semidifférentielle sur le cône négatif standard. Un tel vecteur fournit « direction de descente » qui permet de trouver une nouvelle politique conduisant à un point fixe strictement plus petit que le point fixe courant.

La méthode a été implémentée en C. Le prototype actuel est une amélioration d'un prototype existant implémentant l'itération sur les politiques dans le domaine des intervalles. Le prototype comprend désormais environ 15000 lignes de code.

1.4 Plan de thèse

La première partie constitue un rappel des outils de l'analyse statique par interprétation abstraite ainsi que les méthodes de calcul de point fixe utilisées actuellement en interprétation abstraite. Dans le chapitre 2, nous rappelons également les principes de base de l'interprétation abstraite et notamment la construction des fonctions sémantiques et des domaines abstraits nécessaires au calcul du plus petit point fixe. Nous terminons ce chapitre par quelques domaines numériques les plus fréquemment utilisés en interprétation abstraite. Dans le chapitre 3, nous présentons les méthodes de calcul de point fixe de l'interprétation abstraite. Nous débutons par des rappels sur l'itération de Kleene et les techniques d'accélération couplées à cette méthode. Puis, nous donnons des approches alternatives qui n'utilisent pas des techniques d'accélération. En particulier, nous développons dans ce chapitre, la méthode d'itération sur les politiques, en respectant l'ordre historique puisque nous commençons par traiter l'itération sur les politiques dans le cadre des jeux stochastiques pour finir par son utilisation en interprétation abstraite.

Dans la deuxième partie, nous présentons notre première contribution : les domaines numériques abstraits des sous-niveaux. Celle-ci a fait l'objet de l'article de conférence [AGG10]. Dans le chapitre 4, nous définissons les concepts de base du domaine abstrait des sous-niveaux ainsi que les théorèmes importants de l'interprétation abstraite comme l'existence d'une correspondance de Galois entre les ensembles convexes abstraits et les fonctions convexes abstraites et l'isomorphisme de treillis. Dans le chapitre 5, nous construisons les équations de points fixes à résoudre pour calculer nos invariants numériques et nous proposerons une itération sur les politiques « dynamique » dans le domaine des sous-niveaux basée sur la théorie de la dualité lagrangienne et de la programmation semi-infinie. Le chapitre 6 fournit un exemple de calculs d'invariants dans des domaines numériques avec une base finie de fonctions de base quadratiques, nous donnons dans ce chapitre des exemples d'analyse de programmes dans ces domaines. Enfin, le chapitre 7 conclut cette partie.

Dans la dernière partie, nous donnons une réponse partielle au problème de minimalité, notamment dans le cadre des fonctions croissantes affines par morceaux contractantes au sens large. Cette partie correspond à l'article de conférence [AGG08a]. Dans le chapitre 8, nous proposons une caractérisation d'un plus petit point fixe fini grâce à la semidifférentielle et aux rayons spectraux non-linéaires. Cette caractérisation est la base du raffinement de l'algorithme d'itération sur les politiques pour le calcul d'un plus petit point fixe fini. Dans le chapitre 9, nous appliquons l'itération sur les politiques pour le calcul du plus petit point fixe à des problèmes de jeux stochastiques positifs et à l'interprétation abstraite dans le domaine des intervalles. Le chapitre 10 constitue la conclusion de cette partie.

Première partie

Analyse statique de programmes par interprétation abstraite

CHAPITRE 2

PRINCIPE DE L'ANALYSE STATIQUE PAR INTERPRÉTATION ABSTRAITE

L'analyse statique de programmes consiste à extraire des propriétés sur les programmes. Pour déduire ces propriétés, l'analyse statique se base sur l'étude sémantique de la représentation syntaxique du programme. L'analyse est dite statique puisqu'elle se fait sans exécuter le programme. La propriété à vérifier peut être la preuve que le programme termine, qu'un certain état du programme est atteint, qu'aucune erreur ne se produit pendant l'exécution du programme ou que les valeurs des variables du programme sont bornées... Les propriétés de terminaison et d'atteignabilité sont dites de fatalité, c'est-à-dire qu'elles sont vraies à partir d'un certain temps pour une exécution du programme. La propriété de bornes est qualifiée de propriété de sûreté, car elles assurent qu'une propriété invariante sera toujours respectée quelque soit les exécutions. Cela est souvent utilisé pour prouver que les exécutions possibles restent toujours dans des bornes acceptables et que rien de mauvais ne se passera. Pour vérifier qu'une propriété de sûreté est satisfaite, il faut vérifier que cette propriété est respectée pour tous les comportements du programme, une telle propriété est appelée *invariant* du programme. Dans cette thèse, nous nous intéressons au calcul d'un type d'invariant numérique : trouver des bornes sur les valeurs des variables d'un programme valides pour tout état du programme. Le calcul de ces bornes doit être le plus précis possible afin que l'ensemble déterminé grâce aux bornes contienne peu de valeurs non atteintes par les variables du programme.

Le calcul d'invariant revient à chercher la plus petite solution d'un problème de point fixe dans un ensemble ordonné. Ce problème de point fixe est aussi appelé *équation sémantique* et le plus petit point fixe solution de cette équation est appelé *sémantique concrète*. Cependant, la sémantique concrète d'un langage de programmation, n'est, en général, pas calculable.

Le but de l'analyse statique par interprétation abstraite [CC77] est de construire, à partir de l'équation sémantique concrète, une équation abstraite définie sur un *domaine abstrait*, sur lesquels nous pouvons calculer une sur-approximation sûre de la sémantique concrète. Dans de nombreux cas, la sûreté est assurée par l'existence d'une correspondance de Galois entre le *domaine concret*, sur lequel la sémantique concrète est construite, et le domaine abstrait. La sur-approximation est appelée *sémantique abstraite*, c'est la plus petite solution de l'équation sémantique abstraite.

Pour commencer, nous rappelons quelques définitions sur les ensembles ordonnés et des

théorèmes de points fixes sur ces ensembles, section 2.1. Puis nous expliquons comment construire la sémantique concrète sur un langage jouet, section 2.2. Enfin nous décrivons la méthode de l'interprétation abstraite, section 2.3.

2.1 Rappels sur les ensembles ordonnés et les points fixes

Dans cette section, nous rappelons les structures ensemblistes qui vont être utiles pour décrire le principe de l'interprétation abstraite. Pour ces rappels, nous nous sommes inspirés du livre [GHK+80]. Dans notre problème de calcul d'invariants numériques, la notion de précision et donc la notion d'ordre intervient, ceci explique l'introduction d'ensembles ordonnés, définition 2.1. De plus, nous souhaitons un invariant le plus précis possible, nous devons nous assurer que le plus précis possible a un sens, définition 2.5. Enfin, la sémantique, qu'elle soit concrète ou abstraite, est définie comme le plus petit point fixe d'une application. Nous donnerons par conséquent les deux théorèmes importants de points fixes, le théorème de Tarski 2.1 et le théorème de Kleene 2.2 qui assurent, d'une part, l'existence et d'autre part, une méthode de calcul d'une sémantique.

2.1.1 Les ensembles ordonnés

Définition 2.1 (*Ensemble ordonné*)

On appelle ordre partiel sur un ensemble E, une relation $\preceq$:
- réflexive : $\forall\, x \in E,\ x \preceq x$.
- antisymétrique : $\forall\, x, y \in E,\ x \preceq y$ et $y \preceq x$ implique que $x = y$.
- transitive : $\forall\, x, y, z \in E,\ x \preceq y$ et $y \preceq z$ implique que $x \preceq z$.

On dira que $(E, \preceq)$ muni d'une telle relation est un ensemble ordonné.

Exemple 2.1
Soit E un ensemble non vide, la relation d'inclusion $\subseteq$ entre deux sous-ensembles de E est une relation d'ordre sur $\wp(E)$, l'ensemble des parties de E.

Exemple 2.2
Soit E un ensemble non vide. On définit la relation binaire $\subseteq_2$, sur $\wp(E)^2$, par $(A_1, A_2) \subseteq_2 (B_1, B_2)$ ssi $A_1 \subseteq B_1$ et $A_2 \subseteq B_2$. La relation $\subseteq_2$ est une relation d'ordre sur $\wp(E)^2$.

Remarque 2.1 Par récurrence sur $n \in \mathbb{N}$, pour tout $n \in \mathbb{N}$, $\wp(E)^n$ muni de la relation $\subseteq_n$ définie par $A = (A_i)_{i=1,\ldots,n} \subseteq_n B = (B_i)_{i=1,\ldots,n}$ ssi $A_i \subseteq B_i$, pour tout $i = 1, \ldots, n$, est un ensemble ordonné.

Exemple 2.3
Nous introduisons le graphe orienté $G = (S, A)$ où S désigne l'ensemble des sommets et A l'ensemble des arcs :

On définit pour le graphe 2.1 la relation binaire $\preceq$ par $S_i \preceq S_j$ ssi il existe un chemin menant de S_i à S_j c'est-à-dire il existe $S_{i_1}, \ldots, S_{i_k}$ tels que pour $l \in \{1, \ldots, k-1\}$, $(S_{i_l}, S_{i_{l+1}}) \in A$, $S_{i_1} = S_i$ et $S_{i_k} = S_j$. La relation $\preceq$ est une relation d'ordre sur S.

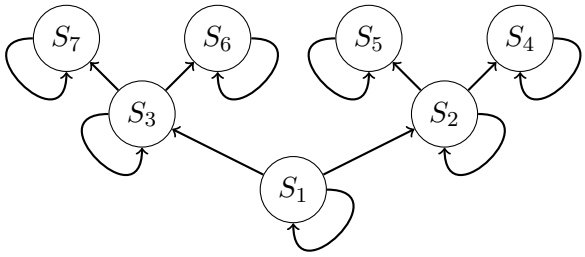


Figure 2.1 – Graphe orienté de l'exemple 2.3

Définition 2.2 (*Borne inférieure*)

Soit $(E, \preceq)$ un ensemble ordonné et $F \subseteq E$, $\underline{e}$ de E est la borne inférieure de F s'il vérifie :

$$\forall\, x \in F,\ \underline{e} \preceq x \text{ et si } \forall\, x \in F,\ y \preceq x \text{ alors } y \preceq \underline{e}\ .$$

Remarque 2.2 Si $\underline{e} \preceq x$, pour tout $x \in F$, on dit que $\underline{e}$ est un minorant de F. La borne inférieure de F est donc le plus grand des minorants.

Définition 2.3 (*Borne supérieure*)

Soit $(E, \preceq)$ un ensemble ordonné et $F \subseteq E$, $\overline{e}$ de E est la borne supérieure de F s'il vérifie :

$$\forall\, x \in F,\ x \preceq \overline{e} \text{ et si } \forall\, x \in F,\ x \preceq y \text{ alors } \overline{e} \preceq y\ .$$

Remarque 2.3 Si $x \preceq \overline{e}$, pour tout $x \in F$, on dit que $\overline{e}$ est un majorant de F. La borne supérieure de F est donc le plus petit des majorants.

Exemple 2.4

Soit $A, B \in \wp(E)$, $A \cup B$ est la borne supérieure de $\{A, B\}$. Sa borne inférieure est $A \cap B$.

Définition 2.4 (*Treillis*)

Un ensemble ordonné $(E, \preceq)$ est un treillis si tout sous-ensemble fini de E possède une borne inférieure et une borne supérieure.

Exemple 2.5

$\mathbb{R}$ muni de l'ordre usuel est un treillis.

Contre-exemple 2.1

Nous reprenons l'exemple 2.3. Le sous-ensemble $\{S_2, S_3\}$ n'admet pas de borne supérieure dans $(S, \preceq)$ car l'ensemble $\{S_2, S_3\}$ n'admet pas de majorant. Sa borne inférieure est S_1.

Définition 2.5 (*Treillis complet*)

Un treillis $(E, \preceq)$ est dit complet si tout sous-ensemble quelconque de E possède une borne inférieure et une borne supérieure.

On notera $\bot_E$ et $\top_E$ respectivement le plus petit et le plus grand élément de E. La borne inférieure d'un sous-ensemble F de E sera notée $\inf(F)$, sa borne supérieure $\sup(F)$.

Remarque 2.4 Lorsque le sous-ensemble ne contiendra que deux éléments x, y, on notera la borne inférieure de $\{x, y\}$, $\inf(x, y)$ et sa borne supérieure $\sup(x, y)$.

Exemple 2.6
Pour tout ensemble non vide E, $\wp(E)$ muni de la relation d'inclusion est un treillis complet avec pour plus grand élément E et pour plus petit élément $\emptyset$.

Contre-exemple 2.2
$\mathbb{R}$ muni de l'ordre usuel n'est pas un treillis complet.

Définition 2.6 (*Chaîne*)
Soit $(E, \preceq)$ un ensemble ordonné. Une chaîne F est un sous-ensemble totalement ordonné c'est-à-dire :

$$\forall\, x, y \in F,\ x \preceq y \text{ ou } y \preceq x\ .$$

Exemple 2.7
Pour l'exemple 2.3, tous les ensembles de sommets d'un chemin du graphe forment des chaînes sur $(S, \preceq)$.

Définition 2.7 (*CPO*)
Un ensemble ordonné $(E, \preceq)$ est dit complet, ou CPO (en anglais complete partial order), si toute chaîne non vide possède une borne supérieure.

Exemple 2.8
Les treillis complets sont des CPO.

Exemple 2.9
L'ensemble ordonné $(S, \preceq)$ défini à l'exemple 2.3 est un CPO.

Contre-exemple 2.3
Considérons la suite définie par $u_0 = 0$ et, pour tout $n \in \mathbb{N}$, $u_{n+1} = \sqrt{6 - u_n}$. Cette suite converge vers 2, elle est donc bornée. Mais l'ensemble $U = \{u_n \mid n \in \mathbb{N}\}$ muni de l'ordre usuel de $\mathbb{R}$, n'est pas un CPO puisque la chaîne $\{u_{2n} \mid n \in \mathbb{N}\}$ n'admet pas de borne supérieure. En effet, la fonction $x \mapsto \sqrt{6 - x}$ est décroissante, d'où, la croissance de la suite u_{2n}, la suite est donc majorée par 2 mais il n'existe pas d'entier $n \in \mathbb{N}$ tel que $u_n = 2$ et on conclut que $\{u_{2n} \mid n \in \mathbb{N}\}$ n'admet pas de borne supérieure dans U.

Exemple 2.10
Reprenons le contre-exemple 2.3, l'ensemble $U \cup \{2\}$ est un CPO.

2.1.2 Points fixes sur les ensembles ordonnés

Les applications croissantes forment une classe d'applications primordiale pour la théorie des points fixes sur les ensembles ordonnés.

Définition 2.8 (*Fonction croissante*)
Soient deux ensembles ordonnés $(E, \preceq_E)$ et $(F, \preceq_F)$. Une application f de E dans F est dite croissante si

$$\forall\, x, y \in E,\ x \preceq_E y \implies f(x) \preceq_F f(y)\ .$$

Si ($x \preceq_E y$ et $x \neq y$) implique que ($f(x) \preceq_F f(y)$ et $f(x) \neq f(y)$) alors f est dite strictement croissante.

Exemple 2.11
Les applications $\cup$ et $\cap$ de $(\wp(E)^2, \subseteq_2)$ dans $(\wp(E), \subseteq)$ sont croissantes.

Définition 2.9 (*Fonctions continues sur un CPO*)
Soient $(E, \preceq_E)$ et $(F, \preceq_F)$ deux CPO. Une application f de E dans F est dite continue si :
- f est croissante.
- Pour toute chaîne G de E :

$$\sup_F(\{f(x) \mid x \in G\}) = f\left(\sup_E(G)\right) \quad .$$

Autrement dit, une fonction continue sur un CPO est croissante et préserve les bornes supérieures.

Remarque 2.5 Un CPO peut être muni d'une topologie spéciale : la topologie de Scott, ainsi la définition de la continuité 2.9, correspond à la continuité au sens de la topologie de Scott (voir par exemple [GHK+80]).

Exemple 2.12
Soit $(E, \preceq)$ un CPO, la fonction Id, telle que, pour tout $x \in E$, $Id(x) = x$ est continue sur $(E, \preceq)$.

Exemple 2.13
Reprenons l'exemple 2.10. Une fonction continue sur $\mathbb{R}$, pour la topologie usuelle, est continue sur le CPO $U \cup \{2\}$.

Contre-exemple 2.4
La fonction f, telle que, pour tout $u \in U$, défini à l'exemple 2.10, $f(u) = 0$ et $f(2) = 1$ n'est pas continue de $(U \cup \{2\}, \leq)$ sur $(\{0, 1\}, \leq)$. Considérons la chaîne $\{u_{2n+1} \mid n \in \mathbb{N}\} \cup \{2\}$, par décroissance de $x \mapsto \sqrt{6 - x}$, la suite $(u_{2n+1})_{n\in\mathbb{N}}$ est décroissante et minorée par 2. Par conséquent, $\sup_U\{u_{2n+1} \mid n \in \mathbb{N}\} \cup \{2\} = u_1 = \sqrt{6}$ d'où $f(\sup_U\{u_{2n+1} \mid n \in \mathbb{N}\} \cup \{2\}) = 0$ mais la borne supérieure de $\{f(u) \mid u \in \{u_{2n+1} \mid n \in \mathbb{N}\} \cup \{2\}\} = f(2) = 1$.

Définition 2.10 (*Point fixe*)
Soient X un ensemble et f une application sur X. Un élément $x \in X$ est un point fixe si $f(x) = x$.
On notera $\text{fix}(f)$ l'ensemble des points fixes de f. Le plus petit point fixe de f, s'il existe, sera noté $\text{lfp}(f)$, son plus grand point fixe, s'il existe, $\text{gfp}(f)$.

Exemple 2.14
Soit f la fonction de $[-30, 6]$ dans $[-30, 6]$, telle que pour tout $x \in [-30, 6]$, $f(x) = \sqrt{6 - x}$. Le réel 2 est un point fixe de f.

Définition 2.11 (*Post point fixe et pré point fixe*)
Soient $(X, \preceq)$ un ensemble ordonné et f une application sur X. Un élément $x \in X$ est un post point fixe si $f(x) \preceq x$. Un élément $x \in X$ est un pré point fixe si $x \preceq f(x)$

Théorème 2.1 (*Théorème du point fixe de Tarski, [Tar55]*)

Soient $(E, \preceq)$ un treillis complet et f une application croissante sur E. Les points fixes $\mathrm{lfp}(f)$ et $\mathrm{gfp}(f)$ existent et :

$$\mathrm{lfp}(f) = \inf\{x \in E \mid f(x) \preceq x\} \text{ et } \mathrm{gfp}(f) = \sup\{x \in E \mid x \preceq f(x)\} \ .$$

Pour une application f d'une ensemble E dans lui-même, nous notons f^i, l'application définie par $f^i = f$ si $i = 1$ et $f^{i+1} = f(f^i)$.

Théorème 2.2 (*Théorème du point fixe de Kleene, [Kle52]*)

Soient $(E, \preceq)$ un CPO et f une application continue sur E. Le point $\mathrm{lfp}(f)$ existe et :

$$\mathrm{lfp}(f) = \sup\{f^i(\bot_E) \mid i \in \mathbb{N}\} \ .$$

Remarque 2.6 Le théorème de Kleene admet également une formulation duale pour le plus grand point fixe $\mathrm{gfp}(f)$.

2.2 Sémantique concrète d'un programme

Pour analyser un programme P, nous commençons par donner la syntaxe, sous-section 2.2.1 permettant d'écrire un programme P puis l'interprétation mathématique du langage à l'aide du graphe de flot de contrôle, sous-section 2.2.2 et enfin la construction de notre équation de point fixe nécessaire au calcul d'invariants numériques, sous-section 2.2.3.

2.2.1 Syntaxe des programmes

Dans cette sous-section, nous décrivons la syntaxe utilisée dans le reste de la thèse. Elle est inspirée de la syntaxe SIMPLE e.g [Win93]. Une autre référence remarquable est la thèse d'Antoine Miné [Min04]. Elle définit quel type de programmes nous pouvons écrire et étudier. Nous posons $\overline{\mathbb{Z}} = \mathbb{Z} \cup \{+\infty\} \cup \{-\infty\}$, $\overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$, puis $\mathbb{G} \in \{\mathbb{Z}, \mathbb{R}\}$ et enfin $\overline{\mathbb{G}} \in \{\overline{\mathbb{Z}}, \overline{\mathbb{R}}\}$.

Nous notons d le nombre de variables. La notation $\mathcal{V}(P)$ désigne le vecteur des d variables du programme P. Nous introduisons également, la notation $\mathcal{V}_1(P)$ qui représente l'ensemble des variables d'un programme P. L'arithmétique utilisée pendant cette thèse est donnée par la classe des applications polynomiales :

Définition 2.12 (*Fonction polynomiale*)

On dit que $T : \mathbb{G}^d \mapsto \mathbb{G}$ est une fonction polynomiale multivariée en y si c'est une somme finie d'éléments de la forme :

$$p_\alpha y_1^{\alpha_1} y_2^{\alpha_2} \dots y_d^{\alpha_d}$$

où $p_\alpha \in \mathbb{G}$ et $\alpha_1, \alpha_2, \dots, \alpha_d$ sont des entiers naturels. Le vecteur $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)$ est appelé multi-entier.

Les expressions sont des intervalles fermés de $\mathbb{G}$, ou des images des variables du programme par une fonction polynomiale. L'image d'un sous-ensemble de $\mathbb{G}$ par une application polynomiale est l'union des images des vecteurs de ce sous-ensemble.

Les tests *test* peuvent être construits comme comparaison entre une image des variables du programme par une fonction polynomiale et une constante de $\overline{\mathbb{G}}$. Nous autorisons les mots-clés booléens **and** et **not** pour respectivement la conjonction et la négation de tests. Cependant, même si elles sont redondantes, les constantes booléennes **true** et **false** existent dans notre syntaxe. En effet, on peut choisir a égal à $+\infty$ et donc $expr \leq a$ est toujours vrai, de même qu'en choisissant a égal à $-\infty$, $expr \leq a$ est toujours faux.

Les instructions élémentaires *inst* sont des affectations, elles associent une nouvelle expression à une variable. On autorise aussi le mot-clé **assume** qui comme le nom l'indique, permet de supposer que le vecteur de variables, à un certain moment du programme appartient à un ensemble C. L'instruction **skip** permet d'ajouter des instructions qui ne modifient pas l'état de la mémoire. Cette instruction sera utile plus tard pour ajouter des points de contrôle sans modifier sémantiquement la fonction. Les branchements conditionnels **if then else**, indiquent qu'on exécute une instruction si un test est vrai et une autre instruction sinon. Les boucles correspondent à l'exécution d'une instruction tant qu'un test est satisfait.

Les expressions, tests et instructions utilisés sont décrits plus formellement à la figure 2.2.

expr	::=	a	$a \in \mathbb{G}$
	|	$[a, b]$	$a, b \in \mathbb{G}$
	|	$T(X)$	$T : \mathbb{G} \mapsto \mathbb{G}$ polynomiale $X \in \mathcal{V}_1(\cdot)$
test	::=	**true**	
	|	**false**	
	|	$T(X) \leq a$	$a \in \overline{\mathbb{G}}$, $X \in \mathcal{V}_1(\cdot)$ $T : \mathbb{G} \mapsto \mathbb{G}$ polynomiale
	|	*test* **and** *test*	
	|	**not** *test*	
inst	::=	**assume** $(X_1, \ldots, X_d) \in C$	$(X_1, \ldots, X_d) \in \mathcal{V}(\cdot)$ $C \subseteq \mathbb{G}^d$
	|	**skip**	
	|	$X = expr$	$X \in \mathcal{V}_1(\cdot)$
	|	**if** *test***then** $\{inst\}$ **else** $\{inst\}$	
	|	**while** *test* $\{inst\}$	
	|	*inst* ; *inst*	

Figure 2.2 – Syntaxe autorisée pour écrire un programme

La syntaxe étant définie, il faut en donner un sens mathématique manipulable pour nos calculs d'invariants.

Exemple 2.15
Le programme C suivant est un programme bien écrit dans la syntaxe définie par notre grammaire 2.2 :

```
x=1;
while (x<=10){
  x=x+1;
  }
```

Exemple 2.16
Le programme avec un branchement conditionnel suivant est un programme bien écrit dans la syntaxe définie par notre grammaire 2.2 :

```
x=[0,1];
y=[0,1];
if (x-y<=0)
    then {
      x=x+1;
      }
    else {
      y=y+1;
      }
```

2.2.2 Graphe de flots de contrôle et sémantique concrète

Pour analyser un programme, nous analysons les variables en tant que vecteur de $\mathbb{G}^d$. Ainsi les expressions constantes deviennent des sous-ensembles de $\mathbb{G}^d$, les fonctions polynomiales deviennent des fonctions polynomiales à valeurs vectorielles (chaque coordonnée de la fonction est une fonction polynomiale). Ces fonctions à valeurs vectorielles sont supposées à valeurs dans $\mathbb{G}^d$. Même si la fonction ne modifie pas toutes les variables, on peut compléter cette fonction par la fonction identité sur les composantes des variables non affectées par la fonction. Les tests sont traités comme des inégalités vectorielles au sens de l'ordre produit usuel. De manière similaire aux fonctions, on peut compléter l'inégalité vectorielle par des inégalités toujours vraies sur les composantes n'intervenant pas dans le test. Les affectations sont vues comme des affections parallèles vectorielles.

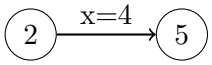
Graphe de flots de contrôle

Le graphe de flots de contrôle est un graphe orienté représentant la suite des instructions d'un programme informatique. Pour décrire un programme, celui-ci peut être découpé à l'aide de points de contrôle étiquetés par des entiers car en nombre fini. Ces points de contrôle jouent le rôle de sommets dans le graphe. Un arc du graphe représente une transition entre deux points de contrôle. Une transition est soit un vecteur d'affectations soit un vecteur de tests de la syntaxe 2.2. Un arc sera étiqueté par une transition. Le graphe de flots de contrôle étant construit à partir d'un programme informatique, ce graphe contient un sommet dont le degré entrant est 0 et un sommet dont le degré sortant est 0. Sur le graphe, l'unique sommet de degré entrant nul sera symbolisé par le nombre 0, les sommets de degré sortant nul seront caractérisés par un nombre entier non nul encadré qui symbolise les points de sortie du programme, les autres sommets sont des nombres entiers non nuls associés aux points de contrôle.

On notera $\mathcal{B}(P)$ l'ensemble des points de contrôle du programme P, $\mathcal{L}(P)$, l'ensemble des affectations et des tests de P et $\mathcal{G}(P) = (\mathcal{B}(P), \mathcal{T}(P), \mathcal{L}(P))$ le graphe de flots de contrôle

associé à P avec $\mathcal{T}(P) \subseteq \mathcal{B}(P)^2$. Une relation est notée $((i,j),\ell)$ où $i,j \in \mathcal{B}(P)$ et $\ell \in \mathcal{L}(P)$. Nous notons l'ensemble des relations d'un programme étiqueté P, $\mathcal{R}(P)$.

Exemple 2.17
La transition entre les points de contrôle 2 et 5, $x = 4$ sera symbolisée par la relation $((2,5), x = 4)$.



Sur le graphe, un sommet aura deux arcs sortants si le point de contrôle associé représente une boucle ou un branchement conditionnel. Un arc sera étiqueté par le test d'entrée de boucle ou de branchement conditionnel, l'autre arc sera étiqueté par la négation du test.

A la manière de Miné [Min04], on ajoute un point de contrôle et un arc non étiqueté qui représente des instructions **skip** après chaque affectation :

- si elle précède **while** ou si celle-ci est la dernière affectation d'une boucle **while**.
- si c'est la dernière affectation des parcours **then** et **else** d'un branchement conditionnel **if**.

Ce découpage sera très pratique pour la résolution de l'équation de point fixe.

Exemple 2.18
A la figure 2.3, nous nous intéressons à un simple programme contenant une boucle qui, jusqu'à la valeur 11, à chaque itération, incrémente la variable x initialisée à 0.

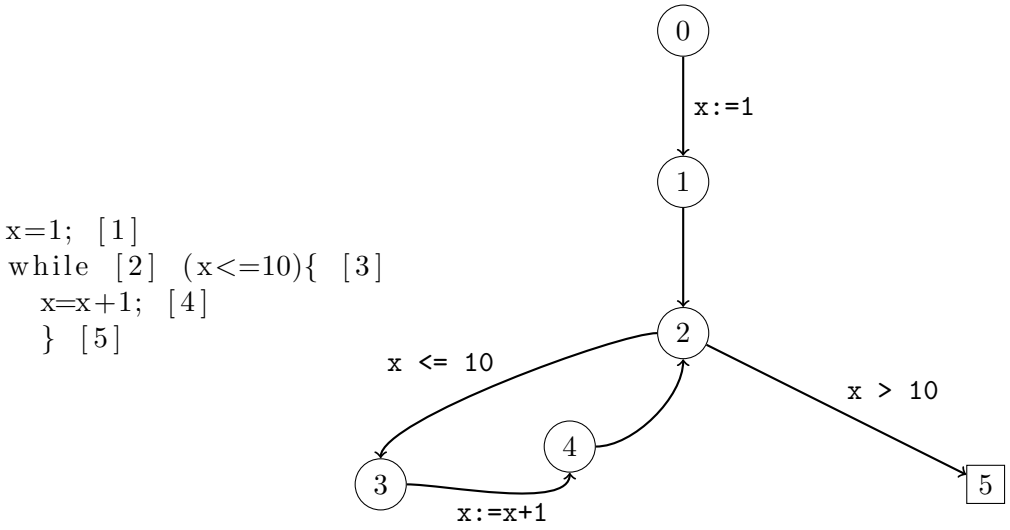


Figure 2.3 – Un programme contenant une boucle à gauche et son graphe de flot de contrôle à droite

Exemple 2.19
A la figure 2.4, nous regardons un simple programme avec un branchement conditionnel **if**.

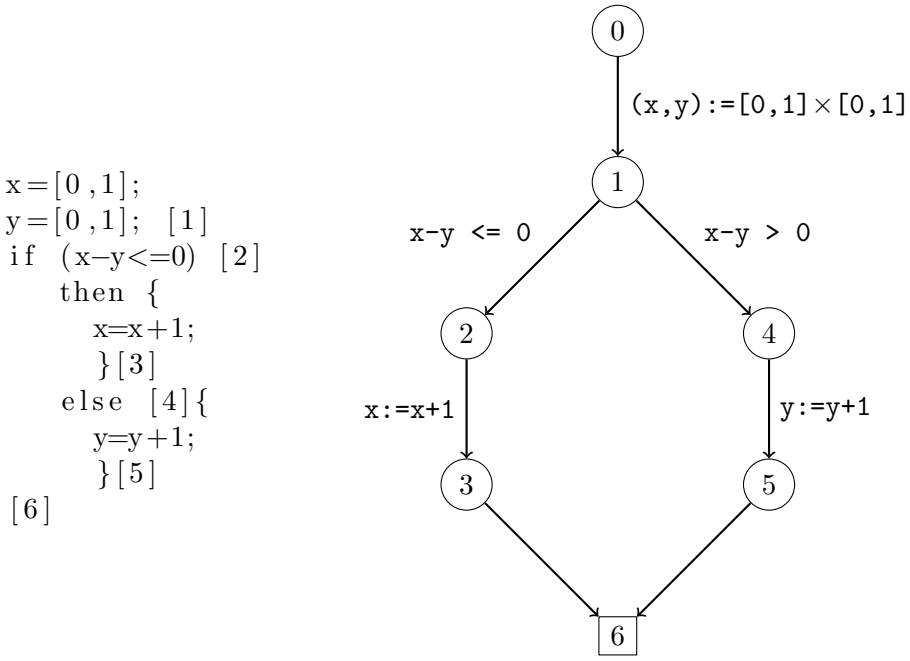


Figure 2.4 – Un programme avec un branchement conditionnel à gauche et son graphe de flot de contrôle

Sémantique concrète

Pour étudier un programme P écrit grâce à la syntaxe définie à la figure 2.2, il faut donner une interprétation de chaque élément composant cette syntaxe. Toutefois, comme nous représentons le programme par un graphe de flots de contrôle, nous devons interpréter la syntaxe 2.2. En effet, grâce aux exemples 2.18, 2.19, il suffit d'interpréter les expressions, les tests et les affectations pour pouvoir analyser un programme. L'interprétation des éléments de la syntaxe 2.2 est donnée par une sémantique. Une sémantique décrit tous les états possibles d'un programme en fonction des états initiaux. Pour définir plus formellement ce qu'est une sémantique, nous introduisons la notion d'environnement. Un environnement est une fonction qui associe aux variables du programme une valeur ou plutôt un vecteur dans $\mathbb{G}^d$ dans notre cas. Nous notons Σ l'ensemble des environnements c'est-à-dire $\Sigma = \{\sigma \mid \sigma : \mathcal{V}(P) \mapsto \wp(\mathbb{G}^d)\}$. Concrètement, un environnement retourne les valeurs des variables dans la mémoire de la machine à un instant donné de l'exécution du programme P. D'un point de vue mathématique, une sémantique correspond à une fonction sur l'ensemble des environnements qui renvoie un sous-ensemble de $\mathbb{G}^d$ dans le cas des expressions, un booléen dans le cas des tests et un environnement dans le cas des instructions. Une sémantique est dite concrète quand elle décrit précisément la manière dont s'exécute un programme.

Nous rappelons qu'une expression est considérée comme vectorielle, c'est-à-dire, un élément de $\mathbb{G}^d$. Pour une expression $expr$, $[\![expr]\!]$ désigne une application de Σ dans $\wp(\mathbb{G}^d)$. Soit $\sigma \in \Sigma$, l'interprétation des expressions est détaillée à la figure 2.5 :

L'interprétation d'un sous-ensemble de $\mathbb{G}^d$ est le sous-ensemble lui-même car il représente une constante dans le programme et donc ne varie pas au cours de l'exécution de celui-ci et,

$$\begin{array}{lcl} [\![C]\!](\sigma) & = & C \\ [\![T(X)]\!](\sigma) & = & \{T(a) \mid a \in \sigma(X)\} \end{array}$$

Figure 2.5 – Interprétation des expressions

par conséquent, sa valeur est indépendante de l'état mémoire. L'interprétation des variables dans un environnement est le vecteur de valeurs en mémoire. Enfin, l'interprétation de l'image d'une expression par un polynôme est l'image des valeurs prises par cette expression dans un environnement donné.

Les tests sont pour rappels de la forme $r(X) \leq a$ où r est à valeurs dans $\mathbb{G}^d$ et $a \in \overline{\mathbb{G}}^d$. Pour un test *test*, $[\![test]\!]$ est une application de Σ dans l'ensemble des parties des booléens $\mathbb{B}$ c'est-à-dire $\wp(\{\mathbf{true}, \mathbf{false}\})$. Soit $\sigma \in \Sigma$, l'interprétation des tests est donnée à la figure 2.6 :

$$\begin{array}{lcl} [\![\mathbf{true}]\!](\sigma) & = & \mathbf{true} \\ [\![\mathbf{false}]\!](\sigma) & = & \mathbf{false} \\ [\![T(X) \leq a]\!](\sigma) & = & \begin{cases} \{\mathbf{true}, \mathbf{false}\} & \text{si } \exists\, y,\ z \in [\![T(X)]\!](\sigma), y \leq a,\ z > a \\ \{\mathbf{true}\} & \text{si } \exists\, y \in [\![T(X)]\!](\sigma),\ y \leq a \\ \{\mathbf{false}\} & \text{si } \exists\, y \in [\![T(X)]\!](\sigma),\ y > a \\ \emptyset & \text{sinon} \end{cases} \\ [\![test1 \ \mathbf{and}\ test2]\!](\sigma) & = & \{t_1 \wedge t_2 \mid t_1 \in [\![test1]\!](\sigma),\ t_2 \in [\![test2]\!](\sigma)\} \\ [\![\mathbf{not}\ test]\!](\sigma) & = & \{\neg t \mid t \in [\![test]\!](\sigma)\} \end{array}$$

Figure 2.6 – Interprétation des tests

Les tests **true** et **false** sont indépendants de l'environnement, la signification du test $T(X) \leq a$, où T est une fonction polynomiale, pour un environnement donné est l'union des booléens **true** et **false** si l'interprétation de $T(X)$ contient à la fois des éléments plus petits et strictement plus grands, au sens de l'ordre usuel de $\overline{\mathbb{G}}^d$, que l'élément $a \in \overline{\mathbb{G}}^d$. Le test **not** *test* est interprété comme l'ensemble des négations des éléments de l'interprétation de *test*.

L'affectation d'une expression à une variable change son état mémoire, il faut donc interpréter cette affectation comme un nouvel état mémoire, c'est-à-dire un environnement, $[\![X = expr]\!]$ est, par conséquent, une application de l'ensemble des environnements dans lui-même, et on a, pour $\sigma \in \Sigma$,

$$[\![X = expr]\!](\sigma) = \sigma[X \to [\![expr]\!](\sigma)],$$

où, $\sigma[X \to [\![expr]\!](\sigma)]$ signifie que l'on remplace les valeurs mémoires du vecteur X par les valeurs mémoires de *expr* interprétées dans l'environnement σ.

Exemple 2.20
L'affectation $x := 5$ est interprétée par $[\![x = 5]\!](\sigma) = \sigma[x \to [\![5]\!](\sigma)] = \sigma[x \to 5]$. Finalement, $\sigma[x \to 5](x) = 5$.

Exemple 2.21
Supposons que, $\sigma(x) = 2$ (2 est la valeur en mémoire au moment où on évalue la variable x), l'affectation $x = x^3$ s'interprète de la manière suivante : posons $\sigma' = \sigma[x \to [\![x^3]\!](\sigma)] = \sigma[x \to (\sigma(x))^3]$, d'où, $\sigma'(x) = 8$.

2.2.3 Sémantique collectrice

Nous voulons trouver des bornes sur les valeurs des variables à chaque point de contrôle. Les valeurs de ces variables sont données par les environnements. Une méthode pour borner ces valeurs est de collecter toutes les valeurs. Cette méthode est classique, on collecte tous les environnements construits à un point de contrôle donné, on parle dans ce cas de sémantique collectrice. Nous définissons cette sémantique à partir du graphe de flots de contrôle. Il faut donc donner l'interprétation des affectations, et des tests. Puisqu'on agrège des environnements, nous devons travailler avec des sous-ensembles d'environnements. La sémantique collectrice est donc une fonction sur les parties de Σ.

Comme, nous avons ajouté des arcs sans étiquette qui représentent des instructions **skip**. La sémantique collectrice $[\![.]\!]_C$ sur une instruction **skip** est l'identité, en d'autres termes, $[\![\textbf{skip}]\!]_C(S) = S$ où S est un sous-ensemble de Σ.

Nous définissons la sémantique collectrice $[\![.]\!]_C$ sur les affectations comme l'union des environnements construits à partir de cette affectation c'est-à-dire, pour un sous-ensemble non vide S de Σ :

$$[\![X = expr]\!]_C(S) = \bigcup_{\sigma \in S} \{\sigma[X \to [\![expr]\!](\sigma)]\}$$

Bien évidemment, $[\![X = expr]\!]_C(S) = \emptyset$ si $S = \emptyset$.

Pour les tests, on filtre les environnements de S qui satisfont les conditions du test, plus formellement, la sémantique collectrice d'un test $test$:

$$[\![test]\!]_C(S) = \{\sigma \in S \mid \textbf{true} \in [\![test]\!](\sigma)\}$$

Bien évidemment, $[\![test]\!]_C(S) = \emptyset$ si $S = \emptyset$.

Le nombre entier n désigne le nombre de point de contrôles pour analyser le programme P. A présent, on peut définir la fonction sémantique concrète F^C de $(\wp(\Sigma))^{n+1}$ dans lui même, construite à partir de la sémantique collectrice. La coordonnée i-ème de F^C désigne le nouvel ensemble d'environnements créé au point de contrôle i. La fonction sémantique concrète F^C est définie par :

$$\forall\, i \in \{0, \ldots, n\},\ F_i^C(S_0, S_1, \ldots, S_n) = \begin{cases} S_0 & \text{si } i = 0 \\ \displaystyle\bigcup_{((j,i),\ell) \in \mathcal{R}(P)} [\![\ell]\!]_C(S_j) & \text{si } i \neq 0 \end{cases}$$

Soit $\bar{S} = (\bar{S}_i)_{i=0,\ldots,n}$ le vecteur des ensembles d'environnement associés au point de contrôle i.

Nous ne nous intéressons pas aux valeurs des cases mémoires des variables d'un programme avant son lancement. Par conséquent, nous supposons que l'ensemble d'environnement S_0 ne contient que l'environnement qui renvoie l'ensemble vide, cet environnement est noté σ_0.

L'image de $\bar{S}$ par F^C est exactement le vecteur $\bar{S}$. Pour trouver les ensembles d'environnements à chaque point de contrôle, il faut donc résoudre l'équation de point fixe dans $\wp(\Sigma)^{n+1}$:

$$S = F^C(S) \tag{2.1}$$

Exemple 2.22
A la figure 2.7, nous considérons l'équation sémantique du programme C de l'exemple 2.18.

```
x=1; [1]
while [2] (x<=10){ [3]
  x=x+1; [4]
  } [5]
```

$$
\begin{array}{lcl}
S_0 & = & \{\sigma_0\} \\
S_1 & = & [\![x = 1]\!](S_0) \\
S_2 & = & [\![\textbf{skip}]\!](S_1) \cup [\![\textbf{skip}]\!](S_4) \\
S_3 & = & [\![x <= 10]\!](S_2) \\
S_4 & = & [\![x = x + 1]\!](S_3) \\
S_5 & = & [\![x > 10]\!](S_2)
\end{array}
$$

Figure 2.7 – Un programme C à gauche et l'équation sémantique à droite

Exemple 2.23
A la figure 2.8, nous considérons l'équation sémantique du programme C de l'exemple 2.19.

```
x=[0,1];
y=[0,1]; [1]
if (x-y<=0) [2]
   then {
      x=x+1;
      }[3]
   else [4]{
      y=y+1;
      }[5]
[6]
```

$$
\begin{array}{lcl}
S_0 & = & \{\sigma_0\} \\
S_1 & = & [\![x = [0, 1], y = [0, 1]]\!](S_0) \\
S_2 & = & [\![x - y <= 0]\!](S_1) \\
S_3 & = & [\![x = x + 1]\!](S_2) \\
S_4 & = & [\![y - x < 0]\!](S_1) \\
S_5 & = & [\![y = y + 1]\!](S_4) \\
S_6 & = & [\![\textbf{skip}]\!](S_3) \cup [\![\textbf{skip}]\!](S_5)
\end{array}
$$

Figure 2.8 – Un programme C à gauche et l'équation sémantique à droite

De plus, nous souhaitons que les ensembles d'environnement exacts, ceux qui apparaissent vraiment pendant l'exécution du programme, la solution de l'équation (2.1), appelée aussi équation sémantique doit être la plus précise possible. Par ailleurs, l'ensemble $\wp(\Sigma)$ muni de l'ordre de l'inclusion est un treillis complet alors la plus précise signifie la plus petite au sens de l'inclusion dont l'existence est assurée par le théorème de Tarski 2.1, F^C étant croissante par construction.

Nous rappelons qu'un environnement est une fonction des variables d'un programme vers un vecteur de valeurs entières ou réelles. Dans cette thèse, nous souhaitons avoir des bornes sur les variables à chaque point de contrôle. Ces bornes peuvent être données à partir des ensembles d'environnements à chaque point de contrôle. En effet, prenons un vecteur de variables x qui correspond à un vecteur de $\mathbb{G}^d$, et appliquons un ensemble d'environnement $S = (S_i)_{i=0,\ldots,n}$, nous obtenons un sous-ensemble de $\mathbb{G}^d$, nous posons, pour un point de contrôle i, $X_i = S_i(x)$. L'équation sémantique (2.1) devient alors le problème 2.1 sur $\wp(\mathbb{G}^d)^{n+1}$.

Problème 2.1
Résoudre :

$$\inf\{X \in \wp(\mathbb{G}^d)^{n+1} \mid X = F^C(X)\} \ .$$

D'après le théorème de Tarski, comme F^C est croissante, le problème 2.1, est équivalent à :

Problème 2.2
Résoudre :
$$\inf\{X \in \wp(\mathbb{G}^d)^{n+1} \mid F^C(X) \subseteq X\} \ .$$

Exemple 2.24
Pour le programme de l'exemple 2.18, nous devons calculer la plus petite solution de l'équation :

$$\begin{array}{lcl} X_0 & = & \emptyset \\ X_1 & = & \{1\} \\ X_2 & = & X_1 \cup X_4 \\ X_3 & = & X_2 \cap]-\infty, 10] \\ X_4 & = & (X_3 + 1) \\ X_5 & = & X_2 \cap]10, +\infty[\end{array}$$

Exemple 2.25
Pour le programme de l'exemple 2.19, nous devons calculer la plus petite solution de l'équation :

$$\begin{array}{lcl} X_0 & = & \emptyset \\ X_1 & = & [0,1] \times [0,1] \\ X_2 & = & X_1 \cap \{x, y \in \mathbb{G}^d \mid x - y \leq 0\} \\ X_3 & = & X_2 + (1,0) \\ X_4 & = & X_1 \cap \{x, y \in \mathbb{G}^d \mid y - x < 0\} \\ X_5 & = & X_4 + (0,1) \\ X_6 & = & X_2 \cup X_5 \end{array}$$

Par le théorème de Tarski, l'équation (2.1) admet une solution. Néanmoins, cette équation est, en général, impossible à résoudre algorithmiquement. Le but de l'interprétation abstraite est donc de construire à partir de la fonction sémantique F^C, une fonction sémantique abstraite sur un domaine abstrait qui donne une sur-approximation de la solution de l'équation (2.1).

2.3 Principes de l'interprétation abstraite

La solution de l'équation (2.1) n'est pas calculable en général. Ce problème de calculabilité a deux causes principales :

– le domaine concret (les parties de $\mathbb{G}^d$) n'est pas représentable en machine ;
– la fonction sémantique concrète n'est pas calculable.

Le rôle de l'interprétation abstraite est de palier ce problème de calculabilité en calculant une sur-approximation de la sémantique concrète. solution d'un problème de point fixe dans un domaine dit *abstrait* pour une fonction sémantique dite *abstraite*.

Le plus petit point fixe calculé pour la fonction sémantique abstraite dans le domaine abstrait doit être une sur-approximation de la plus petite solution. Pour pouvoir utiliser le

théorème de Kleene ou le théorème de Tarski, on peut choisir comme domaine abstrait un CPO ou un treillis complet. La fonction sémantique abstraite choisie devra, par conséquent, être continue (au sens des CPO) si le domaine abstrait est un CPO et monotone si le domaine abstrait est un treillis complet.

Dans cette section, nous rappelons dans la sous-section 2.3.1, les définitions de domaine et sémantique abstraits puis dans la sous-section 2.3.2, nous présentons quelques domaines abstraits communément utilisés.

2.3.1 Domaines et sémantiques abstraits

Les définitions de cette sous-section sont inspirées de [CC92a]. On peut, a priori, définir un domaine abstrait comme la donnée d'un ensemble représentable en machine et d'une fonction γ du domaine abstrait vers le domaine concret. Nous avons besoin de la fonction γ pour associer, à une solution calculée dans le domaine abstrait, un élément concret. Par ailleurs, le but de l'interprétation abstraite est, certes, de sur-approximer la sémantique abstraite, mais aussi de garantir que cette sur-approximation est la plus précise possible. Par conséquent, pour pouvoir comparer les éléments abstraits, on requiert que le domaine abstrait soit un ensemble ordonné. De plus, la relation d'ordre choisie doit satisfaire que la fonction γ associée au domaine abstrait soit croissante. Ceci assure qu'un plus petit élément abstrait représente un élément concret plus précis. On en déduit la définition suivante :

Définition 2.13 (*Domaine abstrait*)

Un domaine abstrait $(\mathcal{D}_a, \gamma)$ est la donnée d'un ensemble ordonné $\mathcal{D}_a$ représentable en machine et d'une fonction croissante $\gamma : \mathcal{D}_a \mapsto \wp(\mathbb{G}^d)$.

Exemple 2.26
Soient l'ensemble :

$$\mathcal{I}(\overline{\mathbb{Z}}) = \{[a,b] \mid a \in \mathbb{Z} \cup \{-\infty\}, b \in \mathbb{Z} \cup \{+\infty\},\ a \leq b\} \cup \emptyset,$$

et la fonction γ de $\mathcal{I}(\overline{\mathbb{Z}})\backslash\{\emptyset\}$ dans $\wp(\mathbb{Z})$ définie par :

$$\gamma([a,b]) = \begin{cases} \mathbb{Z} & \text{si } a = -\infty,\ b = +\infty \\ ]-\infty, b] & \text{si } b \in \mathbb{Z},\ a = -\infty \\ [a, +\infty[& \text{si } a \in \mathbb{Z},\ b = +\infty \\ [a,b] & \text{si } a, b \in \mathbb{Z} \end{cases}$$

On pose également $\gamma(\emptyset) = \emptyset$. La paire $(\mathcal{I}(\overline{\mathbb{Z}}), \gamma)$ forme un domaine abstrait. On appelle ce domaine le domaine des intervalles.

Remarque 2.7 On prend comme convention $\emptyset = [+\infty, -\infty]$. On peut définir un isomorphisme entre $\mathcal{I}(\overline{\mathbb{Z}})$ et $\{(a,b) \in \overline{\mathbb{Z}}^2 \mid a \in \mathbb{Z} \cup \{-\infty\},\ b \in \mathbb{Z} \cup \{+\infty\},\ a \leq b\} \cup \{(+\infty, -\infty)\}$, en associant à un intervalle $[a,b]$ non vide à bornes entières le couple (a,b).

Exemple 2.27
Par extension de l'exemple 2.26, on définit :

$$\mathcal{I}(\overline{\mathbb{Z}}^d) = \left\{ \prod_{i=1}^{d} [a_i, b_i] \mid [a_i, b_i] \in \mathcal{I}(\overline{\mathbb{Z}}) \right\}$$

et la fonction $\rho : \mathcal{I}(\overline{\mathbb{Z}}^d) \mapsto \wp(\overline{\mathbb{Z}}^d)$ qui, à $\prod_{i=1}^{d}[a_i, b_i]$ associe $\prod_{i=1}^{d}\gamma([a_i, b_i])$ où γ est la fonction définie à l'exemple 2.26. L'ensemble $(\mathcal{I}(\overline{\mathbb{Z}}^d), \rho)$ forme un domaine abstrait.

Remarque 2.8 Nous notons respectivement x_i^- et x_i^+ la borne inférieure et la borne supérieure de l'intervalle x_i. Un élément x est un vecteur d'intervalles si toutes ses coordonnées sont de la forme $[x_i^-, x_i^+]$.

Définition 2.14 (*Fonction de concrétisation*)

Nous appelons fonction de concrétisation la fonction γ associée à un domaine abstrait $(\mathcal{D}_a, \gamma)$.

Exemple 2.28
Les fonctions γ de l'exemple 2.26 et ρ de l'exemple 2.27 sont des fonctions de concrétisation.

On s'intéresse aussi aux abstractions des éléments concrets :

Définition 2.15 (*Fonction d'abstraction*)

Nous appelons fonction d'abstraction sur un domaine abstrait $(\mathcal{D}_a, \gamma)$, une fonction croissante $\alpha : \wp(\mathbb{G}^d) \mapsto \mathcal{D}_a$.

Exemple 2.29
Nous reprenons l'exemple 2.27, la fonction α de $\wp(\mathbb{Z}^d)$ dans $\mathcal{I}(\overline{\mathbb{Z}}^d)$ définie par :

$$\alpha(C) = \prod_{i=1}^{d}[a_i, b_i],$$

où $a_i = \sup\{c_i \in \overline{\mathbb{Z}} \mid c_i \leq \pi_i(x),\ x \in C\}$ et $b_i = \inf\{d_i \in \overline{\mathbb{Z}} \mid \pi_i(x) \leq d_i,\ x \in C\}$. La fonction π_i est la fonction de projection sur la coordonnée i.

Dans la définition d'une fonction abstraction, on voit qu'une fonction d'abstraction dépend du domaine abstrait considéré. Une fonction de concrétisation est une partie intégrante du domaine abstrait, le choix de la fonction d'abstraction peut donc être conditionné par la fonction de concrétisation. En général, en interprétation abstraite, on utilise, lorsqu'elle existe, une paire particulière de fonction d'abstraction et de concrétisation :

Définition 2.16 (*Correspondance de Galois*)

Soient $(E, \preceq_E)$ et $(F, \preceq_F)$ deux ensembles ordonnés. Une correspondance de Galois entre E et F est une paire d'applications croissantes $f : E \mapsto F$ et $g : F \mapsto E$ telles que :

$$\forall\, x \in E, \forall y \in F,\ f(x) \preceq_F y \iff x \preceq_E g(y) \tag{2.2}$$

Exemple 2.30
La paire d'applications (α, γ) définies aux exemples 2.28 et 2.29 forme une correspondance de Galois.

Remarque 2.9 En théorie de Galois classique, voir [Ore44], les paires d'applications (f, g) sont appelées correspondances de Galois lorsque les applications f et g sont décroissantes c'est-à-dire inversent les ordres et que l'équivalence (2.2) est vérifiée. Une application f telle qu'il existe une application g vérifiant (2.2) est appelée application résiduable [BJ72].

Quand une fonction d'abstraction α et une fonction de concrétisation γ forment une correspondance de Galois, la meilleure abstraction d'un élément concret X est donnée par $\alpha(X)$. De plus, la fonction de concrétisation γ associée à la fonction d'abstraction α est déterminée de manière unique. Ainsi, nous pouvons construire la fonction sémantique abstraite sur un domaine abstrait, grâce à une correspondance de Galois (α, γ), où α est une fonction d'abstraction et γ une fonction de concrétisation.

Définition 2.17 (*Fonction sémantique abstraite*)

La fonction sémantique abstraite $F^\sharp : (\mathcal{D}_a)^{n+1} \mapsto (\mathcal{D}_a)^{n+1}$, est définie, pour une coordonnée $i \in \{0, \ldots, n\}$ par :

$$X \mapsto F_i^\sharp(X_0, X_1, \ldots, X_n) = \alpha(F_i^C(\gamma(X_0), \gamma(X_1), \ldots, \gamma(X_n))) \tag{2.3}$$

où α est une application d'abstraction, γ une application de concrétisation, (α, γ) forme une correspondance de Galois et F^C est la fonction sémantique concrète.

On écrira par la suite, simplement :

$$F^\sharp = \alpha \circ F^C \circ \gamma \ .$$

Exemple 2.31
La fonction sémantique abstraite dans le domaine abstrait des intervalles associée au programme de l'exemple 2.18. Soit X un vecteur d'intervalles de $\mathcal{I}(\overline{\mathbb{Z}}^6)$.

$$\begin{array}{lcl} F_0^\sharp(X) & = & \emptyset \\ F_1^\sharp(X) & = & [1,1] \\ F_2^\sharp(X) & = & [\inf(x_1^-, x_4^-), \sup(x_1^+, x_4^+)] \\ F_3^\sharp(X) & = & [x_2^-, \inf(10, x_2^+)] \\ F_4^\sharp(X) & = & [x_3^- + 1, x_3^+ + 1] \\ F_5^\sharp(X) & = & [\sup(10, x_2^-), x_2^+] \end{array}$$

Exemple 2.32
La fonction sémantique abstraite dans le domaine abstrait des intervalles associée au programme de l'exemple 2.19. Soit X un vecteur d'intervalles de $\mathcal{I}(\overline{\mathbb{Z}}^7)$.

$$\begin{array}{lcl} F_0^\sharp(X) & = & \emptyset \\ F_1^\sharp(X) & = & [0,1] \times [0,1] \\ F_2^\sharp(X) & = & [x_1^-, \inf(x_1^+, y_1^+)] \times [\sup(x_1^-, y_1^-), y_1^+] \\ F_3^\sharp(X) & = & [x_2^- + 1, x_2^+ + 1] \times [y_2^-, y_2^+] \\ F_4^\sharp(X) & = & [\sup(x_1^-, y_1^- + 1), x_1^+] \times [y_1^-, \inf(y_1^+, x^+ - 1)] \\ F_5^\sharp(X) & = & [x_4^-, x_4^+] \times [y_4^- + 1, y_4^+ + 1] \\ F_6^\sharp(X) & = & [\inf(x_2^-, x_5^-), \sup(x_2^+, x_5^+)] \times [\inf(y_2^-, y_5^-), \sup(y_2^+, y_5^+)] \end{array}$$

Nous rappelons que le but est de pouvoir calculer automatiquement une sur-approximation de la sémantique concrète. Par construction, la fonction sémantique abstraite est croissante et par définition, le domaine abstrait est un ensemble ordonné. La plus précise sur-approximation de la sémantique concrète est donnée par le plus petit point fixe de la fonction sémantique abstraite (s'il existe !). En pratique, on impose au domaine abstrait d'être un treillis complet. Par le théorème de Tarski, le plus petit point fixe de la fonction sémantique abstraite (appelé sémantique abstraite) existe.

2.3.2 Quelques domaines abstraits courants en interprétation abstraite

Deux grandes classes de domaines abstraits sont couramment utilisées :
- les *domaines non relationnels* qui sont plus simples à manipuler mais qui ne conservent pas les relations entre les variables d'un programme.
- les *domaines relationnels* plus coûteux en mémoire et en temps de calcul mais qui conservent certaines relations entre les variables.

Domaines non relationnels

Le domaine abstrait $\mathcal{D}_a$ s'écrit, dans ce cas, D^k, où k est un entier naturel non nul inférieur au nombre de variables et D est un treillis complet ou un CPO.

Nous présentons brièvement, dans cette sous-section, quelques domaines non relationnels.

Propagation de constantes Ce domaine abstrait a été introduit par Kildall [Kil73]. Ce domaine n'est, en pratique, pas utilisé pour calculer des bornes sur les variables d'un programme mais permet d'optimiser l'exécution d'un code. En effet, le but de l'analyse est de remplacer, à chaque point de contrôle, les « fausses » variables, celles qui ne vont être pas modifiées par l'instruction au point de contrôle par leurs valeurs en mémoire. Ceci simplifie l'écriture du programme et est effectué par la plupart des compilateurs actuels. Un nouvel exécutable, sans variables superflues, est ainsi créé. Les invariants numériques au point de contrôle i sont de la forme $(x_{j_1} = c^i_{j_1}, x_{j_2} = c^i_{j_2}, \ldots, x_{j_k} = c^i_{j_k})$ où k représente le nombre de *fausses* variables et $c^i_{j_l}$ une constante dans $\overline{\mathbb{G}}^d$.

Signes Ce domaine a été introduit par Cousot et Cousot [CC76]. Ce domaine est un domaine académique qui n'est pas utilisé en pratique pour calculer des invariants numériques. L'invariant cherché, pour un point de contrôle donné est le produit cartésien d'ensembles de la forme $x_j \leq 0$ ou $x_j \geq 0$. Les valeurs de la variable importe peu, seul son signe est étudié. Ce domaine est intéressant pour des raisons numériques évidentes, par exemple, pour prendre la racine carrée d'une variable, il faut vérifier que la variable soit positive pour éviter toute erreur d'exécution d'un programme. Ce domaine apparaît également en théorie des systèmes linéaires notamment pour savoir si la solution d'un système est positive [BS95].

Une concrétisation géométrique est présentée à la figure 2.9.

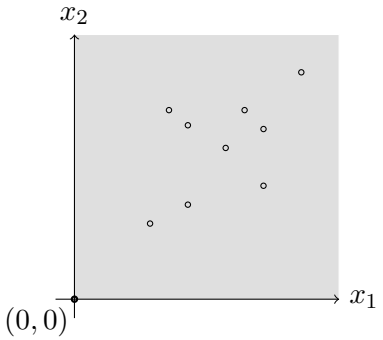


Figure 2.9 – Concrétisation géométrique d'une abstraction dans le domaine des signes

Les intervalles Le domaine des intervalles ont été introduits par Cousot et Cousot [CC76]. A chaque point de contrôle, nous cherchons les intervalles les plus précis possibles qui contiennent les valeurs possibles prises par les variables du programme. Les invariants numériques au point de contrôle i sont de la forme :

$$(x_1 \in [a_1^i, b_1^i], x_2 \in [a_2^i, b_2^i], \ldots, x_d \in [a_d^i, b_d^i])$$

où pour tout $j = 1, \ldots, d$, a_j^i et b_j^i appartiennent à $\overline{\mathbb{G}}^d$ et $a_j^i \leq b_j^i$. Les bornes des intervalles peuvent donc être infinies. Ce domaine contient le domaine introduit par Kildall, en effet, un singleton peut s'écrire $[a, b]$ avec $a = b$. Enfin, ce domaine contient aussi le domaine des signes puisque $x_j \leq 0$ équivaut à $x_i \in [0, +\infty]$ et $x_j \geq 0$ équivaut à $x_i \in [-\infty, 0]$. La réalisation géométrique d'un invariant calculé dans le domaine des intervalles est décrit à la figure 2.10.

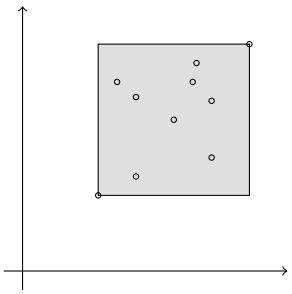


Figure 2.10 – Abstraction dans le domaine des intervalles

Le simple exemple 2.33 montre la perte de précision due à l'emploi des intervalles.

Exemple 2.33
Considérons le programme suivant :

```
x=[0,10];   [1]
y=x;        [2]
```

Une analyse par intervalles retourne $x \in [0, 10], y \in [-\infty, +\infty]$ au point de contrôle [1] et $x \in [0, 10], y \in [0, 10]$ au point de contrôle [2].

Dans l'exemple 2.33, la relation, pourtant forte, entre la variable x et la variable y disparaît au moment du calcul de l'invariant. Les domaines abstraits relationnels comblent, en partie, ces pertes de relations entre les variables.

Domaines relationnels

Polyèdres Ce domaine est introduit par Cousot et Halbwachs [CH78]. Les invariants numériques sont des polyèdres convexes fermés, ils sont par conséquent de la forme $\{X \in \mathbb{R}^d \mid AX \leq B\}$, où X est l'abstraction des variables du programme, A est une matrice de taille $d \times d$ et B un vecteur de $\mathbb{R}^d$. Les formes linéaires définissant les faces de ces polyèdres sont construites à partir des relations linéaires entre les variables du programmes. Cependant le nombres faces de ces polyèdres grandit de manière exponentielle par rapport au nombre de relations linéaires entre les variables. Un invariant calculé dans le domaine des polyèdres est décrit par la figure 2.11.

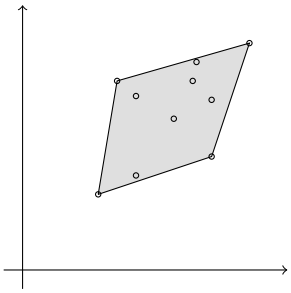


Figure 2.11 – Concrétisation géométrique d'une abstraction dans le domaine des polyèdres

Zones et Octogones Miné [Min04] a partiellement répondu au problème de complexité de l'analyse de programme par les polyèdres tout en conservant certaines relations entre les variables. En effet, dans le domaine des zones, les invariants numériques sont de la forme $X_i - X_j \leq c_{ij}$. Dans le domaine des octogones, les invariants prennent la forme du type $\pm X_i \pm X_j \leq c_{ij}$. La figure 2.12 représente un invariant numérique calculé dans le domaine des zones.

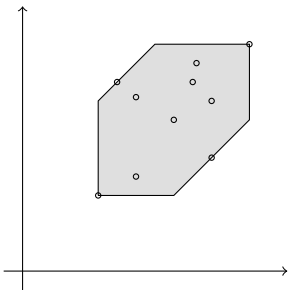


Figure 2.12 – Concrétisation géométrique d'une abstraction dans le domaine des zones

Gabarits (templates) linéaires Sankaranarayanan et al [SSM05] ont développé une méthode intermédiaire entre les polyèdres et les zones et octogones de Miné. Dans les zones et octogones, seules les différences entre les variables est analysées. Nous pouvons voir ces différences comme des formes linéaires. Les gabarits linéaires sont une généralisation des zones et des octogones. On fixe des formes linéaires avant l'analyse du programme. Les invariants en chaque point de contrôle du programme sont décrits par un polyèdre. Le nombre de faces n'explose pas puisque fixé par avance. Le polyèdre de la figure 2.13 est la concrétisation géométrique d'un invariant calculé dans le domaine des gabarits linéaires.

On remarque que les figures 2.11 et 2.13 sont similaires, dans le domaine des polyèdres, les faces du polyèdres sont calculées au cours d'analyse alors que, pour les gabarits linéaires, on obtient la figure 2.13 en fixant par avance les quatre faces du polyèdres.

Domaines des formes affines Dans l'article [GP], l'invariant numérique est un invariant de type fonctionnel. L'invariant numérique est une abstraction des relations entre

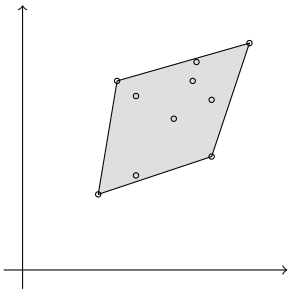


Figure 2.13 – Concrétisation géométrique d'une abstraction dans le domaine des gabarits linéaires

les entrées du programme et ses sorties. Chaque variable du programme est une combinaison linéaire de "symboles de bruit" (c'est-à-dire l'intervalle $[-1, 1]$). Les relations entre les variables du programme est mise en évidence par le fait qu'elles ont des symboles de bruit communs dans les combinaisons linéaires les représentant. La concrétisation géométrique des invariants est un zonotope c'est-à-dire une somme de Minkowski de segments de $\mathbb{R}^d$. Un exemple de zonotope est représenté par la figure 2.14.

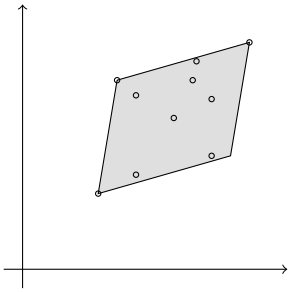


Figure 2.14 – Abstraction dans le domaine des formes affines

Ellipsoïdes Feret [Fer04] étudie des systèmes linéaires d'ordre 2 stables. L'invariant numérique est donné par l'ellipse invariante associée à une fonction de Lyapunov du système linéaire. L'invariant numérique est donc de la forme $\alpha X^2 + \beta Y^2 + \gamma XY \leq \delta$. La figure 2.15 représente l'abstraction du nuage de points considéré dans les figures précédentes.

D'un point de vue expressivité, le domaine des signes est le domaine le moins expressif puisque l'on s'intéresse qu'aux signes des variables. On peut voir les invariants comme le produit d'intervalles de la forme $[0, +\infty]$ alors que le domaine des intervalles contient tout type d'intervalles. Par ailleurs, les intervalles sont des zones particulières et donc le domaine des zones est plus expressif que celui des zones. Dans le domaine des octogones, Miné a ajouter deux types de relations par rapport au domaine des zones, le domaine des octogones est donc plus expressif que celui des zones. En outre, le domaines des zontopes contient tous les polyèdres à symétrie centrale dont les faces sont encore à symétrie centrale ; une propriété qui est vérifiée par les octogones et les zones, le domaine des zonotopes est plus expressif que celui des octogones. Le domaine des gabarits linéaires de Sankaranarayanan permet de

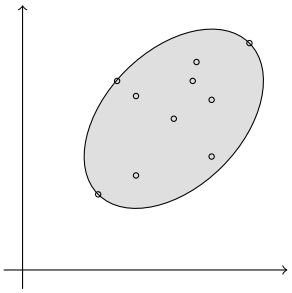


Figure 2.15 – Concrétisation géométrique d'une abstraction dans le domaine des ellipsoïdes

considérer un nombre fini de formes linéaires et permet donc de représenter un zonotope.

CHAPITRE 3

ALGORITHMES DE POINT FIXES EN INTERPRÉTATION ABSTRAITE

Dans le chapitre précédent, nous avons construit les équations de point fixe à résoudre pour trouver des invariants numériques. La méthode classiquement utilisée pour résoudre ces équations, découle directement du théorème de Kleene 2.2. Il s'agit d'une méthode itérative, appelée itération de Kleene, pour laquelle l'itérée est la fonction sémantique abstraite. La fonction sémantique abstraite est essentiellement constituée de min, de max et d'opérations affines, dans le cas où le programme ne contient pas de division ou de multiplication de variables. Ceci nous permet de faire une analogie avec les opérateurs de programmation dynamique de la théorie des jeux. Ainsi, l'itération de Kleene s'identifie à l'itération sur les valeurs. Cette méthode est connue pour être souvent longue. En interprétation abstraite, l'itération de Kleene est couplée avec des techniques d'accélération de convergence pour assurer une convergence en temps fini.

En théorie des jeux, une alternative à l'itération sur les valeurs est l'itération dans l'espace des stratégies. Contrairement à l'itération sur les valeurs qui suit la dynamique de l'opérateur, l'itération sur les politiques explore la structure des espaces d'actions. Dans le cas des jeux finis, l'itération sur les politiques converge en temps fini. Dans la section 3.1, nous rappelons l'itération de Kleene ainsi que les techniques d'élargissement et de rétrécissement. Dans la section 3.2, nous présentons l'itération sur les politiques : nous commençons par des rappels sur les jeux stochastiques et nous montrons comment l'algorithme de Hoffman et Karp [HK66] permet de résoudre des problèmes de jeux stochastiques. Nous terminons cette section par l'extension de l'itération sur les politiques pour l'interprétation abstraite.

3.1 Itération de Kleene

L'application F est une fonction sémantique abstraite associée à un domaine abstrait. Nous supposons que le domaine abstrait est un treillis complet. Par construction, l'application F est croissante et donc par le théorème de Tarski, possède un plus petit point fixe. Dans cette section, nous commençons par rappeler à la sous-section 3.1.1 l'itération de Kleene puis dans la sous-section 3.1.2, nous rappelons les techniques d'accélération de convergence.

3.1.1 Itération de Kleene

L'itération de Kleene découle directement du théorème de Kleene 2.2. Le plus petit point fixe de F est exactement $\sqcup_{n\in\mathbb{N}} F^n(\bot)$ lorsque F est Scott-continue. L'algorithme 1 reprend le principe du théorème de Kleene pour calculer le plus petit point fixe de fonctions continues.

Algorithme 1: Itération de Kleene

$X_0 = \bot,\ n = 0;$
tant que $X_n \neq F(X_n)$ **faire**
  $X_{n+1} = X_n \sqcup F(X_n);$
  $n = n + 1;$

Dans l'algorithme 1, la fonction F n'est, en pratique, pas exactement la fonction sémantique abstraite construite au chapitre 2 mais la fonction sémantique abstraite après quelques réductions : à la manière de la propagation des constantes, on réduit le nombre de variables, en remplaçant les variables "constantes" par leur valeur et on remplace dès que possible, les variables par leurs affectations. Par ailleurs, la composante F_0 est un ensemble de valeurs initiales qui ne dépend pas du programme analysé : nous ne l'incluons pas dans notre équation de point fixe. La fonction sémantique abstraite réduite demeure une fonction croissante.

Exemple 3.1
La figure 3.1 décrit la fonction sémantique abstraite réduite de l'exemple 2.31.

$$\begin{array}{lcl} F_1(X) & = & [1,1] \\ F_2(X) & = & [\inf(1, x_2^- + 1), \sup(1, \inf(10, x_2^+) + 1)] \\ F_3(X) & = & [x_2^-, \inf(10, x_2^+)] \\ F_4(X) & = & [x_2^- + 1, \inf(10, x_2^+) + 1] \\ F_5(X) & = & [\sup(11, x_2^-), x_2^+] \end{array}$$

Figure 3.1 – Fonction sémantique abstraite après réductions

Exemple 3.2
L'itération de Kleene sur la fonction sémantique abstraite réduite du programme de l'exemple 2.24 dans le domaine des intervalles est décrite par la figure 3.2.

De plus, l'itération de Kleene définie à l'algorithme 1 est la version théorique de base de l'algorithme. En pratique, il existe des méthodes plus efficaces qui utilise les structures syntaxiques du programme à analyser. Pour plus de détails sur ces techniques, le lecteur pourra consulter [Bou93] et [Mar98].

3.1.2 Accélération de convergence

L'itération de Kleene est une itération lente, la suite des itérées est fortement dépendante de la dynamique du programme. De plus, le fait de considérer des fonctions croissantes non nécessairement continues peut amener un nombre ordinal d'itération pour calculer le plus petit point fixe. En pratique, la convergence en temps fini vers le plus petit point fixe n'est pas

$$\begin{array}{lcl} X_1^1 = [1,1] \\ X_2^1 = [1,1] \\ X_3^1 = \emptyset \\ X_4^1 = \emptyset \\ X_5^1 = \emptyset \end{array} \quad \begin{array}{lcl} X_1^2 = [1,1] \\ X_2^2 = [1,2] \\ X_3^2 = [1,1] \\ X_4^2 = [2,2] \\ X_5^2 = \emptyset \end{array} \quad \begin{array}{c} \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{array} \quad \begin{array}{c} \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{array} \quad \begin{array}{lcl} X_1^{11} = [1,1] \\ X_2^{11} = [1,11] \\ X_3^{11} = [1,10] \\ X_4^{11} = [2,11] \\ X_5^{11} = \emptyset \end{array}$$

$$\begin{array}{lcl} X_1^{12} = [1,1] \\ X_2^{12} = [1,11] \\ X_3^{12} = [1,10] \\ X_4^{12} = [2,11] \\ X_5^{12} = [11,11] \end{array}$$

Figure 3.2 – Itérées successives de l'itération de Kleene

assurée. En raison de ces trois problèmes, l'itération de Kleene nécessite l'ajout de techniques d'accélération de convergence.

L'accélération de convergence se décompose en deux étapes, la première est une technique d'élargissement. Si la suite des itérées est croissante, la technique d'élargissement (en anglais *widening*) introduite dans [CC92b] consiste à donner une extrapolation de la limite. La deuxième étape est une technique de rétrécissement. Il se peut que l'extrapolation de la limite donnée par le *widening* soit trop large, la technique de rétrécissement permet de réduire la « largeur » de l'invariant. Pour définir ces deux techniques nous nous munissons d'un treillis $(E, \preceq)$.

Définition 3.1 (*Opérateur d'élargissement*)

L'opérateur $\triangledown : E \times E \mapsto E$ est un opérateur d'élargissement si :

1. $\forall\, x, y \in E,\ x \preceq x \triangledown y$;
2. $\forall\, x, y \in E,\ y \preceq x \triangledown y$;
3. Pour toute suite croissante $(x_n)_{n\in\mathbb{N}}$ dans E, la suite définie par $y_0 := x_0$ et $y_{n+1} := y_n \triangledown x_{n+1}$ est stationnaire.

Exemple 3.3

Dans les intervalles l'opérateur défini par :

$$[a,b] \triangledown [c,d] = [e,f] \text{ avec } e = \begin{cases} -\infty & \text{si } c < a \\ a & \text{sinon} \end{cases} \text{ et} f = \begin{cases} +\infty & \text{si } b < d \\ b & \text{sinon} \end{cases}$$

Exemple 3.4 (Elargissements avec seuils)

Ce type d'élargissement plus fin est décrit dans [BCC+02]. Dans un intervalle, on peut définir un *widening* plus fin. On se donne un ensemble T fini de nombres réels et on définit $\triangledown_T$ par :

$$[a,b] \triangledown_T [c,d] = \left[\begin{cases} a & \text{si } a \leq c \\ \max\{t \in T \mid t \leq c\} & \text{sinon} \end{cases}, \begin{cases} b & \text{si } d \leq b \\ \min\{t \in T \mid d \leq t\} & \text{sinon} \end{cases} \right]$$

Définition 3.2 (*Opérateur de rétrecissement*)

L'opérateur $\triangle$: $E \times E \mapsto E$ est un opérateur de rétrécissement si :

1. $\forall\, x, y \in E,\ y \preceq x \implies y \preceq x \triangle y \preceq x$;
2. Pour toute suite décroissante $(x_n)_{n\in\mathbb{N}}$ dans E, la suite définie par $y_0 := x_0$ et $y_{n+1} := y_n \triangle x_{n+1}$ est stationnaire.

Exemple 3.5

Dans les intervalles, l'opérateur défini par :

$$[a,b] \triangle [c,d] = [e,f] \text{ avec } e = \begin{cases} c & \text{si } a = -\infty \\ a & \text{sinon} \end{cases} \text{ et } f = \begin{cases} c & \text{si } b = +\infty \\ b & \text{sinon} \end{cases}$$

est un opérateur de rétrécissement.

Une manière simple pour augmenter la précision de l'invariant après un opérateur d'élargissement est d'effectuer des itérations descendantes. En effet, la fonction sémantique abstraite (dans ce chapitre F) est une fonction croissante et si, un vecteur $X \in E$ vérifie $F(X) \preceq X$ alors la suite $(F^k(X))_{k\in\mathbb{N}}$ est décroissante. Cette méthode est appelée *itérations descendantes*.

Exemple 3.6

Nous reprenons l'exemple 3.2 et nous effectuons un *widening* dès la deuxième itération. La deuxième itérée de l'itération de Kleene et son image sont :

$$\begin{array}{lcl} X_1^2 &=& [1,1] \\ X_2^2 &=& [1,2] \\ X_3^2 &=& [1,1] \\ X_4^2 &=& [2,2] \\ X_5^2 &=& \emptyset \end{array} \qquad \begin{array}{lcl} F_1(X^2) &=& [1,1] \\ F_2(X^2) &=& [1,3] \\ F_3(X^2) &=& [1,2] \\ F_4(X^2) &=& [2,3] \\ F_5(X^2) &=& \emptyset \end{array}$$

En pratique, on prend les vecteurs précédemment calculés au cours de l'itération, par exemple, pour calculer $F_5(X^2)$, le vecteur X^2 que l'on considère est $[1,+\infty[$ et ainsi, on trouve :

$$\begin{array}{lcl} X_1^2 \triangledown F_1(X^2) &=& [1,1] \\ X_2^2 \triangledown F_2(X^2) &=& [1,+\infty] \\ X_3^2 \triangledown F_3(X^2) &=& [1,+\infty] \\ X_4^2 \triangledown F_4(X^2) &=& [2,+\infty] \\ X_5^2 \triangledown F_5(X^2) &=& [11,+\infty] \end{array}$$

En effectuant un *narrowing* à l'itération suivante, on obtient, en prenant $F_5(X^3) = [11,11]$:

$$\begin{array}{lcccl} X_1^3 \triangle F_1(X^3) &=& [1,1] \triangle [1,1] &=& [1,1] \\ X_2^3 \triangle F_2(X^3) &=& [1,+\infty] \triangle [1,11] &=& [1,11] \\ X_3^3 \triangle F_3(X^3) &=& [1,+\infty] \triangle [1,10] &=& [1,10] \\ X_4^3 \triangle F_4(X^3) &=& [2,+\infty] \triangle [2,11] &=& [2,11] \\ X_5^3 \triangle F_5(X^3) &=& [11,+\infty] \triangle [11,11] &=& [11,11] \end{array}$$

Puis avec une itération de Kleene supplémentaire on trouve :

$$\begin{array}{rcl} X_1^4 & = & [1,1] \\ X_2^4 & = & [1,11] \\ X_3^4 & = & [1,10] \\ X_4^4 & = & [2,11] \\ X_5^4 & = & [11,11] \end{array}$$

On retrouve exactement le point fixe solution de l'exemple 3.2.

La combinaison de ces deux techniques peut dégrader la qualité du point fixe généré par l'itération de Kleene accélérée, il se peut que, dans certains cas, l'algorithme retourne un post-point fixe ($F(X) \preceq X$) ou que le point fixe retourné par l'algorithme ne soit pas le plus petit point fixe. De plus, d'un point de vue pratique, les opérateurs de *widening* et *narrowing* sont difficiles à mettre en oeuvre. La difficulté augmente avec la complexité des opérations de treillis et les opérations de *clôture* utilisées.

Exemple 3.7
Dans cet exemple, nous nous intéressons à une implémentation du filtre de Butterworth d'ordre un [But30] [1].

```
x1 =0;
y  =0;
assume 1<=u<=2; [1]
while [2] (true){
  xn1 = 0.9048*x1+0.9524*u;
  y = 0.09524*x1+0.4762*u;
  x1 = xn1;[3]
  }
```

$$\begin{array}{rcl} X_1 & = & \{0\} \times \{0\} \\ X_2 & = & X_1 \cup X_3 \\ X_3 & = & T(X_2) \end{array}$$

Il est facile de voir que les valeurs des variables $x1$ et y sont toujours positives et comme initialement, $x1$ et y sont nuls, la borne inférieure des valeurs prises par $x1$ et y est 0. Nous nous utilisons la méthode d'interprétation abstraite pour analyser les bornes supérieures des variables $x1$ et y. La fonction T à prendre en compte est donc la fonction qui à $(x1, y)$ associe :

$$\begin{pmatrix} 0.9048 & 0 \\ 0.09524 & 0 \end{pmatrix} \begin{pmatrix} x1 \\ y \end{pmatrix} + \begin{pmatrix} 0.9524 * u \\ 0.4762 * u \end{pmatrix}$$

Si on effectue une itération de Kleene, on trouve en 327 itérations au point de contrôle 2 : $0 \leq x1 \leq 20.008$ et $0 \leq y \leq 2.0008$.

Considérons maintenant un élargissement avec seuil uniquement sur la variable $x1$ en prenant $T = \{1, 2, 4, 8, 16, 32\}$. On trouve en remplaçant l'union par l'opérateur d'élargissement avec seuil : $0 \leq x1 \leq 32$ et $0 \leq y$. Après 40 itérations descendantes, on obtient : $0 \leq x1 \leq 20.4966$ et $0 \leq y \leq 2.0522$.

Dans cet exemple, la méthode de Kleene avec élargissement combiné avec des itérations descendantes ne permet pas de retrouver le plus petit point fixe.

1. Je remercie Olivier Bouissou qui m'a proposé cet exemple

On peut citer également, le travail de Leroux et Sutre [LS07a] sur les techniques d'accélération pour les problèmes de contrôle de flot de données. Le travail [LS07b] décrit des techniques d'accélération pour des programmes dont les tests sont des polyèdres convexes et les affectations sont des translations.

3.2 Itération sur les politiques

L'itération sur les politiques est un algorithme utilisé en théorie du contrôle et en théorie des jeux. Pour décrire cet algorithme, nous présentons quelques types de jeux et la résolution du problème de calcul de la valeur du jeu par itération sur les politiques.

3.2.1 Jeux stochastiques finis à somme nulle

Modèle originel de Shapley

Nous reprenons le cadre de Shapley [Sha53] qui a été le précurseur des jeux stochastiques. Le jeu stochastique considéré par Shapley est un jeu avec des probabilités d'arrêt non nulles à chaque état du jeu. Gillette [Gil57] a, par la suite, introduit un modèle équivalent avec des probabilités nulles d'arrêts mais avec taux d'escompte.

Le jeu se déroule sur un espace d'états finis $S = \{1, \ldots, n\}$. le jeu se déroule avec deux joueurs : le joueur $J1$ qui est un minimiseur et le joueur $J2$ qui est un maximiseur. A chaque état $s \in S$, le joueur $J1$ choisit une action $a \in A(s)$ et le joueur $J2$ choisit une action $b \in B(s)$. Dès que, pour l'état s, les actions $a \in A(s)$ et $b \in B(s)$ ont été annoncées, le jeu passe de l'état s à l'état t avec une probabilité $P_{st}(a, b)$ et le joueur $J1$ paie $R_s(a, b)$ au joueur $J2$. Le jeu est à information complète : les deux joueurs observent les actions possibles de l'autre ainsi que ses possibilités d'action, les deux joueurs observent également les revenus de leurs actions. Le jeu se répète infiniment (horizon infini) et le joueur $J1$ cherche à minimiser le paiement final tandis que le joueur $J2$ cherche à maximiser ce paiement que nous définirons par la suite.

Jeux stochastiques à information parfaite

Le modèle du jeu à information parfaite est un cas particulier de jeu stochastique : les joueurs connaissent les décisions prises par l'autre joueur avant de jouer. Ainsi un joueur peut donc utiliser cette information pour prendre ses décisions. Un jeu à information parfaite est donc séquentiel au sens où une action $b \in B(s)$ peut être choisie dans un ensemble $b \in B(s, a)$ ou a désigne une action du joueur $J1$ à l'état s. Mathématiquement, la particularité de ce type de jeu vient des ensembles d'actions des joueurs. Quitte à ajouter des états, on peut supposer qu'à chaque état s, au moins l'un des ensembles $A(s)$ ou $B(s)$ est un singleton. Si les deux ensembles $A(s)$ et $B(s)$ sont des singletons, l'état s n'est contrôlé par aucun des deux joueurs. L'espace d'états peut ainsi être décomposé en trois ensembles disjoints S_0, S_1 et S_2, ces ensembles représentant respectivement les états non contrôlés, les états contrôlés par $J1$ et les états contrôlés par $J2$. Nous reviendrons sur les jeux à information parfaite à la fin de cette thèse.

Ces jeux apparaissent souvent en informatique, on peut citer Gimbert [Gim06] pour son travail sur les jeux stochastiques à information parfaite pour des jeux d'atteignabilité et les jeux positionnels.

Modèle à un joueur

Le modèle du jeu à un seul joueur est appelé *processus de décision markovien* (MDP en anglais). Le modèle à un joueur est le cas particulier d'un jeu stochastique à information parfaite où les ensembles d'actions $B(s)$ sont des singletons pour tous les états $s \in S$, le deuxième joueur ne contrôle aucun état, ne prend pas de décision. Pour plus de détails sur les problèmes de contrôle stochastique, le lecteur pourra consulter [Put94].

Politiques

En théorie des jeux, il peut être utile de jouer une probabilité sur les actions plutôt que de jouer une action particulière. On définit ainsi les actions mixtes :

Définition 3.3 (*Action mixte*)

Soit $s \in S$. Une action mixte x, pour le joueur $J1$, est une probabilité sur l'ensemble $A(s)$ des actions disponibles à l'état s. Nous notons $\Delta(A(s))$ l'ensemble des actions mixtes sur $A(s)$. On définit de même, les actions mixtes du joueur $J2$ et on note $\Delta(B(s))$ l'ensemble des actions mixtes sur $B(s)$.

Remarque 3.1 (Action pure) Pour chaque état, l'ensemble des actions mixtes contient l'ensemble des actions disponibles (que l'on appelle actions pures). En effet, une masse de Dirac est une probabilité qui charge un unique point et donc une masse de Dirac est une action mixte qui charge une unique action et donc est assimilée à une certaine action.

Pour un couple d'actions mixtes $(x, y) \in \Delta(A(s)) \times \Delta(B(s))$, on associe de nouveaux paiements et probabilités de transition :

$$\begin{aligned} \tilde{R}_s(x,y) &= \sum_{l=1}^{\mathrm{card}(A(s))} \sum_{m=1}^{\mathrm{card}(B(s))} x_l^i y_m^i R_s(a_l, b_m) \\ \tilde{P}_{st}(x,y) &= \sum_{l=1}^{\mathrm{card}(A(s))} \sum_{m=1}^{\mathrm{card}(B(s))} x_l^i y_m^i P_{st}(a_l, b_m) \ . \end{aligned}$$

Les vecteurs $\tilde{R}_s(x, y)$ et $\tilde{P}_{st}(x, y)$ sont appelés extension mixte.

On introduit maintenant la notion de politique qui sont des vecteurs d'actions. Une politique permet de regarder les décisions des joueurs à chaque date sans se focaliser directement sur l'état courant du jeu.

Définition 3.4 (*Politiques*)

Une politique pour le joueur $J1$ est une application $\pi : S \mapsto \mathbf{A} = \delta(\bigcup_{s \in S} A(s))$ telle que $\pi(s) := \pi_s(a) = 0$ si $a \notin A(s)$, $\pi_s(a) \geq 0$ pour tout $a \in A(s)$ et $\sum_{a \in A(s)} \pi_s(a) = 1$.

Une politique peut être vue comme un profil d'actions pour les joueurs. On note Π^1 l'ensemble des politiques du joueur $J1$. De même, on définit et on note Π^2 l'ensemble des politiques du joueur $J2$. On note $\mathbf{B}$ l'union de toutes les actions disponibles de $J2$. Une politique π est dite *pure* si, pour tout $s \in S$, $\pi(s)$ est une masse de Dirac c'est-à-dire, il existe $a \in A(s)$, tel que $\pi_s(a) = 1$ et $\pi_s(a') = 0$ pour tout $a' \in A(s)$ différent de a.

Stratégies

Puisque le jeu est répété, nous nous intéressons à une suite de décisions. Les décisions prises peuvent dépendre du temps ou de la suite des décisions prises précédemment. Les deux joueurs savent que le jeu débute à l'état initial s^0 et à chaque date k, $J1$, $J2$ choisissent respectivement une politique π^k et π'^k et des actions a^k et b^k sont tirées.

Nous introduisons ainsi la notion d'histoire : pour une date $k \geq 1$, l'histoire passée est $h_k = ((s^0, a^0, b^0), (s^1, a^1, b^1), \ldots, (s^{k-1}, a^{k-1}, b^{k-1}, s^k)$ où s^j désigne la suite (processus aléatoire) d'état du jeu et a^j (resp. b^j) la suite (processus aléatoire) d'actions du joueur $J1$ (resp. $J2$). On note H_k l'ensemble des histoires de longueur k, qui est formellement représenté par le produit cartésien $(S \times A \times B)^{k-1} \times S$ puis $H = \bigcup_{k \in \mathbb{N}} H_k$ l'ensemble de toutes les histoires de longueur finie et enfin H_∞ l'ensemble $(S \times A \times B)^{\mathbb{N}^*}$. On peut maintenant définir les paiements à partir de cette espérance.

Définition 3.5 (*Stratégie*)

Une stratégie pour le joueur $J1$ (resp. $J2$) est une application σ (resp. τ) de H dans Π^1 (resp. Π^2).

S'il existe $\pi \in \Pi^1$, tel que pour tout $t \geq 1$, pour tout $h \in H_t$, $\sigma(h_t) = \pi$, on dit que la stratégie est *stationnaire*. Jouer une stratégie stationnaire signifie que le joueur choisit sa politique uniquement par rapport à l'état et indépendamment du temps.

On note $\mathcal{S}$ l'ensemble des stratégies de $J1$ et $\mathcal{T}$ l'ensemble des stratégies de $J2$. D'après le théorème de Ionescu-Tulcea, un état initial s et un couple de stratégies (σ, τ) définit une unique mesure de probabilité $P_s(\sigma, \tau)$ sur H_∞ muni de la tribu produit, on note $E_s^{\sigma,\tau}$ l'espérance associée.

Paiements

En horizon infini, le coût est actualisé, à chaque date k, le coût est multiplié par un facteur $\beta \in]0, 1[$ appelé *taux d'escompte*. Le taux d'escompte β permet de donner plus d'importance aux actions présentes que futures, d'un point de vue mathématique, il permet d'assurer l'existence et l'unicité de la solution. D'autres types de paiement existent mais nous nous concentrons uniquement sur les paiements à taux d'escomptes qui ont été initialement considérés par Shapley.

Définition 3.6 (*Paiement escompté*)

Pour un couple de stratégie (σ, τ), on définit le paiement escompté comme l'espérance de la série des paiements quotidiens actualisés par un taux d'escompte β sous la probabilité définie par le couple (σ, τ). En notant $\sigma = (\pi^1, \pi^2, \ldots)$ et $\tau = (\rho^1, \rho^2, \ldots)$.

$$E_s^{\sigma,\tau}\left(\sum_{k=0}^{\infty} \beta^{k+1} \tilde{R}_{s^k}(\pi^{k+1}(s^k), \rho^{k+1}(s^k))\right). \tag{3.1}$$

Le paiement escompté est donc une fonction qui dépend de l'état initial du jeu.

Remarque 3.2 Comme nous le disions plus tôt, il existe d'autres fonction de paiement, comme le paiement moyen, on s'intéresse dans ce cas au paiement moyen "quotidien". Maitra et Sudderth [MS96, Chapter 7] ont aussi exprimé la valeur à partir de paiements lim sup, où

la série de 3.1 est remplacé par la lim sup des paiements instantanés. Le modèle de Maitra et Sudderth ne possède pas de taux d'escompte.

Le but des deux joueurs est de minimiser le paiement escompté pour le joueur $J1$ et de le maximiser pour le joueur $J2$. Le joueur $J1$ évalue donc le paiement grâce à la fonction inf-sup :

$$\overline{V_s} = \inf_{\sigma} \sup_{\tau} E^{\sigma,\tau} \left(\sum_{k=0}^{\infty} \beta^{k+1} \tilde{R}_{s^k}(\pi^{k+1}(s^k), \rho^{k+1}(s^k)) \right). \tag{3.2}$$

Le joueur $J2$ évalue donc le paiement grâce à la fonction sup-inf :

$$\underline{V_s} = \sup_{\tau} \inf_{\sigma} E^{\sigma,\tau} \left(\sum_{k=0}^{\infty} \beta^{k+1} \tilde{R}_{s^k}(\pi^{k+1}(s^k), \rho^{k+1}(s^k)) \right). \tag{3.3}$$

On a toujours $\underline{V_s} \leq \overline{V_s}$ et on dit que *le jeu possède une valeur* lorsque $\underline{V_s} = \overline{V_s} = v_s$, indépendamment de l'état i initial et le vecteur v est appelé *valeur du jeu*. Shapley dans [Sha53, Theorem 3] a montré que les jeux stochastiques ont une valeur et a donné une méthode itérative pour calculer la valeur d'un jeu stochastique à paiement escompté.

On commence par définir l'opérateur de Shapley :

Définition 3.7 (*Opérateur de Shapley*)

Soit $\beta \in [0, 1[$. L'opérateur de Shapley (escompté) est l'opérateur F de $\mathbb{R}^n$ dans $\mathbb{R}^n$ défini par :

$$\begin{aligned} F_i(v) &= \min_{x \in \Delta(A(i))} \max_{y \in \Delta(B(i))} \tilde{R}_i(x,y) + \beta \sum_{j=1}^{n} \tilde{P}_{ij}(x,y) v_j \\ &= \max_{y \in \Delta(B(i))} \min_{x \in \Delta(A(i))} \tilde{R}_i(x,y) + \beta \sum_{j=1}^{n} \tilde{P}_{ij}(x,y) v_j \end{aligned} \tag{3.4}$$

L'égalité de l'équation (3.4) lorsqu'on intervertit le min et le max n'est pas triviale et découle d'un théorème de minmax dû à von Neumann [vN28]. De plus, le théorème affirme que les min et max sont atteints, ce qui signifie en théorie des jeux que les joueurs ont des actions optimales en tout état. Par ailleurs, ce théorème est valide uniquement pour les *actions mixtes*, ce n'est plus vrai quand les joueurs choisissent des actions pures.

Remarque 3.3 L'interversion du min et du max préserve la valeur du minmax, on dit que le jeu en un coup a une *valeur*.

Théorème 3.1 (*Shapley [Sha53]*)

La valeur d'un jeu stochastique (escompté) est l'unique point fixe de l'opérateur de Shapley 3.7.

Le théorème de Shapley permet également de conclure que les joueurs ont des stratégies stationnaires optimales puisqu'il suffit de choisir les politiques qui atteignent le min et le max pour $v \in \mathbb{R}^d$ tel que $F(v) = v$ et choisir ces politiques à chaque date pour obtenir un paiement final optimal.

L'opérateur de Shapley est l'extension naturelle de l'opérateur de programmation dynamique défini en 1952 par Bellman [Bel52] pour les problèmes de contrôle stochastique. Une

version plus détaillée de ses travaux sur la théorie de la programmation dynamique se trouve dans [Bel57]. Toutefois, dans les problèmes de contrôle stochastique, dans les modèles à taux d'escompte, le joueur peut jouer de manière optimale en ne sélectionnant que des actions pures. L'équivalent de l'opérateur de Shapley pour les problèmes de contrôle stochastique porte le nom d'*opérateur de programmation dynamique* qui est défini pour chaque état $i \in S$ par :

$$v \mapsto F_i(v) = \left(\min_{x \in A(i)} R_i(x) + \beta \sum_{j=1}^{n} P_{ij}(x) v_j \right) \quad . \tag{3.5}$$

Le théorème 3.1 de Shapley est également une généralisation du principe de programmation dynamique de Bellman qui affirme que la valeur du problème de contrôle stochastique est l'unique point fixe de l'opérateur de programmation dynamique (3.5).

Dans les modèles de jeux stochastiques en information parfaite les joueurs ont également des stratégies optimales pures, l'opérateur de Shapley est donc de la forme :

$$v \mapsto F_i(v) = \min_{x \in A(i)} \max_{y \in B(i)} R_i(x, y) + \beta \sum_{j=1}^{n} P_{ij}(x, y) v_j \tag{3.6}$$

sachant que $A(i)$ ou $B(i)$ est un singleton.

Dans tous les cas, un min et un max (qui signifie que la borne inférieure et la borne supérieure sont atteintes) apparaissent en raison de la finitude des espaces d'actions à chaque état. On aurait pu également mentionner les jeux à espaces d'actions compacts avec des fonctions de paiements et des transitions de probabilités semi-continues par rapport aux espaces d'actions. Le théorème de minmax qui existe à ce niveau de généralité est dû à Sion [Sio58].

3.2.2 Calcul de la valeur d'un jeu stochastique

Soit $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ une application possédant un unique point fixe telle que :

$$F = \inf_{\pi \in \Pi} f^{\pi} \tag{3.7}$$

où Π est un espace fini assimilé à l'espace des politiques et f^{π} sont des applications croissantes de $\mathbb{R}^n$ dans $\mathbb{R}^n$, possédant un unique point fixe. Puisque Π désigne un espace de politiques alors $\Pi = \Delta(\bigcup_i A(i))$ où $A(i)$ désigne les actions disponibles à l'état i, de plus, $\pi \in \Pi$ implique que $\pi_i \in \Delta(A(i))$, et donc de manière équivalente, on peut réécrire (3.7) :

$$F_i = \inf_{\pi \in \Pi} f_i^{\pi_i} \tag{3.8}$$

Dans le cadre du contrôle stochastique, F est l'opérateur de programmation dynamique et les fonctions f^{π} sont les applications de coordonnées : $v \mapsto R^{\pi} + \beta P^{\pi} v$, et dans le cadre des jeux stochastiques, F est l'opérateur de Shapley et les fonctions f^{π} sont les applications de coordonnée : $v \mapsto \max_{b \in \Delta(B(i))} R_i^{\pi(i)b} + \beta P_i^{\pi(i)b} v$, on peut remplacer $\Delta(A(i))$ et $\Delta(B(i))$ par $A(i)$ et $B(i)$ pour les jeux à information parfaite.

Algorithme de Shapley et itérations sur les valeurs

L'opérateur de Shapley est un opérateur contractant au sens strict pour la norme infinie c'est-à-dire $\|F(x) - F(y)\| \leq \beta\|x - y\|$ où β est le taux d'escompte et donc appartient à $[0, 1[$. Ceci implique, par le théorème de point fixe de Banach, que F admet un unique point fixe que l'on note v_∞. De plus, ce point fixe se calcule en itérant l'opérateur de Shapley. Soit $\varepsilon > 0$ petit. On déduit naturellement un algorithme appelé algorithme de Shapley :

Algorithme 2: Algorithme de Shapley

début
  Choisir $k = 0$, v_0;
  tant que $\|v_{k+1} - v_k\|_\infty \geq \varepsilon$ **faire**
    $v_{k+1} = F(v_k)$;
    $k = k + 1$;
Choisir pour tout i, $\pi(i) \in \arg\min_{a \in A(i)} f_i^a(v_\infty)$;

L'algorithme de Shapley est l'extension de l'itération sur les valeurs pour les problèmes de contrôle stochastique : pour l'itération sur les valeurs, l'opérateur F est l'opérateur de programmation dynamique. L'opérateur de programmation dynamique ayant les mêmes propriétés de contraction, l'itération sur les valeurs calcule son unique point fixe.

Le temps de calcul dépend très fortement de la fonction valeur initiale. L'algorithme est efficace si on a déjà "une estimation" de la fonction valeur. En pratique, l'opérateur de Shapley est lent et coûteux puisqu'il faut résoudre un jeu matriciel (les espaces d'actions sont finis) à deux joueurs à chaque itération. En effet, pour résoudre un jeu matriciel à deux joueurs, il faut commencer par déterminer si les joueurs ont des actions pures optimales, si ce n'est pas le cas, il faut calculer des actions mixtes optimales, ce qui se fait par programmation linéaire.

Programmation linéaire

Pour le problème de jeux à un joueur, il est possible de calculer la fonction valeur par programmation linéaire.

$$\begin{aligned} \text{Max} \quad & \sum_{i \in S} v_i \\ s.c \quad & v_i \leq R_i(a) + \beta \sum_{j \in E} P_{ij}(a) v_j \\ & \forall i \in E,\ \forall a \in A(i) \end{aligned}$$

On peut minimiser à la place de maximiser, ce changement entraînerait un renversement des inégalités pour les contraintes. Le calcul de la valeur d'un jeu matriciel peut également s'effectuer par programmation linéaire, pour les détails sur les programmes linéaires à résoudre le lecteur peut consulter [Vor77].

On peut aussi remarquer que, pour les jeux à deux joueurs, en fixant une stratégie du joueur $J1$, le problème de jeu devient un problème de jeu à un joueur et on utilise la programmation linéaire pour résoudre ce jeu. L'itération sur les politiques proposée par Howard pour les jeux stochastiques se base sur ce principe.

Itérations sur les politiques

L'itération sur les politiques a d'abord été construite pour les problèmes de contrôle stochastique par Howard [How60] puis étendu par Hoffman et Karp [HK66] pour les jeux stochastiques. Nous présentons uniquement cette dernière.

Définition 3.8 (*Propriété de sélection*)

La fonction F possède la propriété de sélection si pour tout $x \in \mathbb{R}^d$, il existe une stratégie $\pi \in \Pi$ qui atteint l'infimum dans 3.7. En d'autres termes :

$$\forall\, x \in \mathbb{R}^d \;\exists\, \pi \in \Pi \text{ tel que } F(x) = f^{\pi}(x)$$

La propriété de sélection est satisfaite quand les espaces d'actions du joueur $J1$ sont finis, cette propriété demeure vraie dans les jeux à espaces d'états compacts lorsque les paiements et les probabilités de transitions sont semi-continus inférieurement par rapport aux actions du minimiseur.

Algorithme 3: Itération sur les politiques en jeux

début
- Choisir $\pi^0 \in \Pi$, $k = 0$;
- **tant que** $x \neq F(x)$ **faire**
  - Calculer x^k tel que $f^{\pi^k}(x^k) = x^k$;
  - Evaluer $F(x^k)$;
  - **si** $F(x^k) = x^k$ **alors**
    - Retourner x^k
  - **sinon**
    - prendre π^{k+1} tel que $F(x^k) = f^{\pi^{k+1}}(x^k)$
  - k=k+1;

Il existe bien évidemment d'autres algorithme pour résoudre un problème de contrôle ou un jeu stochastique escompté. Une présentation de quelques algorithmes pour résoudre des jeux stochastiques peut se trouver dans [FR91]. Condon propose quant à elle des algorithmes pour des jeux stochastiques à information parfaite [Con93]. Enfin, un recueil d'algorithmes de résolution de problèmes de jeux stochastiques à un joueur peut être trouvé dans [Put94].

Nous terminons la présentation de l'itération sur les politiques dans le cadre des jeux en présentant un tableau à la figure 3.3 qui évoque l'analogie entre les problèmes de points fixes qui apparaissent en jeux et les problèmes de jeux d'interprétation abstraite.

Jeux stochastiques	Interprétation abstraite
Systèmes dynamiques	programme
Opérateur de Shapley	Fonction sémantique abstraite
Problème à horizon n	n pas logiques
Limite de la valeur à horizon n	Bornes optimales
Itération sur les valeurs	Itération de Kleene

Figure 3.3 – Analogie entre les jeux stochastiques et l'interprétation abstraite

3.2.3 Extension à l'analyse statique

Un algorithme d'itération sur les politiques a été introduit en analyse statique par Gaubert, Goubault et al dans [CGG+05]. L'algorithme d'itération sur les politiques a été étendu pour les treillis complets. L'opérateur de programmation dynamique n'est donc plus nécessairement contractant mais conservant sa croissance, il possède un point fixe. Les avantages de l'itération sur les politiques :

- ne nécessite pas d'opérateur d'élargissement
- retourne à chaque itération un post-point fixe et peut donc être stoppée à une chaque itération.

Algorithme 4: Itération sur les politiques en analyse statique

début
  Soit $k = 1$, choisir π^k, f^{π^k};
  tant que $x^k \neq f(x^k)$ **faire**
    Calculer le plus petit point fixe x_k de f^{π^k};
    Evaluer $f(x_k)$;
    si $f(x_k) = x_k$ **alors**
      retourner x_k
    sinon
      prendre π^{k+1} tel que $f(x_k) = f^{\pi^{k+1}}(x_k)$;
      $k = k + 1$

Dans le treillis des intervalles

Dans le domaine des intervalles, les politiques sont données par les *test*. Quitte à ajouter des opérations *no op* dans le graphe de flot de contrôle, on peut toujours supposer que les opérateurs d'intersections ne sont composés que de deux opérandes. De plus, l'intersection de deux intervalles $[a, b]$ et $[c, d]$, si elle n'est pas vide, est l'un des quatre intervalles suivant :

$$[a, b],\ [c, d],\ [a, d], [c, b]$$

Ce choix de quatre intervalles va fournir l'ensemble d'actions disponibles à un point de contrôle représentant une intersection.

Pour des problèmes liés à la croissance de la fonction, on écrira une variable intervalle x_i de la manière : $[-x_i^-, x_i^+]$. En effet, si on conserve la notation habituelle d'un intervalle, on obtient $[a, b] \subseteq [c, d]$ équivaut à $c \leq a$ et $b \leq d$ ce qui signifie que la fonction du treillis des intervalles vers $\overline{\mathbb{R}}^2$ qui à un intervalle associe les bornes n'est pas croissante.

Exemple 3.8 (Nouvelles règles de calcul dans le domaine des intervalles)

- Si $x_0 = [1, 2]$ alors $x_0^- = -1$ et $x_0^+ = 2$.
- Si $x_1 = x_2 + [1, 2]$ alors $-x_1^- = -x_2^- + 1$ et $x_1^+ = x_2^+ + 2$ ce qui équivaut à $x_1^- = x_2^- - 1$ et $x_1^+ = x_2^+ + 2$.
- Si $x_3 = x_1 \cup x_2$ alors $-x_3^- = \min(-x_1^-, -x_2^-)$ et $x_3^+ = \max(x_1^+, x_2^+)$ d'où $-x_3^- = -\max(x_1^-, x_2^-)$ et $x_3^+ = \max(x_1^+, x_2^+)$ et finalement $x_3^- = \max(x_1^-, x_2^-)$ et $x_3^+ = \max(x_1^+, x_2^+)$.
- Si $x_3 = x_1 \cap x_2$ alors $-x_3^- = \max(-x_1^-, -x_2^-)$ et $x_3^+ = \min(x_1^+, x_2^+)$ d'où $-x_3^- = -\max(x_1^-, x_2^-)$ et $x_3^+ = \min(x_1^+, x_2^+)$ et finalement $x_3^- = \min(x_1^-, x_2^-)$ et $x_3^+ = \min(x_1^+, x_2^+)$.

Définition 3.9 (*Politiques dans le domaine des intervalles*)

Une politique propose donc un choix entre quatre intervalles dès qu'une opération d'intersection apparaît dans la fonction sémantique abstraite. Pour une intersection entre un intervalle $[-a, b]$ et $[-c, d]$, on associe les politiques suivantes :

- ll ($left\ left$) : on choisit comme borne inférieure $-a$ et comme borne supérieure b.
- lr ($left\ right$) : on choisit comme borne inférieure $-a$ et comme borne supérieure d.
- rl ($right\ left$) : on choisit comme borne inférieure $-c$ et comme borne supérieure b.
- rr ($right\ right$) : on choisit comme borne inférieure $-c$ et comme borne supérieure d.

Lorsqu'on choisit une politique, une fonction sans intersection (min) est alors sélectionnée. Cette nouvelle fonction contient uniquement des affectations et des unions (max). Comme, dans l'itération sur les politiques de la théorie des jeux, dès qu'une politique pour le minimiseur est annoncée, le problème devient un jeu à un joueur qu'il faut résoudre. Dans la théorie classique, un jeu à un joueur peut se résoudre par itération sur les valeurs ou par itération sur les politiques ou par programmation linéaire. Dans un premier temps, Gaubert, Goubault et al dans [CGG+05], en analyse statique, calculaient le plus petit point fixe d'une fonction associée à une politique par itération de Kleene, mais ont, par la suite, dans [GGTZ07] utilisé la programmation linéaire pour ce calcul.

Définition 3.10 (*Heuristique de choix de la politique initiale dans les intervalles*)

Pour choisir une stratégie initiale dans les intervalles, on hiérarchise les politiques, l'ordre de préférence est :

1. Les bornes constantes ;
2. Les bornes variables ;
3. Les bornes infinies.

Dans le cas où les deux bornes inférieures (supérieures) sont variables, on prend la borne de gauche (pour éviter un choix aléatoire).

Exemple 3.9 (Exemple introductif)

Nous reprenons, le programme de l'exemple 2.31.

La fonction sémantique concrète peut être réduite à :

$$\begin{array}{lcl} F_1(x) & = & [1,1] \\ F_2(x) & = & ([1,1] \cup (x_2+1)) \cap]-\infty, 10] \\ F_3(x) & = & x_2+1 \\ F_4(x) & = & ([1,1] \cup (x_2+1)) \cap [11, +\infty[\end{array}$$

D'après les règles de calcul sur les intervalles décrites à l'exemple 3.8, on obtient :

$$\begin{array}{lcl} \left(F_1(x)^-, F_1(x)^+\right) & = & (-1,1) \\ \left(F_2(x)^-, F_2(x)^+\right) & = & \left(\max(x_2^- - 1, -1), \min\{10, \max(1, x_2^+ + 1)\}\right) \\ \left(F_3(x)^-, F_3(x)^+\right) & = & \left(x_2^- - 1, x_2^+ + 1\right) \\ \left(F_4(x)^-, F_4(x)^+\right) & = & \left(\min\{-11, \max(x_2^- - 1, -1)\}, \max(1, x_2^+ + 1)\right) \end{array}$$

Au point de contrôle 2 et 4, un min apparaît, il faut donc faire un choix pour initialiser l'itération sur les politiques, nous utilisons l'heuristique classique et nous initialisons à partir des constantes et donc nous commençons par la politique lr pour le point 2 et rl pour le point 4, c'est-à-dire la fonction :

$$\begin{array}{lcl}
\big(f_1(x)^-, f_1(x)^+\big) & = & (-1, 1) \\
\big(f_2(x)^-, f_2(x)^+\big) & = & \big(\max(x_2^- - 1, -1), 10\big) \\
\big(f_3(x)^-, f_3(x)^+\big) & = & \big(x_2^- - 1, x_2^+ + 1\big) \\
\big(f_4(x)^-, f_4(x)^+\big) & = & (-11, \max(1, x_2 + 1))
\end{array}$$

Il faut calculer le plus petit point fixe de f, nous utilisons la programmation linéaire et nous cherchons à résoudre :

$$\text{Min} \sum_{i=1}^{4} (x_i^+ + x_i^-) \,\text{s.c}\;\; f_j(x)^- \leq x_j^- \text{ et } f_j(x)^+ \leq x_j^+,\; \forall j = 1, 2, 3, 4$$

ou encore :

$$\begin{array}{lccccccc}
\text{Min} & & & \sum_{i=1}^{4}(x_i^+ + x_i^-) & & & & \\
& -1 & \leq & x_1^- & 1 & \leq & x_1^+ \\
& -1 & \leq & x_2^- & 10 & \leq & x_2^+ \\
& x_2^- - 1 & \leq & x_2^- & x_2^+ + 1 & \leq & x_3^+ \\
& x_2^- - 1 & \leq & x_3^- & 1 & \leq & x_4^+ \\
& -11 & \leq & x_4^- & x_2^+ + 1 & \leq & x_4^+
\end{array}$$

On trouve :

$$\begin{array}{lcl}
\big(\overline{x}_1^-, \overline{x}_1^+\big) & = & (-1, 1) \\
\big(\overline{x}_2^-, \overline{x}_2^+\big) & = & (-1, 10) \\
\big(\overline{x}_3^-, \overline{x}_3^+\big) & = & (-2, 11) \\
\big(\overline{x}_4^-, \overline{x}_4^+\big) & = & (-11, 11)
\end{array}$$

et comme :

$$\begin{array}{lcll}
\big(F_1(\overline{x})^-, F_1(\overline{x})^+\big) & = & (-1, 1) & \\
\big(F_2(\overline{x})^-, F_2(\overline{x})^+\big) & = & \big(\max(\overline{x}_2^- - 1, -1), \min\{10, \max(1, \overline{x}_2^+ + 1)\}\big) & = (-1, 10) \\
\big(F_3(\overline{x})^-, F_3(\overline{x})^+\big) & = & \big(\overline{x}_2^- - 1, \overline{x}_2^+ + 1\big) & = (-2, 11) \\
\big(F_4(\overline{x})^-, F_4(\overline{x})^+\big) & = & \big(\min\{-11, \max(\overline{x}_2^- - 1, -1)\}, \max(1, \overline{x}_2^+ + 1)\big) & = (-11, 11)
\end{array}$$

La solution $\overline{x}$ est donc un point fixe de la fonction F et l'itération sur les politiques se termine en retournant cette solution. L'invariant pour le programme est donc le vecteur d'intervalles :

$$\begin{array}{lcl}
x_1 & = & [1, 1] \\
x_2 & = & [1, 10] \\
x_3 & = & [2, 11] \\
x_4 & = & [11, 11]
\end{array}$$

Exemple 3.10 (Exemple avec changement de politique)
Nous introduisons un nouveau programme :

```
x=0;[1]
while (x<=100) [2]{
  x=10+0.5*x;[3]
  } [4]
```

La fonction sémantique concrète peut être réduite à :

$$\begin{array}{lcl} F_1(x) & = & [0,0] \\ F_2(x) & = & ([0,0] \cup (10 + 0.5x_2)) \cap]-\infty, 100] \\ F_3(x) & = & 10 + 0.5x_2 \\ F_4(x) & = & ([0,0] \cup (10 + 0.5x_2)) \cap [101, +\infty[\end{array}$$

D'après les règles de calcul sur les intervalles décrites à l'exemple 3.8, on obtient :

$$\begin{array}{lcl} \left(F_1(x)^-, F_1(x)^+\right) & = & (0,0) \\ \left(F_2(x)^-, F_2(x)^+\right) & = & \left(\max(0.5x_2^- - 10, 0), \min\{100, \max(0, 10 + 0.5x_2^+)\}\right) \\ \left(F_3(x)^-, F_3(x)^+\right) & = & \left(0.5x_2^- - 10, 10 + 0.5x_2^+\right) \\ \left(F_4(x)^-, F_4(x)^+\right) & = & \left(\min\{-101, \max(0.5x_2^- - 10, 0)\}, \max(0, 10 + 0.5x_2^+)\right) \end{array}$$

Au point de contrôle 2 et 4, un min apparaît, il faut donc faire un choix pour initialiser l'itération sur les politiques, nous utilisons l'heuristique classique et nous initialisons à partir des constantes et donc on commence par la politique lr pour le point 2 et rl pour le point 4 :

$$\begin{array}{lcl} \left(f_1(x)^-, f_1(x)^+\right) & = & (0,0) \\ \left(f_2(x)^-, f_2(x)^+\right) & = & \left(\max(0.5x_2^- - 10, 0), 100\right) \\ \left(f_3(x)^-, f_3(x)^+\right) & = & \left(0.5x_2^- - 10, 0.5x_2^+ + 10\right) \\ \left(f_4(x)^-, f_4(x)^+\right) & = & \left(-101, \max(0, 0.5x_2^+ + 10)\right)) \end{array}$$

Il faut calculer le plus petit point fixe de f, nous utilisons la programmation linéaire et nous cherchons à résoudre :

$$\text{Min} \sum_{i=1}^{4} (x_i^+ + x_i^-) \,\text{s.c}\;\; f_j(x)^- \leq x_j^- \text{ et } f_j(x)^+ \leq x_j^+ \;\forall j = 1,2,3,4$$

ou encore :

$$\begin{array}{lll} \text{Min} & \displaystyle\sum_{i=1}^{4} (x_i^+ + x_i^-) & \\ & 0 \leq x_1^- \qquad 0 \leq x_1^+ & \\ & 0 \leq x_2^- \qquad 100 \leq x_2^+ & \\ & 0.5x_2^- - 10 \leq x_2^- \qquad 10 + 0.5x_2^+ \leq x_3^+ & \\ & 0.5x_2^- - 10 \leq x_3^- \qquad 0 \leq x_4^+ & \\ & -101 \leq x_4^- \qquad 10 + 0.5x_2^+ \leq x_4^+ & \end{array}$$

On trouve :

$$\begin{array}{lcl} \left(\overline{x}_1^-, \overline{x}_1^+\right) & = & (0,0) \\ \left(\overline{x}_2^-, \overline{x}_2^+\right) & = & (0,100) \\ \left(\overline{x}_3^-, \overline{x}_3^+\right) & = & (-10,60) \\ \left(\overline{x}_4^-, \overline{x}_4^+\right) & = & (-101,60) \end{array}$$

et comme :

$$\begin{array}{lcll} \left(F_1(\overline{x})^-, F_1(\overline{x})^+\right) & = & (0,0) & \\ \left(F_2(\overline{x})^-, F_2(\overline{x})^+\right) & = & \left(\max(0.5\overline{x}_2^- - 10, 0), \min\{100, \max(0, 0.5\overline{x}_2^+ + 10)\}\right) & = (0,60) \\ \left(F_3(\overline{x})^-, F_3(\overline{x})^+\right) & = & \left(\overline{x}_2^- - 1, \overline{x}_2^+ + 1\right) & = (-10,60) \\ \left(F_4(\overline{x})^-, F_4(\overline{x})^+\right) & = & \left(\min\{-101, \max(0.5\overline{x}_2^- - 10, 0)\}, \max(0, 0.5\overline{x}_2^+ + 10)\right) & = (-101,60) \end{array}$$

On voit qu'à la deuxième ligne, $\min\{100, \max(0, 0.5\overline{x}_2^+ + 10)\} \neq 100$ et donc, la solution $\overline{x}$ n'est pas un point fixe de la fonction F et il faut changer de politique, la valeur 60 est atteinte par la fonction $\max(0, 0.5\overline{x}_2^+ + 10)$ et donc on résout une nouveau problème de point fixe avec la fonction :

$$\begin{array}{lcl} \left(f_1(x)^-, f_1(x)^+\right) & = & (0,0) \\ \left(f_2(x)^-, f_2(x)^+\right) & = & \left(\max(0.5x_2^- - 10, 0), \max(0, 0.5\overline{x}_2^+ + 10)\right) \\ \left(f_3(x)^-, f_3(x)^+\right) & = & \left(0.5x_2^- - 10, 0.5x_2^+ + 10\right) \\ \left(f_4(x)^-, f_4(x)^+\right) & = & \left(-101, \max(0, 0.5x_2^+ + 10)\right)\end{array}$$

et par programmation linéaire on trouve le vecteur :

$$\begin{array}{lcl} \left(x_1^-, x_1^+\right) & = & (0,0) \\ \left(x_2^-, x_2^+\right) & = & (0,60) \\ \left(x_3^-, x_3^+\right) & = & (-10,60) \\ \left(x_4^-, x_4^+\right) & = & (-101,60)\end{array}$$

Ce vecteur est également un point fixe de F, l'itération sur les politiques se termine en retournant cette solution. On conclut que l'invariant du programme est le vecteur d'intervalles :

$$\begin{array}{lcl} x_1 & = & [0,0] \\ x_2 & = & [0,60] \\ x_3 & = & [10,60] \\ x_4 & = & \emptyset\end{array}$$

L'invariant calculé permet de savoir que le programme boucle à l'infini et que le point de contrôle 4 n'est jamais atteint.

Dans les zones et gabarits de Sankaranarayanan

L'itération sur les politiques a été étendue par Taly et al [GGTZ07] aux travaux de Miné [Min04] et Sankaranarayanan et al [SSM05]. Nous décrivons brièvement la méthode dans ce paragraphe mais nous reviendrons sur la méthode d'itérations sur les politiques dans les zones et gabarits de Sankaranarayanan dans le chapitre 5. Nous verrons que les invariants, dans ces domaines, sont calculés en résolvant des problèmes d'optimisation linéaire avec contraintes (linéaires). Une méthode classique d'optimisation est de "dualiser" le problème. Cette dualisation (lagrangienne) remplace le problème de maximisation dû à la recherche d'invariant par un problème de minimisation. Les variables "duales" donnent l'ensemble des politiques. De plus, on sait qu'en optimisation linéaire, l'ensemble des solutions d'un problème sous contraintes se trouvent sur les sommets du polyèdre des contraintes et donc en nombre fini. L'ensemble des politiques est, par conséquent, l'ensemble de sommets du polyèdre des contraintes du problème dual.

3.3 Itération max-stratégies

On peut mentionner également les travaux de H. Seidl et de T. Gawlitza [GS07a, GS07b]. Ce type d'itération est une itération sur les politiques croissante. La méthode consiste à calculer le plus petit point fixe de la fonction sémantique abstraite en considérant les unions plutôt que les intersections comme politiques. L'initialisation de la méthode se fait en assignant

$-\infty$ pour toutes les variables puis de la même manière que l'itération sur les politiques sur les min, la fonction atteignant le max est sélectionnée et le plus grand point fixe de la fonction est calculée. L'inconvénient de cette méthode est qu'une suite croissante de points est calculée mais sous-approxime le plus petit point fixe de la fonction sémantique abstraite, la méthode ne donne aucune information pertinente tant que le plus petit point fixe n'est pas atteint. L'avantage est que par cette méthode le plus petit point fixe est atteint si l'itération termine.

Deuxième partie

Domaines numériques abstraits des sous-niveaux

CHAPITRE 4

DOMAINES DES SOUS-NIVEAUX

Au chapitre 2, nous avons vu que le problème 2.1 n'était pas calculable en général. Une méthode classique de l'analyse statique pour pouvoir calculer une sur-approximation valide est l'interprétation abstraite. Dans ce chapitre, nous prenons $\mathbb{G}^d = \mathbb{R}^d$ et nous considérons le problème 2.1 comme un problème d'optimisation continue. Une méthode classique de l'optimisation est de relâcher le problème lorsque celui-ci n'est pas réalisable. Il s'agit de résoudre un nouveau problème d'optimisation où la fonction à minimiser (appelée aussi fonction objectif) est plus grande, au sens fonctionnel, que la fonction objectif du problème initial et où l'ensemble de contraintes est plus fin (c'est-à-dire contient moins d'éléments). Dans le problème 2.1, la fonction objectif est l'identité Id, l'ensemble des contraintes est composé de deux contraintes :

$$X \in \wp(\mathbb{R}^d)^{n+1} \tag{4.1}$$
$$F^C(X) \subseteq X \tag{4.2}$$

Nous pouvons construire un problème relâché en conservant comme une fonction objectif l'identité, mais en choisissant deux ensembles, le premier inclus dans $\wp(\mathbb{R}^d)^{n+1}$, le deuxième inclus dans $\{X \in \wp(\mathbb{R}^d)^{n+1} \mid F^C(X) \subseteq X\}$. La contrainte (4.1), peut être relâchée en fixant la forme géométrique des contraintes. Cette forme géométrique est, en fait, la concrétisation d'éléments d'un domaine abstrait $\mathcal{D}_a$. Ainsi la contrainte (4.2) peut être rendue plus fine, en considérant une application F sur le domaine abstrait, telle que la concrétisation des éléments vérifiant $\{X \in \mathcal{D}_a^{n+1} \mid X = F(X)\}$ satisfait les contraintes (4.1) et (4.2).

Dans ce chapitre, il est question de généraliser le travail de Sankaranarayan, Sipma et Manna [SSM05] sur les gabarits linéaires. Le domaine des gabarits linéaires consiste à se donner un nombre fini de formes linéaires qui définissent les faces d'un polyèdres, la forme géométrique est donc un certain polyèdre. Suivant l'idée de Sankaranarayan, les formes géométriques sont, également déterminées à partir d'un ensemble *non nécessairement fini* de fonctions de base *non nécessairement linéaires*.

Ainsi, les parties de $\mathbb{R}^d$ que l'on considère sont les intersections de sous-niveaux des fonctions de cette base. On s'intéresse également aux ensembles de lignes de niveaux sur ces fonctions de base.

Dans la section 4.1, nous commençons par de brefs rappels d'analyse convexe. Dans la section 4.2, nous définissons plus formellement les concrétisations géométriques ainsi que les éléments du domaine abstrait. Dans la section 4.3, nous construisons les treillis nécessaires aux calculs des invariants numériques. Enfin, dans la section 4.4, nous présenterons quelques ensembles de fonctions de base utiles en interprétation abstraite.

4.1 Rappels d'analyse convexe

Nous rappelons quelques notions de base sur la convexité, pour plus de détails en dimension finie, le lecteur est invité à consulter [Roc96]. Ici l'ensemble E désigne un espace vectoriel réel, l'addition et la multiplication scalaire sont donc définies au sens de cet espace vectoriel. L'ensemble F désigne un espace topologique, les sous-ensembles fermés sont définis à partir de cette topologie. Dans cette partie, les fonctions sur E sont toutes à valeurs réelles.

Nous rappelons qu'un ensemble convexe contient tous ses segments, plus formellement :

Définition 4.1 (*Ensembles convexes*)

Un ensemble $C \subseteq E$ est convexe si, pour tout $x, y \in C$, pour tout $\lambda \in]0,1[$, $\lambda x + (1-\lambda)y \in C$.

Remarque 4.1 Par convention, l'ensemble vide est convexe.

Exemple 4.1
Un ensemble convexe contient tous ses segments, ceci est illustré par l'ellipse de la figure 4.1

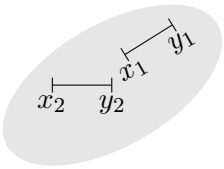


Figure 4.1 – Une ellipse est un ensemble convexe

Une fonction convexe est caractérisée par le fait que les images de segments sont plus petites que les segments image.

Définition 4.2 (*Fonctions convexes*)

Une fonction f sur E est convexe si pour tout $x, y \in E$, pour tout $\lambda \in]0,1[$,
$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$.

Exemple 4.2
La fonction $x :\mapsto \frac{1}{4}x^2$ est une fonction convexe. L'inégalité de convexité est illustrée par la figure 4.2.

Définition 4.3 (*Fonctions concaves*)

Une fonction f est concave si $-f$ est une fonction convexe.

La convexité d'une fonction est équivalente à la convexité d'un ensemble lié à la fonction appelé l'épigraphe.

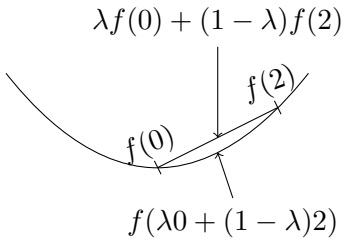


Figure 4.2 – Inégalité de convexité

Définition 4.4 (*Epigraphe*)

Soit f une fonction sur E. On appelle épigraphe de f, l'ensemble noté epi(f) défini par $\{(x, \alpha) \in \mathbb{R}^d \times \mathbb{R} \mid f(x) \leq \alpha\}$

Exemple 4.3

Nous reprenons la fonction f convexe de l'exemple 4.2. L'épigraphe de f est la partie grisée de la figure 4.3.

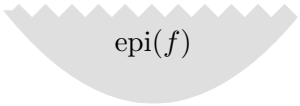


Figure 4.3 – L'épigraphe de la fonction f

Proposition 4.1 (*Fonction convexe et épigraphe*)

Une fonction sur E est convexe si et seulement si son épigraphe est un sous-ensemble convexe de $E \times \mathbb{R}$.

Grâce à cette caractérisation, on en déduit les propriétés suivantes :

Propriété 4.1 (*Opérations sur les fonctions convexes*)

1. Soient I un ensemble fini, $\{f_i\}_{i\in I}$ une famille de fonctions convexes et $(\lambda_i)_{i\in I}$ une famille finie de nombres réels positifs alors $\sum_{i\in I} \lambda_i f_i$ est une fonction convexe.
2. Soit $\{f_i\}_{i\in I}$ une famille quelconque de fonctions convexes alors $\sup_{i\in I} f_i$ est une fonction convexe.

Pour minimiser une fonction, on peut s'intéresser à une classe d'ensembles de points dont l'image ne dépasse un certain réel.

Définition 4.5 (*Sous-niveaux*)

Soit f une fonction de E et soit $\alpha \in \mathbb{R}$, on appelle sous-niveau de f au niveau α, l'ensemble $S_\alpha(f) = \{x \in E \mid f(x) \leq \alpha\}$.

Exemple 4.4

Le sous-niveau de la fonction convexe $x \mapsto x^2$ au niveau $\alpha = 1$ est décrit par la figure 4.4.

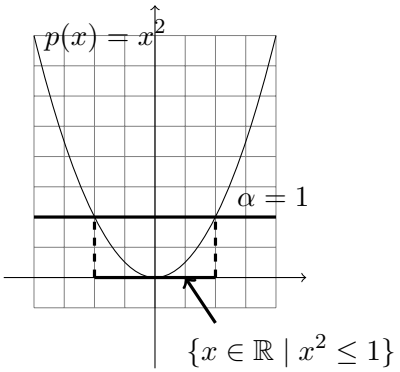


Figure 4.4 – Le sous-niveau de la fonction carrée pour $\alpha = 1$

Proposition 4.2 (*Fonctions convexes et sous niveaux*)

Soit f une fonction convexe sur E. Pour tout $\alpha \in \mathbb{R}$, $S_\alpha(f)$ est un sous-ensemble convexe de E.

En analyse convexe, la continuité des fonctions n'est pas forcément nécessaires :

Définition 4.6 (*Fonction semi-continue inférieurement*)

On dit qu'une fonction f sur F est semi-continue inférieurement (en abrégé s.c.i) si pour tout $\alpha \in \mathbb{R}$, $S_\alpha(f)$ est un sous-ensemble fermé de F.

Définition 4.7 (*Fonction semi-continue supérieurement*)

Une fonction g est semi-continue supérieurement (en abrégé s.c.s) si $-g$ est semi-continue inférieurement.

Exemple 4.5

On définit une fonction g par :

$$g(x) = \begin{cases} g(0) = 0 \\ g(x) = x + 2 & \text{si } x > 0 \\ g(x) = -x + 1 & \text{si } x < 0 \end{cases}$$

La fonction g est représentée par le graphe de la figure 4.5. Cette fonction est semi-continue inférieurement.

Propriété 4.2 (*Supremum de fonctions semi-continue inférieurement*)

1. Soit $\{f_i\}_{i \in I}$ une famille quelconque de fonctions s.c.i alors $\sup_{i \in I} f_i$ est une fonction s.c.i.
2. Soit $\{g_i\}_{i \in I}$ une famille quelconque de fonctions s.c.s alors $\inf_{i \in I} g_i$ est une fonction s.c.s.

Proposition 4.3 (*Fonction semi-continue inférieurement et épigraphe*)

Une fonction sur F est s.c.i si et seulement si son épigraphe est un sous-ensemble fermé de $F \times \mathbb{R}$.

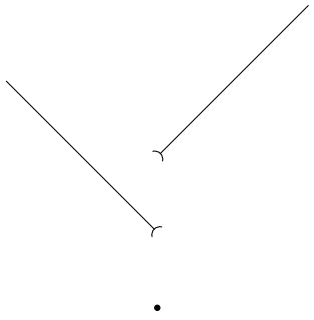


Figure 4.5 – La fonction g de l'exemple 4.5

Définition 4.8 (*Conjuguée de Fenchel-Moreau*)

Soit F un espace vectoriel topologique et F' le dual topologique de F. Soit $f : F \mapsto \mathbb{R} \cup \{+\infty\}$. On appelle conjuguée de Fenchel de f la fonction $f^* : F' \mapsto \mathbb{R} \cup \{+\infty\}$ définie par :

$$x' \mapsto f^*(x') = \sup_{x \in F} x'(x) - f(x)$$

Exemple 4.6

Soit $g : \mathbb{R} \mapsto \mathbb{R} \cup \{+\infty\}$ définie par :

$$x \mapsto g(x) = \begin{cases} 0 & \text{si } x \in [0,1] \\ +\infty & \text{sinon} \end{cases}$$

La conjuguée de g est la fonction définie $g^* : \mathbb{R} \mapsto \mathbb{R}$ par :

$$x' \mapsto g^*(x') = \begin{cases} 0 & \text{si } x' \leq 0 \\ x' & \text{sinon} \end{cases}$$

La fonction g définie dans l'exemple 4.6 est appelée fonction caractéristique convexe de l'ensemble $[0,1]$. Nous réutiliserons ce type de fonctions pour caractériser une $\mathbb{P}$-fonction support.

Définition 4.9 (*biconjuguée de Fenchel-Moreau*)

Soit F un espace de Banach réflexif. Soit $f : F \mapsto \mathbb{R} \cup \{+\infty\}$. On appelle biconjuguée de Fenchel de f, la fonction $f^{**} : F \mapsto \mathbb{R} \cup \{+\infty\}$ définie par :

$$x \mapsto f^{**}(x) = \sup_{x' \in F'} x'(x) - f^*(x')$$

Exemple 4.7

Nous reprenons la fonction g de l'exemple 4.6.

La biconjuguée de g est la fonction définie $g^{**} : \mathbb{R} \mapsto \mathbb{R}$ par :

$$x \mapsto g^{**}(x) = \sup_{x' \in \mathbb{R}} x'x - g^*(x') = \max\left(\sup_{x' \leq 0} x'x, \sup_{x' > 0} x'(x-1)\right)$$

Si $x > 1$ alors $\sup_{x'>0} x'(x-1) = +\infty$ et si $x < 0$ alors $\sup_{x'\leq 0} x'x = +\infty$ et si $x \in [0,1]$, $\sup_{x'\leq 0} x'x = \sup_{x'>0} x'(x-1) = 0$. On conclut que $g^{**} = g$.

Dans l'exemple 4.7, nous avons conclu que la biconjuguée de la fonction g était égale à elle-même. En fait, la classe des fonctions égales à leur biconjuguée est bien connue puisqu'il s'agit soit de la fonction identiquement égales à la fonction $-\infty$ ou l'ensemble des fonctions convexes semi-continues qui ne prennent pas la valeur $-\infty$.

Théorème 4.1 (*Fonctions convexes s.c.i et biconjuguée*)

Soit F un espace de Banach réflexif et soit $f : F \mapsto \mathbb{R} \cup \{+\infty\}$. On suppose qu'il existe $x \in F$ tel que $f(x) \in \mathbb{R}$.

$$f \text{ convexe s.c.i} \iff f = f^{**}$$

Pour la preuve, le lecteur pourra consulter [Bon06, Théorème 4.17, Corollaire 4.18].

Remarque 4.2 L'hypothèse "f est à valeurs $\mathbb{R} \cup \{+\infty\}$" est importante, puisque la fonction f définie par :

$$x \mapsto f(x) = \begin{cases} -\infty & x \in K \\ +\infty & \text{sinon} \end{cases}$$

où $K \subsetneq F$ est un ensemble convexe fermé non vide. La fonction f est convexe s.c.i mais qui n'est pas égale à sa biconjuguée.

4.2 $\mathbb{P}$-sous niveaux et $\mathbb{P}$-fonction support

Dans le but de figer la géométrie des invariants numériques cherchés, on se munit d'un ensemble $\mathbb{P}$ de fonctions de $\mathbb{R}^d$ dans $\mathbb{R}$. Nous introduisons ainsi un ensemble $\mathbb{P}$, appelée une configuration et les éléments de cette configuration, des fonctions de base. Nous introduisons également $\overline{\mathbb{R}}^{\mathbb{P}}$ des fonctions de l'ensemble $\mathbb{P}$ vers $\overline{\mathbb{R}}$. Ces fonctions vont jouer le rôle de degré de liberté sur les fonctions de base.

Nous notons $\mathbb{R}^{\mathbb{P}}$ l'ensemble des éléments de $\overline{\mathbb{R}}^{\mathbb{P}}$ qui prennent des valeurs finies sur $\mathbb{P}$ et $\mathbb{R}_+^{\mathbb{P}}$ l'ensemble des fonctions de $\mathbb{P}$ à valeurs positives. L'ensemble $\overline{\mathbb{R}}^{\mathbb{P}}$ est muni de l'ordre partiel classique $\leq$ défini par $f \leq g$ ssi pour tout $p \in \mathbb{P}$, $f(p) \leq g(p)$. L'ensemble $\wp(\mathbb{R}^d)$ des parties de $\mathbb{R}^d$ est lui muni de l'ordre de l'inclusion. Enfin, puisqu'il est question de géométrie, nous munissons $\mathbb{R}^d$ de la norme euclidienne que nous noterons $\|\cdot\|$. Nous noterons le produit scalaire de $x, y \in \mathbb{R}^d$ par $x \cdot y$.

En analyse convexe, il est naturel de travailler avec les formes linéaires, nous appelons donc *configuration linéaire classique* la configuration $\mathbb{P}$ qui regroupe toutes les formes linéaires sur $\mathbb{R}^d$, par le théorème de Riesz, cette configuration est isomorphe à $\mathbb{R}^d$.

Pour une configuration fixée $\mathbb{P}$, pour une fonction de $\overline{\mathbb{R}}^{\mathbb{P}}$, nous nous intéressons aux fonctions de bases qui ont une image finie, nous appelons cet ensemble de fonctions de bases le domaine d'un élément de $\overline{\mathbb{R}}^{\mathbb{P}}$.

Définition 4.10 (*Domaine*)

Soit v un élément de $\overline{\mathbb{R}}^{\mathbb{P}}$ tel que, pour tout $p \in \mathbb{P}$, $v(p) > -\infty$. On appelle domaine de v, l'ensemble noté $\mathrm{dom}(v)$ défini par :

$$\{p \in \mathbb{P} \mid v(p) < +\infty\}$$

Exemple 4.8
On pose $\mathbb{P} = \{x :\mapsto x^2 - 1,\ x \mapsto -x^2 + 1\}$, et soit v la fonction qui à $p \in \mathbb{P}$ associe son supremum sur tout $\mathbb{R}$. On a $\mathrm{dom}(v) = \{x \mapsto -x^2 + 1\}$.

Exemple 4.9
Prenons comme ensemble de fonctions $\mathbb{P}$, l'ensemble des fonctions continues croissantes sur $\mathbb{R}$. Considérons la fonction v sur $\mathbb{P}$ qui, à chaque fonction $p \in \mathbb{P}$, associe l'intégrale de la valeur absolue p. On a donc $\mathrm{dom}(v) = \mathbb{P} \cap \mathcal{L}^1(\mathbb{R})$ où, $\mathcal{L}^1(\mathbb{R})$ est l'ensemble des fonctions Lebesgue-intégrables sur $\mathbb{R}$.

Définition 4.11 (*Elément presque infini*)

Un élément $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ est dit presque infini si $v(p) > -\infty$ pour tout $p \in \mathbb{P}$ et si $\mathrm{dom}(v)$ est fini.

Exemple 4.10
On considère la configuration linéaire classique. On se donne un polyèdre A de $\mathbb{R}^d$, il existe deux familles finies $(p_i)_{i\in I} \in \mathbb{R}^d$, $(b_i)_{i\in I}$ telles que $A = \{x \in \mathbb{R}^d \mid p_i \cdot x \leq b_i,\ i \in I\}$. On pose $v(p_i) = b_i$ si $i \in I$ et $v(p_i) = +\infty$ si $p_i \in \mathbb{R}^d$ et si $i \notin I$. On a $A = \{x \in \mathbb{R}^d \mid p \cdot x \leq v(p),\ p \in \mathbb{R}^d\}$ et v est un élément presque infini.

Exemple 4.11
Considérons la suite de fonctions définie, pour $n \in \mathbb{N}$, par $p_n : x \mapsto n^{1/4}x^2 + n^{1/2}x + n^{1/2}$. Soit la configuration $\mathbb{P} = \{p_n,\ n \in \mathbb{N}\}$. Soit $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ défini, pour $p \in \mathbb{P}$, par $v(p) = \mathrm{card}(\{x \in \mathbb{R}^d \mid p(x) \leq 0\})$ où $\mathrm{card}(C)$ désigne le cardinal d'un ensemble C. L'élément v est un élément presque infini.

4.2.1 $\mathbb{P}$-sous niveaux

En analyse convexe, il est classique de s'intéresser aux ensembles de sous-niveaux dont la définition est rappelée par la définition 4.5, car ils ont de bonnes propriétés de convexité et de fermeture. Dans notre cadre, on s'intéresse aux intersections de sous-niveaux des fonctions $p \in \mathbb{P}$ où les niveaux sont donnés par une fonction $v \in \overline{\mathbb{R}}^{\mathbb{P}}$. Ces intersections de sous-niveaux décrivent la concrétisation géométrique d'une fonction $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ dans la base de fonctions $\mathbb{P}$.

Définition 4.12 ($\mathbb{P}$-*sous niveaux*)

Soit v un élément de $\overline{\mathbb{R}}^{\mathbb{P}}$, on appelle $\mathbb{P}$-sous niveaux de v l'ensemble noté $v^{\mathbf{s}}$ défini par :

$$\{x \in \mathbb{R}^d \mid p(x) \leq v(p), \forall\, p \in \mathbb{P}\}$$

Si pour tout $v \in \overline{\mathbb{R}}^{\mathbb{P}}$, $v(p) > -\infty$ et que $\mathrm{dom}(v) \neq \emptyset$, on peut remplacer $p \in \mathbb{P}$ par $p \in \mathrm{dom}(v)$.

Remarque 4.3 Un $\mathbb{P}$-sous niveau peut être éventuellement vide même si, pour tout $p \in \mathbb{P}$, $v(p) > -\infty$.

Remarque 4.4 La fonction $v \mapsto v^{\mathbf{s}}$ est, en fait, une fonction de concrétisation, on le verra à la proposition 4.4. En principe, nous devrions appeler cette fonction γ.

Une même figure géométrique peut être la concrétisation de plusieurs $\mathbb{P}$-sous niveaux.

Exemple 4.12
On peut caractériser la boule unité $\mathbb{R}^d$ en tant que $\mathbb{P}$-sous niveau. Nous pouvons choisir $\mathbb{P} = \{x \mapsto \|x\|\}$ et la fonction $v(\|\cdot\|) = 1$ ou bien choisir la configuration linéaire classique et $v(p) = 1$, pour toute forme linéaire p.

En prenant des formes linéaires, nous retrouvons des polyèdres.

Exemple 4.13
Soit $\mathbb{P} = \{p_1, p_2, p_3\}$ où $p_1 : (x, y) \mapsto y - x$, $p_2 : (x, y) \mapsto y + x$ et $p_3 : (x, y) \mapsto -y$. Soit la fonction w définie par $w(p_1) = 3$, $w(p_2) = 3$ et $w(p_3) = 0$. Le $\mathbb{P}$-sous niveau de w est donné par la figure 4.6.

Figure 4.6 – Le P-sous niveau de w avec des fonctions de base linéaires

Une configuration $\mathbb{P}$ étant assez générale, les $\mathbb{P}$-sous niveaux ne sont pas nécessairement convexes comme le montrent les exemples 4.14 et 4.15.

Exemple 4.14
La sphère unité de $\mathbb{R}^d$ peut se coder comme $\{x \in \mathbb{R}^d \mid \|x\| \leq 1,\ -\|x\| \leq -1\}$ c'est-à-dire comme $\mathbb{P}$-sous niveaux de la fonction v avec $\mathbb{P} = \{x \mapsto \|x\|, x \mapsto -\|x\|\}$, $v(\|\cdot\|) = 1$ et $v(-\|\cdot\|) = -1$.

Exemple 4.15
Considérons, la base de fonctions $\mathbb{P}$ suivante : $p_1(x, y) = -y^2 - (x - 3)^2$, $p_2(x, y) = -y^2 - (x + 3)^2$, $p_3(x, y) = -(y - 3)^2 - x^2$, $p_4(x, y) = -(y + 3)^2 - x^2$, $p_5(x, y) = x$, $p_6(x, y) = -x$, $p_7(x, y) = y$, $p_8(x, y) = -y$.

Maintenant, soit $v \in \overline{\mathbb{R}}^{\mathbb{P}}$, telle que $v(p_1) = v(p_2) = v(p_3) = v(p_4) = -4$ et $v(p_5) = v(p_6) = v(p_7) = v(p_8) = 3$.

L'ensemble de $\mathbb{P}$-sous niveau de v est décrit à la figure 4.7.

Conformément à la définition d'une application de concrétisation, on montre que l'application $\mathbb{P}$-sous niveau est croissante.

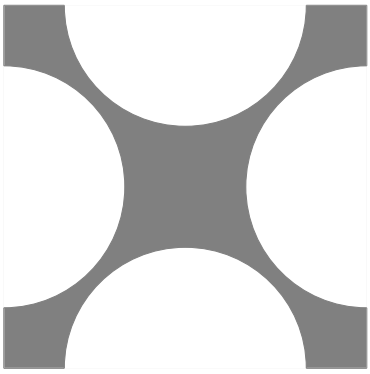


Figure 4.7 – Un $\mathbb{P}$-sous niveau non convexe

Proposition 4.4 (*Monotonie de l'application $\mathbb{P}$-sous niveau*)

L'application de $\overline{\mathbb{R}}^{\mathbb{P}}$ dans $\wp(\mathbb{R}^d)$, $v \mapsto v^{\mathbf{s}}$ est monotone.

Démonstration

Soit $v, w \in \overline{\mathbb{R}}^{\mathbb{P}}$, tels que $v \leq w$. Si $v^{\mathbf{s}}$ est vide, il n'y a rien à montrer. Sinon soient $x \in v^{\mathbf{s}}$ et $p \in \mathbb{P}$. Puisque $x \in v^{\mathbf{s}}$, $p(x) \leq v(p)$, or, comme $v \leq w$ dans $\overline{\mathbb{R}}^{\mathbb{P}}$ alors $p(x) \leq w(p)$ d'où, $x \in w^{\mathbf{s}}$. On conclut que $v \leq w$ dans $\overline{\mathbb{R}}^{\mathbb{P}}$ implique $v^{\mathbf{s}} \subseteq w^{\mathbf{s}}$ et finalement, $v \mapsto v^{\mathbf{s}}$ est monotone. ∎

Définition 4.13 (*Fonctions d'évaluation*)

Nous appelons fonction d'évaluation au point $x \in \mathbb{R}^d$, l'élément de $\mathbb{R}^{\mathbb{P}}$ noté $\mathbf{e}_x$ défini pour $p \in \mathbb{P}$ par $\mathbf{e}_x(p) = p(x)$.

L'introduction des fonctions d'évaluation nous permet de caractériser simplement un $\mathbb{P}$-sous niveau grâce à une inégalité fonctionnelle. Par la suite, les fonctions d'évaluation nous permettront d'établir le lien entre $\mathbb{P}$-sous niveau et convexité abstraite des fonctions.

Propriété 4.3 (*Caractérisation fonctionnelle d'un $\mathbb{P}$-sous niveau*)

Le $\mathbb{P}$-sous niveau $v^{\mathbf{s}}$ peut se caractériser par l'inégalité fonctionnelle :

$$\{x \in \mathbb{R}^d \mid \mathbf{e}_x \in \mathbb{R}^{\mathbb{P}},\ (v - \mathbf{e}_x)(p) \geq 0,\ \forall p \in \mathbb{P}\}$$

ce qui est équivalent à $\{x \in \mathbb{R}^d \mid v - \mathbf{e}_x \in \mathbb{R}^{\mathbb{P}}_+\}$.

4.2.2 $\mathbb{P}$-fonction support

Pour un sous-ensemble de $\mathbb{R}^d$, nous nous intéressons aux calculs des valeurs maximales des fonctions de base sur ce sous-ensemble. Ce principe est classique en analyse convexe. Cependant, en analyse convexe, les fonctions de base sont toutes les formes linéaires sur $\mathbb{R}^d$ et on appelle fonction support d'un sous-ensemble convexe fermé non vide C, la fonction qui, à chaque forme linéaire, associe sa valeur maximale (éventuellement infinie) sur C. Moreau dans [Mor70], a généralisé la notion de fonction support pour un ensemble de fonctions non-linéaires.

Ici, on considère des configurations $\mathbb{P}$ et des parties de $\mathbb{R}^d$ quelconques.

Définition 4.14 ($\mathbb{P}$-*fonction support*)

Soit C un sous ensemble de $\mathbb{R}^d$, on définit la $\mathbb{P}$-fonction support C^σ de C par :

$$p :\mapsto C^\sigma(p) = \sup_{x \in C} p(x)$$

Remarque 4.5 Comme par convention $\sup_{\emptyset} = -\infty$, si C est vide alors $C^\sigma \equiv -\infty$.

Remarque 4.6 La fonction $\mathbb{P}$-fonction support est en réalité une application d'abstraction, ceci sera mise en évidence par la proposition 4.5. En interprétation abstraite, une telle application est notée α.

Si $\mathbb{P}$ est la configuration linéaire classique alors la $\mathbb{P}$-fonction support est, bien évidemment, la classique fonction support de l'analyse convexe.

Exemple 4.16
Soit C la boule unité pour la norme euclidienne de $\mathbb{R}^d$. On suppose que $\mathbb{P}$ est l'ensemble des formes linéaires. La $\mathbb{P}$-fonction support est la classique fonction support et $C^\sigma(p) = 1$ pour tout $p \in \mathbb{P}$.

Exemple 4.17
Reprenons l'exemple 4.13 où $\mathbb{P} = \{p_1, p_2, p_3\}$ avec $p_1 : (x, y) \mapsto y - x$, $p_2 : (x, y) \mapsto y + x$ et $p_3 : (x, y) \mapsto -y$. La fonction w telle que $w(p_1) = 3$, $w(p_2) = 3$ et $w(p_3) = 0$ est la $\mathbb{P}$-fonction support du $\mathbb{P}$-sous niveau 4.6.

Exemple 4.18
Soit C le disque centré en $(0, 0)$ de rayon $\sqrt{5}$. Soit $\mathbb{P}$, l'ensemble des fonctions de base : $p_1 : (x, y) \mapsto y - x$, $p_2 : (x, y) \mapsto y$ et $p_3 : (x, y) \mapsto x$. La fonction C^σ est donné à la figure 4.8.

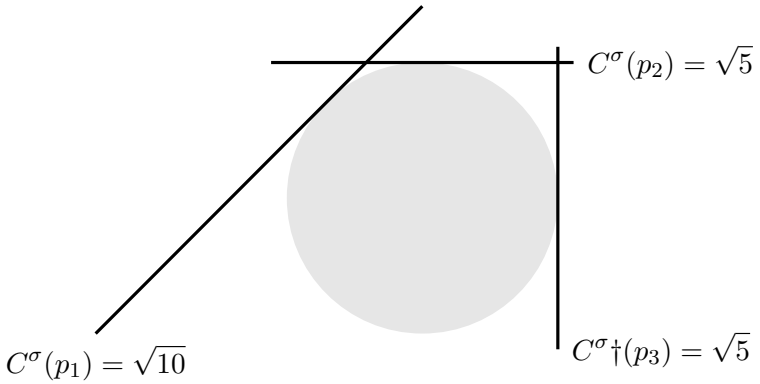


Figure 4.8 – $\mathbb{P}$-fonction support du disque unité

Exemple 4.19
Reprenons le $\mathbb{P}$-sous niveau de l'exemple 4.15 que nous appelons C. On suppose $\mathbb{P} = \{(x, y) :\mapsto \|(x, y)\|\}$, la $\mathbb{P}$-fonction support de C est donnée à la figure 4.9.

La proposition suivante montre que l'application $\mathbb{P}$-fonction support est bien une application d'abstraction.

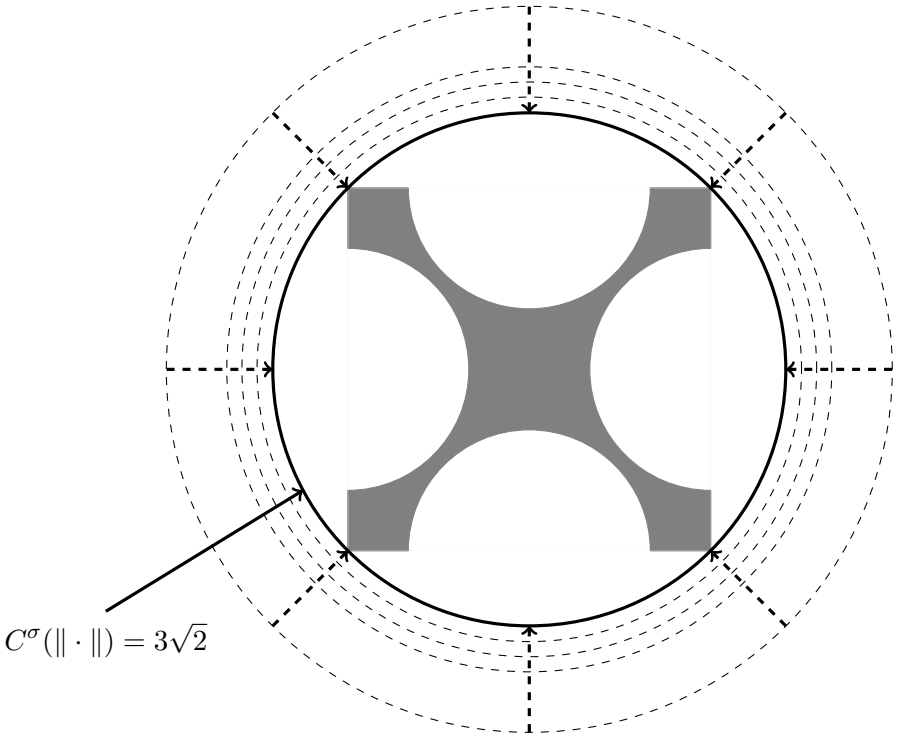


Figure 4.9 – $\mathbb{P}$-fonction support

Proposition 4.5 (*Monotonie de l'application $\mathbb{P}$-fonction support*)

L'application de $\wp(\mathbb{R}^d)$ dans $\overline{\mathbb{R}}^{\mathbb{P}}$, $C \mapsto C^\sigma$ est monotone.

Démonstration

Soient C, D deux sous-ensembles de $\mathbb{R}^d$, tels que $C \subseteq D$. On a, pour tout $p \in \mathbb{P}$, $\sup_{x\in C} p(x) \leq \sup_{x\in D} p(x)$ d'où la monotonie de l'application. ∎

Nous appelons fonction caractéristique convexe d'un ensemble $C \subseteq \mathbb{R}^d$, la fonction χ_C de $\mathbb{R}^d$ sur $\{0, +\infty\}$ définie par $\chi_C(x) = 0$ si $x \in C$ et $\chi_C(x) = +\infty$ si $x \notin C$ et bien évidemment si $C = \emptyset$, $\chi_C \equiv +\infty$. Pour rester dans l'esprit de l'analyse convexe, nous observons que la $\mathbb{P}$-fonction support est une $\mathbb{P}$-conjuguée de Fenchel où l'ensemble des formes linéaires de la conjuguée de Fenchel classique est remplacé par une configuration quelconque $\mathbb{P}$. Cette conjugaison de Fenchel sans linéarité a été étudiée par Moreau [Mor70].

Propriété 4.4 (*Caracterisation d'une $\mathbb{P}$-fonction support*)

Pour un sous-ensemble C de $\mathbb{R}^d$, nous pouvons réécrire la $\mathbb{P}$-fonction support C^σ de la manière suivante :

$$p :\mapsto C^\sigma(p) = \sup_{x\in\mathbb{R}^d} p(x) - \chi_C(x)$$

Remarque 4.7 Si C est vide, on retrouve que $C^\sigma \equiv -\infty$.

Pour construire une sémantique abstraite, on utilise, lorsqu'elle existe, une correspondance de Galois entre $\wp(\mathbb{R}^d)$ et le domaine abstrait. Ici, le domaine abstrait est l'ensemble $\overline{\mathbb{R}}^{\mathbb{P}}$ des fonctions de $\mathbb{P}$ dans $\overline{\mathbb{R}}$. Nous montrons grâce à la proposition 4.6 qu'une correspondance de Galois existe entre $\wp(\mathbb{R}^d)$ et $\overline{\mathbb{R}}^{\mathbb{P}}$.

Proposition 4.6 (*Correspondance de Galois entre $\overline{\mathbb{R}}^{\mathbb{P}}$ et $\wp(\mathbb{R}^d)$*)

La paire d'applications $v \mapsto v^{\mathbf{s}}$ et $C \mapsto C^{\sigma}$ forme une correspondance de Galois entre $\overline{\mathbb{R}}^{\mathbb{P}}$ et $\wp(\mathbb{R}^d)$.

Remarque 4.8 On réutilise la définition d'une correspondance de Galois de l'interprétation abstraite malgré la remarque 2.9.

Démonstration
Nous avons déjà montré aux propositions 4.4, 4.5 que les applications, $v \mapsto v^{\mathbf{s}}$ et $C \mapsto C^{\sigma}$ étaient monotones. Il suffit de montrer que pour tout $C \subseteq \mathbb{R}^d$, $v \in \overline{\mathbb{R}}^{\mathbb{P}}$,

$$C^{\sigma} \leq v \iff C \subseteq v^{\mathbf{s}}.$$

Tout d'abord, montrons que $C^{\sigma} \leq v \implies C \subseteq v^{\mathbf{s}}$. Soient $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ et $C \subseteq \mathbb{R}^d$, tels que, $C^{\sigma} \leq v$. Si C est vide, l'implication est triviale, nous supposons donc $C \neq \emptyset$. Soient $x \in C$ et $p \in \mathbb{P}$, on a $p(x) \leq C^{\sigma}(p)$, de $C^{\sigma} \leq v$, on déduit que $p(x) \leq v(p)$, p étant quelconque dans $\mathbb{P}$, on conclut que $x \in v^{\mathbf{s}}$. En conclusion, $C^{\sigma} \leq v \implies C \subseteq v^{\mathbf{s}}$.

Montrons maintenant la deuxième implication. Soient $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ et $C \subseteq \mathbb{R}^d$, tels que, $C \subseteq v^{\mathbf{s}}$. Si $v^{\mathbf{s}} = \emptyset$ alors $C = \emptyset$ et $C^{\sigma} \equiv -\infty$ et l'implication est montrée. Nous supposons maintenant que $v^{\mathbf{s}}$ est non vide. Soit $p \in \mathbb{P}$, on a $C^{\sigma}(p) = \sup_{x \in C} p(x) \leq \sup_{x \in v^{\mathbf{s}}} p(x)$. Or, pour tout $x \in v^{\mathbf{s}}$, $p(x) \leq v(p)$, d'où, $\sup_{x \in v^{\mathbf{s}}} p(x) \leq v(p)$ et par conséquent, $C^{\sigma}(p) \leq v(p)$, pour tout $p \in \mathbb{P}$. ∎

Les correspondances de Galois ont de très bonnes propriétés calculatoires que nous rappelons à la propriété 4.5.

Propriété 4.5 (*Propriétés des correspondances de Galois*)

Soient $C \subseteq \mathbb{R}^d$ et $v \in \overline{\mathbb{R}}^{\mathbb{P}}$, les propriétés suivantes sont vraies :

1. $C \subseteq (C^{\sigma})^{\mathbf{s}}$ et $(v^{\mathbf{s}})^{\sigma} \leq v$.
2. $C^{\sigma} = \min\{w \in \overline{\mathbb{R}}^{\mathbb{P}} \mid C \subseteq w^{\mathbf{s}}\}$ et $v^{\mathbf{s}} = \max\{D \subseteq \mathbb{R}^d \mid D^{\sigma} \leq v\}$.
3. $((C^{\sigma})^{\mathbf{s}})^{\sigma} = C^{\sigma}$ et $((v^{\mathbf{s}})^{\sigma})^{\mathbf{s}} = v^{\mathbf{s}}$.

Démonstration

1. Si C est vide alors l'inclusion est évidente. Supposons C non vide et soit $x \in C$, on a, pour tout $p \in \mathbb{P}$, $p(x) \leq C^{\sigma}(p)$ d'où, $x \in (C^{\sigma})^{\mathbf{s}}$, et $C \subseteq (C^{\sigma})^{\mathbf{s}}$.
   Si $v^{\mathbf{s}}$ est vide alors $(v^{\mathbf{s}})^{\sigma} \equiv -\infty$, et l'inégalité est vraie. Supposons donc $v^{\mathbf{s}}$ non vide et soit $p \in \mathbb{P}$, pour tout $x \in v^{\mathbf{s}}$, $p(x) \leq v(p)$ et par conséquent, $\sup_{x \in v^{\mathbf{s}}} p(x) \leq v(p)$, en conclusion $(v^{\mathbf{s}})^{\sigma} \leq v$.
2. D'après, la première assertion, $(v^{\mathbf{s}})^{\sigma} \leq v$ et $C \subseteq (C^{\sigma})^{\mathbf{s}}$ d'où $v^{\mathbf{s}} \in \{D \subseteq \mathbb{R}^d \mid D^{\sigma} \leq v\}$ et $C^{\sigma} \in \{w \in \overline{\mathbb{R}}^{\mathbb{P}} \mid C \subseteq w^{\mathbf{s}}\}$. Maintenant, soient $u \in \{w \in \overline{\mathbb{R}}^{\mathbb{P}} \mid C \subseteq w^{\mathbf{s}}\}$ et $E \in \{D \subseteq \mathbb{R}^d \mid D^{\sigma} \leq v\}$, d'après la proposition 4.6, $C \subseteq u^{\mathbf{s}}$ est équivalent à $C^{\sigma} \leq u$ et $E^{\sigma} \leq v$ équivaut à $E \subseteq v^{\mathbf{s}}$. On conclut que $C^{\sigma} = \min\{w \in \overline{\mathbb{R}}^{\mathbb{P}} \mid C \subseteq w^{\mathbf{s}}\}$ et $v^{\mathbf{s}} = \max\{D \subseteq \mathbb{R}^d \mid D^{\sigma} \leq v\}$.

3. D'après la première assertion, $C \subseteq (C^\sigma)^{\mathbf{s}}$ et $((v^{\mathbf{s}})^\sigma \leq v$, et grâce aux propositions 4.4, 4.5, on obtient $C^\sigma \leq ((C^\sigma)^{\mathbf{s}})^\sigma \leq C^\sigma$ et $v^{\mathbf{s}} \subseteq ((v^{\mathbf{s}})^\sigma)^{\mathbf{s}} \subseteq v^{\mathbf{s}}$. En conclusion, $((C^\sigma)^{\mathbf{s}})^\sigma = C^\sigma$ et $((v^{\mathbf{s}})^\sigma)^{\mathbf{s}} = v^{\mathbf{s}}$. ∎

4.3 Treillis des ensembles et fonctions $\mathbb{P}$-convexes

Nous avons vu, dans le chapitre 2, qu'un domaine abstrait devait être au moins un CPO pour qu'on puisse calculer les points fixes d'applications continues (au sens des CPO). Dans cette section, nous allons établir que l'ensemble des parties dites $\mathbb{P}$-convexes forme un treillis complet. Il en sera de même pour les fonctions dite $\mathbb{P}$-convexes. Tout d'abord, nous introduisons la convexité abstraite.

4.3.1 Convexité abstraite

La convexité abstraite, pour laquelle on pourra consulter par exemple [Rub00, Sin97], est une généralisation de la convexité classique. Il s'agit de définir une notion de convexité sans structures algébriques. Ce type de convexité a été introduite par Moreau [Mor70].

Fonctions convexes abstraites

En analyse convexe, les fonctions classiquement utilisées sont les fonctions convexes semi-continue inférieurement. En effet, un théorème classique d'analyse convexe permet de caractériser les fonctions convexes s.c.i comme le supremum point par point des minorantes affines. Ce théorème est illustrée par la figure 4.10.

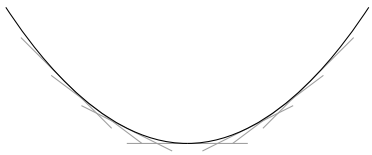


Figure 4.10 – La fonction de l'exemple 4.2 vue comme supremum de fonctions affines

La convexité abstraite reprend le théorème illustré par la figure 4.10. En convexité abstraite, les fonctions convexes abstraites sont des suprema de minorantes quelconques mais préalablement fixées. Plus formellement, comme dans notre cas, on se munit d'un ensemble H de fonctions d'un ensemble (sans structure linéaire) E dans $\overline{\mathbb{R}}$. Puis, on définit la H-enveloppe convexe de $f : E \mapsto \overline{\mathbb{R}}$ comme le supremum des éléments de H qui sont plus petits que f, point par point. Finalement, une fonction f est dite convexe abstraite par rapport à H (on dit aussi H-convexe) si elle est égale à sa H-enveloppe convexe.

Définition 4.15 (*Fonctions convexes abstraites*)

Une fonction $f : E \mapsto \overline{\mathbb{R}}$ est H-convexe où $\emptyset \neq H \subseteq \overline{\mathbb{R}}^E$ si pour tout $x \in E$,

$$f(x) = \sup\{h(x) \mid h \in H,\ h(y) \leq f(y),\ \forall\, y \in E\}$$

En interprétation abstraite, quand les fonctions de concrétisation et d'abstraction forment une correspondance de Galois, nous nous intéressons aux éléments dits clos. Dans notre cas, les éléments clos correspondent aux fonctions $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ qui sont la $\mathbb{P}$-fonction support de leur $\mathbb{P}$-sous niveau, et par la formulation d'un $\mathbb{P}$-sous niveau en terme d'inégalité fonctionnelle de la propriété 4.3, les éléments clos correspondent aux H-convexes abstraits en identifiant l'ensemble abstrait E de la définition 4.15 à $\mathbb{P}$ et l'ensemble de fonctions H à $\{\mathbf{e}_x,\ x \in \mathbb{R}^d\}$.

Ensembles convexes abstraits

Les analogues ensemblistes des fonctions convexes semi-continue inférieurement sont les ensembles convexes fermés. On peut également caractériser les ensembles convexes fermés à partir des formes linéaires. Un ensemble convexe fermé C est l'intersection de demi-espaces fermés le contenant c'est-à-dire des ensembles de la forme $\{y \in E \mid h(y) \leq \alpha\}$ où $h \in E'$ et $\alpha \in \mathbb{R}$. Cette caractérisation est équivalente au théorème de séparation stricte : pour tout point $z \notin C$, il existe une forme linéaire $h \in E'$ telle que $h(z) \geq \alpha > \beta \geq h(x),\ \forall\, x \in C$. Le théorème de séparation stricte illustré par la figure 4.11.

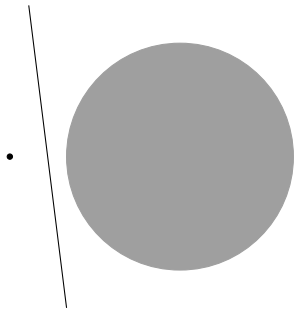


Figure 4.11 – Séparation forte d'un point et d'un ensemble convexe fermé

La convexité abstraite reprend l'idée de la séparation stricte et un ensemble C est dit H-convexe, où H est toujours un ensemble de fonctions d'un ensemble E quelconque dans $\overline{\mathbb{R}}$, tout point extérieur à C peut être séparé strictement de C par un élément de H. La définition formelle est donnée 4.16.

Définition 4.16 (*Ensemble convexe abstrait*)

Soient E un ensemble et H un sous-ensemble non vide de $\overline{\mathbb{R}}^E$. Un ensemble C est H-convexe si tout point $c \in C$ vérifie :

$$\forall\, h \in H,\ h(c) \leq \sup_{y \in C} h(y)\ .$$

En prenant, $E = \mathbb{R}^d$, $H = \mathbb{P}$, un ensemble C est H-convexe s'il est égal au $\mathbb{P}$-sous niveau de sa $\mathbb{P}$-fonction support.

4.3.2 Construction des treillis complets

En interprétation abstraite, l'opération de clôture est importante, elle garantit le calcul de la meilleure concrétisation d'un élément abstrait et de la meilleure abstraction d'un élément

concret. En général, l'opération de clôture est donnée par une correspondance de Galois. Dans notre cas, notre opération de clôture, qui est donnée par la correspondance de Galois 4.6 se ramène à un calcul d'enveloppe convexe abstraite.

Définition 4.17 ($\mathbb{P}$-*enveloppe convexe de fonctions*)

Soit $v \in \overline{\mathbb{R}}^{\mathbb{P}}$, on appelle $\mathbb{P}$-enveloppe convexe de v, la fonction notée $\mathrm{vex}_{\mathbb{P}}(v)$ définie par $\mathrm{vex}_{\mathbb{P}}(v) = (v^{\mathbf{s}})^{\sigma}$ c'est-à-dire, pour $p \in \mathbb{P}$:

$$\begin{aligned}(\mathrm{vex}_{\mathbb{P}}(v))(p) &= \sup\{p(x) \mid x \in \mathbb{R}^d,\ q(x) \leq v(q),\ \forall q \in \mathbb{P}\} \\ &= \sup\{\mathbf{e}_x(p) \mid x \in \mathbb{R}^d,\ \mathbf{e}_x \in \mathbb{R}^{\mathbb{P}},\ \mathbf{e}_x \leq v\}\ .\end{aligned}$$

Le calcul d'une $\mathbb{P}$-enveloppe sert à réduire les degrés de liberté trop larges sur les fonctions $p \in \mathbb{P}$. Cette propriété de réduire les bornes est mathématiquement illustrée par le point 1 de la propriété 4.5.

Exemple 4.20
Dès qu'il existe $p \in \mathbb{P}$, tel que $v(p) = -\infty$ alors $\mathrm{vex}_{\mathbb{P}}(v) \equiv -\infty$.

Exemple 4.21
Soit la configuration $\mathbb{P} = \{p_1 : x, y \mapsto x^2+y^2,\ p_2 : x, y \mapsto x,\ p_3 : x, y \mapsto -x,\ p_4 : x, y \mapsto y,\ p_5 : x, y \mapsto -y\}$. Soit $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ défini par $v(p_1) = 4,\ v(p_2) = 2,\ v(p_3) = 1,\ v(p_4) = 3,\ v(p_5) = 4$. La $\mathbb{P}$-enveloppe convexe $\mathrm{vex}_{\mathbb{P}}(v)$ de v est la fonction définie par $\mathrm{vex}_{\mathbb{P}}(v)(p_1) = 4,\ \mathrm{vex}_{\mathbb{P}}(v)(p_2) = 2,\ \mathrm{vex}_{\mathbb{P}}(v)(p_3) = 1,\ \mathrm{vex}_{\mathbb{P}}(v)(p_4) = 2,\ \mathrm{vex}_{\mathbb{P}}(v)(p_5) = 2$. La construction de $\mathrm{vex}_{\mathbb{P}}(v)$ est décrite par la figure 4.12

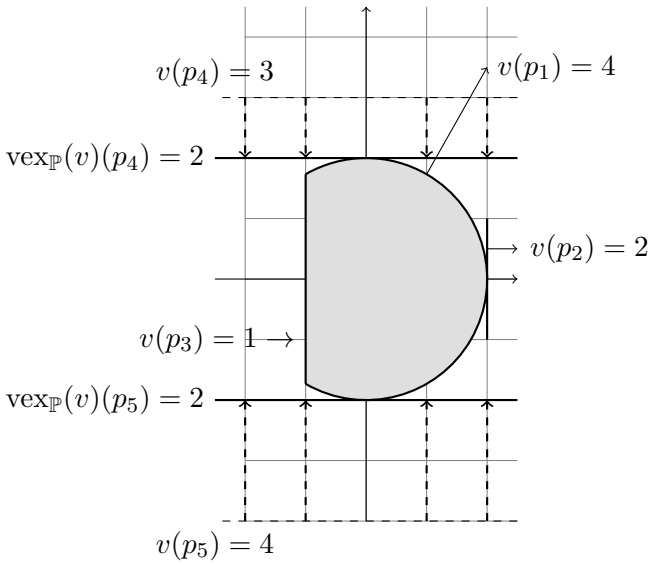


Figure 4.12 – Construction de la $\mathbb{P}$-enveloppe de la fonction v définie à l'exemple 4.21.

De manière similaire, l'opération de clôture pour les sous-ensembles de $\mathbb{R}^d$ correspond à un calcul d'enveloppe convexe abstraite de sous-ensemble de $\mathbb{R}^d$.

Définition 4.18 (*$\mathbb{P}$-enveloppe convexe d'ensembles*)

Soit un ensemble $C \subseteq \mathbb{R}^d$, on appelle $\mathbb{P}$-enveloppe convexe de C, l'ensemble noté $\text{vex}_{\mathbb{P}}(C)$ définie par $\text{vex}_{\mathbb{P}}(C) = (C^\sigma)^{\mathbf{s}}$ c'est-à-dire :

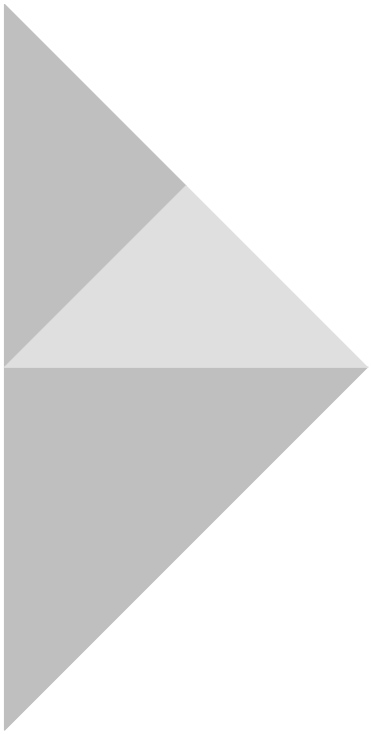
$$\text{vex}_{\mathbb{P}}(C) = \{x \in \mathbb{R}^d \mid p(x) \leq \sup_{y \in C} p(y) \ \forall\, p \in \mathbb{P}\} \ .$$

Remarque 4.9 D'après le point 3 de la propriété 4.5, pour tout $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ et $C \subseteq \mathbb{R}^d$, $(\text{vex}_{\mathbb{P}}(v))^{\mathbf{s}} = v^{\mathbf{s}}$ et $(\text{vex}_{\mathbb{P}}(C))^\sigma = C^\sigma$.

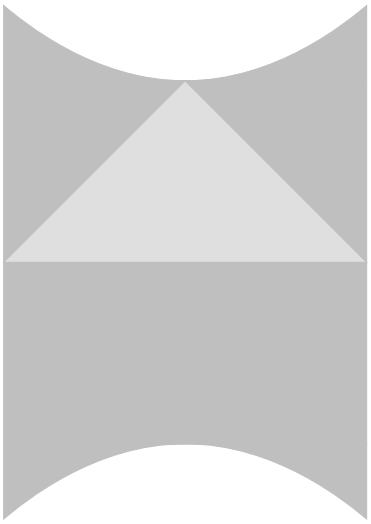
Exemple 4.22
Soit $\mathbb{P}$ la configuration linéaire et soit $C \subseteq \mathbb{R}^d$, la $\mathbb{P}$-enveloppe convexe $\text{vex}_{\mathbb{P}}(C)$ de C correspond à la classique enveloppe convexe de C.

Exemple 4.23
Lorsque $\mathbb{P} = \{(x, y) :\mapsto \|(x, y)\|\}$. Le disque de rayon $3\sqrt{2}$ est la $\mathbb{P}$-enveloppe convexe de l'ensemble défini à l'exemple 4.15.

Exemple 4.24
Nous reprenons le triangle défini à l'exemple 4.13. La figure représente deux $\mathbb{P}$-enveloppe convexe dans deux ensembles $\mathbb{P}$ de fonctions de base différentes. On pose $\mathbb{P}_1 = \{(x, y) \mapsto x - y,\ (x, y) \mapsto x + y,\ (x, y) \mapsto -x\}$ et $\mathbb{P}_2 = \{(x, y) \mapsto y^2 - x^2,\ (x, y) \mapsto x,\ (x, y) \mapsto x\}$.



(a) $\{x - y \leq 3,\ y + x \leq 3,\ -x \leq 3\}$



(b) $\{y^2 - x^2 \leq 9,\ x \leq 3,\ -x \leq 3\}$

Figure 4.13 – Deux $\mathbb{P}$-enveloppes du triangle de la figure 4.13

Nous unissons la notion d'éléments clos au sens d'une correspondance de Galois et la notion d'éléments convexes abstraits dans une notion de $\mathbb{P}$-convexité. Nous utilisons le terme $\mathbb{P}$-convexe, en raison de la très forte dépendance de l'opération de clôture par rapport à l'ensemble $\mathbb{P}$ des fonctions de base.

Définition 4.19 (*Fonction $\mathbb{P}$-convexe*)

La fonction $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ est $\mathbb{P}$-convexe si $v = \text{vex}_{\mathbb{P}}(v)$.

Exemple 4.25
La fonction identiquement égale à $-\infty$ et la fonction qui à tout élément $p \in \mathbb{P}$ associe son supremum sur $\mathbb{R}^d$ sont des fonctions $\mathbb{P}$-convexes.

Exemple 4.26
Toute $\mathbb{P}$-fonction support est une fonction $\mathbb{P}$-convexe. Toute $\mathbb{P}$-enveloppe de fonction est une fonction $\mathbb{P}$-convexe.

Exemple 4.27
La fonction v de l'exemple 4.15 est $\mathbb{P}$-convexe.

Exemple 4.28
Considérons $\mathbb{P} = \{x :\mapsto x^n,\ n \in \mathbb{N}\}$, la fonction $v(p_n) = 0$ si n est impair et 1 sinon est $\mathbb{P}$-convexe. En effet, $v^{\mathbf{s}} = [-1, 0]$ et $\sup_{x \in [-1,0]} x^n = 1$ si n est pair 0 sinon.

Exemple 4.29
Soit $\mathbb{P}$ l'ensemble de toutes les fonctions de $\mathbb{R}^d$ dans $\mathbb{R}$ et soit $a \in \mathbb{R}^d$, on pose, pour $p \in \mathbb{P}$, $v(p) = p(a)$ alors v est $\mathbb{P}$-convexe. En effet, $a \in v^{\mathbf{s}}$ et pour tout $x \in v^{\mathbf{s}}$, pour tout $q \in \mathbb{P}$, $q(x) \leq q(a)$ et par conséquent $(\text{vex}_{\mathbb{P}}(v))(q) = q(a) = v(q)$ pour tout $q \in \mathbb{P}$.

Définition 4.20 (*Ensemble $\mathbb{P}$-convexe*)

Un ensemble $C \subseteq \mathbb{R}^d$ est un ensemble $\mathbb{P}$-convexe si $C = \text{vex}_{\mathbb{P}}(C)$.

Définition 4.21 ($\mathbb{P}$-*polyèdre*)

Un ensemble $C \subseteq \mathbb{R}^d$ est un $\mathbb{P}$-polyèdre s'il est $\mathbb{P}$-convexe et s'il existe un élément $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ presque infini tel que $C = v^{\mathbf{s}}$.

Exemple 4.30
L'ensemble vide et $\mathbb{R}^d$ sont des ensembles $\mathbb{P}$-convexes.

Exemple 4.31
Tout $\mathbb{P}$-sous niveau est un ensemble $\mathbb{P}$-convexe. Toute $\mathbb{P}$-enveloppe d'ensemble est un ensemble $\mathbb{P}$-convexe.

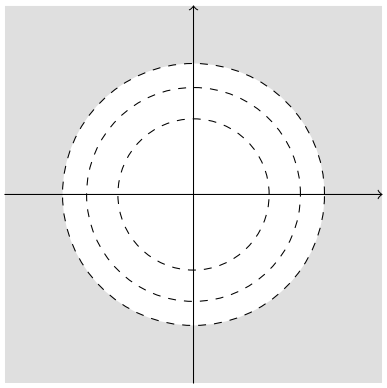
Exemple 4.32
Nous reprenons l'exemple 4.13 le triangle de la figure 4.6 est un ensemble $\mathbb{P}$-convexe avec $\mathbb{P} = \{p_1 : (x, y) \mapsto y - x, p_2 : (x, y) \mapsto y + x, p_3 : (x, y) \mapsto -y\}$.

Exemple 4.33
Soit $(A_j)_{j \in J}$ une famille de sous-ensembles non vides de $\mathbb{R}^d$. On définit, pour $i \in \{1, \ldots, J\}$, la fonction de $\mathbb{R}^d$ dans $\mathbb{R}$:

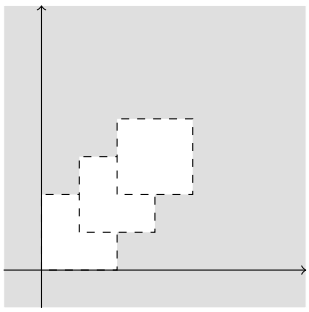
$$x \mapsto p_i(x) = \sum_{k=1}^{i} (\|x\| + 1) \mathbb{1}_{A_k}(x)$$

où $\mathbb{1}_{A_k}(x)$ est la fonction indicatrice de l'ensemble A_k qui vaut 0 si $x \notin A_k$ et 1 sinon. La configuration $\mathbb{P}$ est l'ensemble de fonctions $\{p_i \mid i \in \{1, \dots, J\}\}$.

L'ensemble $\bigcap_{k=1}^{J} A_j^c$ est $\mathbb{P}$-convexe et $(\bigcap_{k=1}^{J} A_j^c)^\sigma \equiv 0$. La figure 4.14 représente cet ensemble dans deux cas, dans le premier cas, les ensembles A_j sont des boules centrées en 0 dans le deuxième, les ensembles A_j sont des rectangles.



(a)
$A_1 = B(0,1), A_2 = B(0,2), A_3 = B(0,3)$



(b)
$A_1 = [0,1]^2, A_2 = [0.5, 0.5]^2, A_3 = [1,2]^2$

Figure 4.14 – La partie grisée est l'ensemble $\mathbb{P}$-convexe de l'exemple 4.33

Exemple 4.34
Si $\mathbb{P}$ est la configuration linéaire classique alors tout polyèdre est un $\mathbb{P}$-polyèdre.

Exemple 4.35
Soit $(A_j)_{j \geq 1}$ une famille infinie de sous-ensembles non vides de $\mathbb{R}^d$ telle que, pour tout $j \leq K$, A_j soit borné et pour tout $j \geq K+1$, A_j soit non borné et $\bigcap_{k=K+1}^{\infty} A_j$ soit non vide. On reprend la famille de fonction p_i définie à l'exemple 4.33 avec $i \in \mathbb{N}^*$ et on prend $\mathbb{P}$ la configuration $\{p_i \mid i \geq 1\}$.

L'ensemble $\left(\bigcap_{k=1}^{K} A_j^c\right) \cap \left(\bigcap_{k=K+1}^{\infty} A_j\right)$ est un $\mathbb{P}$-polyèdre et $\left(\bigcap_{k=1}^{K} A_j^c \cap \bigcap_{k=K+1}^{\infty} A_j\right)^\sigma (p_i) = 0$ si $i \leq K$ et $+\infty$ sinon.

On note $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ l'ensemble des fonctions $\mathbb{P}$-convexes de $\overline{\mathbb{R}}^{\mathbb{P}}$ et $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ l'ensemble des parties $\mathbb{P}$-convexes de $\mathbb{R}^d$.

A présent, nous définissons des opérations de treillis sur $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ et $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$. Nous allons définir les opérations de borne inférieure et de borne supérieure sur des paires d'éléments de $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ et de $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$.

En convexité classique, l'infimum de fonctions convexes n'est pas convexe en général, ceci explique pourquoi nous prenons la $\mathbb{P}$-enveloppe convexe de l'infimum. Le problème de l'infimum est illustré par le contre-exemple 4.1.

Définition 4.22 (*Opérations de treillis dans* $\mathbf{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$)

Soient v, w deux éléments de $\overline{\mathbb{R}}^{\mathbb{P}}$. On note $\inf(v,w)$ et $\sup(v,w)$ les fonctions définies respectivement par, $p \mapsto \inf(v(p), w(p))$ et $p \mapsto \sup(v(p), w(p))$.

On munit $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ des opérations suivantes :

$$v \vee w = \sup(v, w) \tag{4.3}$$

$$v \wedge w = (\inf(v, w)^{\mathbf{s}})^{\sigma} \tag{4.4}$$

Contre-exemple 4.1
Considérons la configuration $\mathbb{P}$ de l'exemple 4.21. Soit $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ défini par $v(p_1) = 4,\ v(p_2) = 1,\ v(p_3) = 1,\ v(p_4) = 2,\ v(p_5) = 2$ et $w \in \overline{\mathbb{R}}^{\mathbb{P}}$ défini par $w(p_1) = 2,\ w(p_2) = \sqrt{2},\ w(p_3) = \sqrt{2},\ w(p_4) = 1,\ w(p_5) = 1\}$. Les éléments v et w appartient à $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ mais $z = \inf(v, w)$ n'appartient pas à $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$. En effet, $z(p_1) = 2,\ z(p_2) = 1,\ z(p_3) = 1,\ z(p_4) = 1,\ z(p_1) = 1$ et donc, $\sup\limits_{(x,y)\in z^{\star}} p_1(x, y) = 1 < 2$.

La convexité classique n'est pas stable par union, pour stabiliser l'union, nous devons prendre la $\mathbb{P}$-enveloppe convexe de l'union. Nous illustrons l'instabilité de l'union sur le contre-exemple 4.2.

Définition 4.23 (*Opérations de treillis dans* $\mathbf{Vex}_{\mathbb{P}}(\mathbb{R}^{\boldsymbol{d}})$)

Soient C, D deux sous-ensembles de $\mathbb{R}^d$. Nous rappelons que $\cup$ et $\cap$ sont respectivement les opérations d'union et d'intersection.

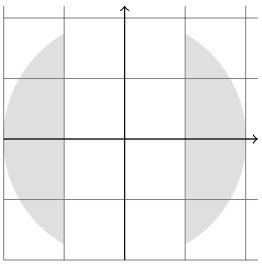
On munit $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ des opérations suivantes :

$$C \sqcup D = ((C \cup D)^{\sigma})^{\mathbf{s}} \tag{4.5}$$

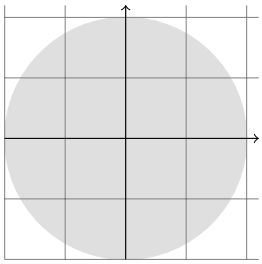
$$C \sqcap D = C \cap D \tag{4.6}$$

Contre-exemple 4.2
Nous reprenons la configuration de l'exemple 4.21. On pose $v(p_1) = 4,\ v(p_2) = -1,\ v(p_3) = 2,\ v(p_4) = 2,\ v(p_5) = 2$ et $w(p_1) = 4,\ w(p_2) = 2,\ w(p_3) = -1,\ w(p_4) = 2,\ w(p_5) = 2$. Les ensembles $v^{\mathbf{s}}$ et $w^{\mathbf{s}}$ sont $\mathbb{P}$-convexes mais $v^{\mathbf{s}} \cup w^{\mathbf{s}}$ n'est pas $\mathbb{P}$-convexe. En effet, $(v^{\mathbf{s}} \cup w^{\mathbf{s}})^{\sigma}(p_1) = \sup\limits_{(x,y)\in v^{\mathbf{s}}\cup w^{\mathbf{s}}} p_1(x, y) = 4$ et $(v^{\mathbf{s}} \cup w^{\mathbf{s}})^{\sigma}(p_i) = \sup\limits_{(x,y)\in v^{\mathbf{s}}\cup w^{\mathbf{s}}} p_i(x, y) = 2\ \forall\, i \in \{2, 3, 4, 5\}$ et donc $\mathrm{vex}_{\mathbb{P}}((v^{\mathbf{s}} \cup w^{\mathbf{s}}))$ est le disque de rayon de 2. L'exemple est illustré par la figure 4.15.

Remarque 4.10 Le contre-exemple 4.2 montre que l'opération de treillis sup dans $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ n'est pas l'enveloppe convexe classique. L'enveloppe convexe au sens classique est l'ensemble de la figure 4.16.



(a) L'ensemble $v^{\mathbf{s}} \cup w^{\mathbf{s}}$



(b) La $\mathbb{P}$-enveloppe convexe de $v^{\mathbf{s}} \cup w^{\mathbf{s}}$

Figure 4.15 – L'illustration du contre-exemple 4.2.

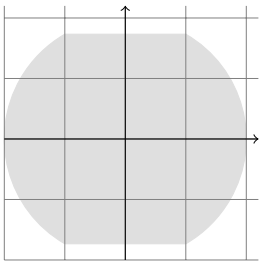


Figure 4.16 – L'enveloppe convexe classique de $v^{\mathbf{s}} \cup w^{\mathbf{s}}$.

Théorème 4.2 (*Structure algébrique de* $\mathbf{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ *et* $\mathbf{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$)

$(\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}}), \wedge, \vee)$ et $(\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d), \sqcap, \sqcup)$ sont des treillis complets.

Le point important du théorème précédent est de vérifier que le supremum de fonctions $\mathbb{P}$-convexes est une fonction $\mathbb{P}$-convexe et que l'intersection d'ensembles $\mathbb{P}$-convexes est un ensemble $\mathbb{P}$-convexe. En analyse convexe classique, ce résultat est classique.

Lemme 4.1 (*Stabilité de la* $\mathbb{P}$*-convexité*)

Soient $\{v_i\}_{i\in I}$ et $\{X_j\}_{j\in J}$ des familles respectives d'applications de fonctions $\mathbb{P}$-convexes et d'ensembles $\mathbb{P}$-convexes. Alors $\sup_{i\in I} v_i$ et $\bigcap_{j\in J} X_j$ sont $\mathbb{P}$-convexes.

Démonstration

On a, en utilisant la propriété 4.5, $\mathrm{vex}_{\mathbb{P}}(\sup_{i\in I} v_i) \leq \sup_{i\in I} v_i$. En utilisant la monotonie de $C \mapsto C^{\sigma}$ et $v \mapsto v^{\mathbf{s}}$, on obtient pour tout $i \in I$, $v_i = \mathrm{vex}_{\mathbb{P}}(v_i) \leq \mathrm{vex}_{\mathbb{P}}(\sup_{i\in I} v_i)$ on conclut que $\sup_{i\in I} v_i \leq \mathrm{vex}_{\mathbb{P}}(\sup_{i\in I} v_i) \leq \sup_{i\in I} v_i$ et finalement, $\mathrm{vex}_{\mathbb{P}}(\sup_{i\in I} v_i) = \sup_{i\in I} v_i$.

La preuve est la même pour les ensembles convexes abstraits. ∎

Démonstration (Démonstration du Théorème 4.2)

Montrons d'abord que $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ est un treillis complet. $v(p) = \sup\{p(x) \mid x \in \mathbb{R}^d\}$ est le plus grand élément de $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$. De plus, $\sup\{p(x) \mid \forall q,\ q(x) \leq -\infty\} = \sup_{\emptyset} p(x) = -\infty$, donc $v = -\infty \in \mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$. La famille $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ possède un plus petit et un plus grand élément.

Le Lemme 4.1 nous permet de conclure. Le fait que $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ soit un treillis complet résulte du même type d'arguments. ∎

Proposition 4.7

Propriétés des fonctions $\mathbb{P}$-convexes Soit $v \in \mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ alors :

1. $v^{\mathbf{s}} = \emptyset \iff v \equiv -\infty$;
2. $\exists\, p \in \mathbb{P}$ tel que $v(p) = -\infty \iff v \equiv -\infty$;
3. $v^{\mathbf{s}} = \mathbb{R}^d \iff v(p) = \sup\limits_{x \in \mathbb{R}^d} p(x),\ \forall\, p \in \mathbb{P}$.

Démonstration

1. Si $v \equiv -\infty$ alors clairement $v^{\mathbf{s}} = \emptyset$. Si $v^{\mathbf{s}} = \emptyset$ et $v \in \mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ alors pour tout $p \in \mathbb{P}$, $v(p) = \sup\limits_{x \in v^{\mathbf{s}}} p(x) = -\infty$.
2. Le sens $v \equiv -\infty \implies \exists\, p \in \mathbb{P}$ tel que $v(p) = -\infty$ est évident. S'il existe $p \in \mathbb{P}$ tel que $v(p) = -\infty$ alors $v^{\mathbf{s}} = \emptyset$ et d'après le point 1, $v \equiv -\infty$.
3. Si $v^{\mathbf{s}} = \mathbb{R}^d$ alors clairement $v(p) = \sup\limits_{x \in \mathbb{R}^d} p(x),\ \forall\, p \in \mathbb{P}$. L'égalité $v(p) = \sup\limits_{x \in \mathbb{R}^d} p(x),\ \forall\, p \in \mathbb{P}$ est équivalente à $v = (\mathbb{R}^d)^{\sigma}$ d'où, $v^{\mathbf{s}} = ((\mathbb{R}^d)^{\sigma})^{\mathbf{s}}$ et comme $\mathbb{R}^d$ est $\mathbb{P}$-convexe alors $v^{\mathbf{s}} = \mathbb{R}^d$. ∎

Proposition 4.8 (*Isomorphisme entre* $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ *et* $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$)

$\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ et $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ sont isomorphes. L'isomorphisme entre $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ et $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ est donné par l'application $\mathbb{P}$-fonction support et l'application inverse par l'application $\mathbb{P}$-sous niveau.

Démonstration

Montrons que l'application $\mathbb{P}$-fonction support de $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ dans $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ est injective. Soit $C, D \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ tels que $C^{\sigma} = D^{\sigma}$ ceci implique que $(C^{\sigma})^{\mathbf{s}} = (D^{\sigma})^{\mathbf{s}}$ d'où $C = D$. Montrons que l'application est surjective et soit $v \in \mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$, on a, par le point 3 de la propriété 4.5, $v^{\mathbf{s}} \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$, et donc l'application $\mathbb{P}$-fonction support est surjective. Finalement, comme, pour tout $C \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$, $C = (C^{\sigma})^{\mathbf{s}}$ alors $(\cdot^{\sigma})^{-1} = \cdot^{\mathbf{s}}$. ∎

Nous concluons cette section par une dernière caractérisation des $\mathbb{P}$-enveloppes convexes par rapport aux éléments $\mathbb{P}$-convexes.

Proposition 4.9 ($\mathbb{P}$-*enveloppe et éléments* $\mathbb{P}$-*convexes*)

Pour tout $v \in \overline{\mathbb{R}}^{\mathbb{P}}$, pour tout $C \subseteq \mathbb{R}^d$.

1. $\mathrm{vex}_{\mathbb{P}}(v)(p) = \sup\{w \in \mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}}) \mid w \leq v\}$
2. $\mathrm{vex}_{\mathbb{P}}(C) = \bigcap\{D \mid D \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d),\ C \subseteq D\}$.

Démonstration

Les deux points se démontrent de la même manière. Pour changer, nous démontrons la proposition pour les ensembles. Posons $K = \cap\{D \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d),\ C \subseteq D\}$. Par le point 1 de la propriété 4.5, $C \subseteq \mathrm{vex}_{\mathbb{P}}(C)$ et $\mathrm{vex}_{\mathbb{P}}(C) \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ donc $K \subseteq \mathrm{vex}_{\mathbb{P}}(C)$. La monotonie des applications $v \mapsto v^{\mathbf{s}}$ et $C \mapsto C^{\sigma}$ implique que $\mathrm{vex}_{\mathbb{P}}(C) \subseteq D$ pour tout $D \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$, tels que $C \subseteq D$, l'inclusion reste vraie pour l'intersection de ces D d'où $\mathrm{vex}_{\mathbb{P}}(C) \subseteq K$. ∎

La proposition 4.9 montre que la $\mathbb{P}$-enveloppe convexe d'un élément $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ est la plus grande fonction $\mathbb{P}$-convexe minorant v. De même, la $\mathbb{P}$-enveloppe convexe d'un ensemble C est le plus petit ensemble $\mathbb{P}$-convexe contenant C.

4.3.3 $\mathbb{P}$-convexité et convexité classique

Par analogie avec la convexité classique, on montre qu'un ensemble est $\mathbb{P}$-convexe ssi tout point à l'extérieur de l'ensemble peut être séparé (fortement) de l'ensemble par une fonction de base $p \in \mathbb{P}$.

Proposition 4.10 (*Séparation forte en $\mathbb{P}$-convexité*)

Un ensemble C est $\mathbb{P}$-convexe ssi pour tout $z \notin C$, il existe $p \in \mathbb{P}$, tel que $p(z) > \sup_{y \in C} p(y)$.

Démonstration

Supposons que C est $\mathbb{P}$-convexe alors par définition $C = \{x \in \mathbb{R}^d \mid p(x) \leq \sup_{y \in C} p(y),\ \forall p \in \mathbb{P}\}$. Soit $z \notin C$, on a donc $z \notin \{x \in \mathbb{R}^d \mid p(x) \leq \sup_{y \in C} p(y),\ \forall p \in \mathbb{P}\}$ et donc $\exists\, q \in \mathbb{P}$ tel que $q(z) > \sup_{y \in C} q(y)$. Maintenant supposons que $z \notin C$ implique qu'il existe $q \in \mathbb{P}$ tel que $q(z) > \sup_{y \in C} q(y)$ ceci équivaut à $z \notin \mathrm{vex}_{\mathbb{P}}(C)$ et donc $\mathrm{vex}_{\mathbb{P}}(C) \subseteq C$ et d'après le point 1 de la propriété 4.5, on conclut que $C = \mathrm{vex}_{\mathbb{P}}(C)$. ∎

Nous rappelons que d'après le théorème de séparation forte en analyse convexe affirme que pour tout convexe fermé C, $x \notin C$ implique qu'il existe $p \in \mathbb{R}^d$ tel que $p \cdot x > \sup_{y \in C} p \cdot y$.

Nous rappelons, dans le corollaire suivant, que, lorsque $\mathbb{P}$ est la configuration linéaire classique, les ensembles $\mathbb{P}$-convexes s'identifient aux ensembles fermés convexes au sens classique de $\mathbb{R}^d$.

Corollaire 4.1 (*Ensemble $\mathbb{P}$-convexes dans la configuration linéaire classique*)

Si $\mathbb{P}$ est la configuration linéaire classique alors $\mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$ est exactement l'ensemble des ensembles convexes fermés de $\mathbb{R}^d$.

Démonstration

Soit $C \in \mathrm{Vex}_{\mathbb{P}}(\mathbb{R}^d)$, alors par définition :

$$C = \{x \in \mathbb{R}^d \mid p \cdot x \leq C^\sigma(p),\ \forall\, p \in \mathbb{R}^d\} = \bigcap_{p \in \mathbb{R}^d} \{x \in \mathbb{R}^d \mid p \cdot x \leq C^\sigma(p)\}\ .$$

C est donc une intersection d'ensemble convexe fermé, il est par conséquent convexe et fermé.

Supposons que maintenant C est convexe fermé. Si $C = \mathbb{R}^d$, alors C est $\mathbb{P}$-convexe. Supposons $C \neq \mathbb{R}^d$ et soit $x \notin C$, alors, par le théorème de séparation forte, il existe $p \in \mathbb{R}^d$ tel que $p \cdot x > \sup_{y \in C} p \cdot y$ et par la proposition 4.10 C est $\mathbb{P}$-convexe. ∎

Nous cherchons désormais à identifier les fonctions $\mathbb{P}$-convexes lorsque $\mathbb{P}$ est la configuration linéaire. Tout d'abord, nous rappelons la définition d'une fonction positivement homogène de degré 1.

Définition 4.24 (*Fonction homogène*)

Une fonction $f : \mathbb{R}^d \mapsto \overline{\mathbb{R}}$ est une fonction (positivement) homogène (de degré 1) si, pour tout $x \in \mathbb{R}^d$, $x \neq 0$, pour tout $\lambda > 0$, $f(\lambda x) = \lambda f(x)$.

Théorème 4.3 (*Fonctions $\mathbb{P}$-convexes dans la configuration linéaire classique*)

Si $\mathbb{P}$ est la configuration linéaire classique alors $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ est l'union de la fonction identiquement égale à $-\infty$ et de l'ensemble des fonctions f convexes semi-continues inférieurement et homogènes telles que $f(0) = 0$.

Lemme 4.2

Soit $\mathbb{P}$ la configuration linéaire alors

1. $v^{\mathbf{s}} \neq \emptyset \implies v(0) \geq 0$.
2. $v^{\mathbf{s}} = \mathbb{R}^d \iff \forall\, p \neq 0,\ v(p) = +\infty$ et $v(0) \geq 0$.

Démonstration

1. Si $v(0) < 0$ alors pour tout $x \in v^{\mathbf{s}}$, $0 \cdot x = 0 < 0$ ce qui est impossible et donc $v^{\mathbf{s}} = \emptyset$.
2. Soit $v \in \overline{\mathbb{R}}^{\mathbb{P}}$, si pour tout $p \neq 0$, $v(p) = +\infty$ et $v(0) \geq 0$, on a clairement $v^{\mathbf{s}} = \mathbb{R}^d$. D'après le point 1, $v^{\mathbf{s}} = \mathbb{R}^d$ implique que $v(0) \geq 0$. Maintenant, supposons que $v^{\mathbf{s}} = \mathbb{R}^d$, qu'il existe $p \neq 0$ tel que $v(p) < +\infty$. Comme $\mathbb{P}$ est isomorphe à $\mathbb{R}^d$ alors en posant $x = (v(p) + 1)(p/\|p\|^2) \in \mathbb{R}^d$, on a $p \cdot x = v(p) + 1 > v(p)$ et donc il existe $x \in \mathbb{R}^d$ tel que $x \notin v^{\mathbf{s}}$. ∎

Lemme 4.3 ($\mathbb{P}$-*enveloppe convexe et biconjuguée de Fenchel*)

Soit $\mathbb{P}$ la configuration linéaire. Pour tout $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ homogène, on a, pour tout $p \in \mathbb{P}$:

$$\mathrm{vex}_{\mathbb{P}}(v)(p) \leq v^{**}(p) \leq (\mathrm{vex}_{\mathbb{P}}(v))(p) + v(0)$$

Démonstration (lemme 4.3)

Par définition de la biconjuguée de Fenchel-Moreau, on a, quelque soit $p \in \mathbb{R}^d$,

$$v^{**}(p) = \sup_{x \in \mathbb{R}^d} p \cdot x + \inf_{q \in \mathbb{R}^d} (v(q) - q \cdot x) \ .$$

Si $v^{\mathbf{s}} = \mathbb{R}^d$, alors, par le lemme 4.2, $v(p) = +\infty$ pour tout $p \neq 0$ et $v(0) \geq 0$, d'où, pour tout $p \in \mathbb{P}$, $v^{**}(p) = \sup_{x \in \mathbb{R}^d} p \cdot x + v(0)$. Comme $v^{\mathbf{s}} = \mathbb{R}^d$ alors $(\mathrm{vex}_{\mathbb{P}}(v))(p) = \sup_{x \in \mathbb{R}^d} p \cdot x$ et $v(0) \geq 0$ d'où la double inégalité.

Supposons maintenant que $v^{\mathbf{s}} \subsetneq \mathbb{R}^d$ et $x \notin v^{\mathbf{s}}$, il existe $q \in \mathbb{R}^d$ tel que $v(q) - q(x) < 0$ et pour tout $\lambda > 0$, puisque v est homogène $\lambda(v(q) - q(x)) < 0$ d'où $\inf_{q \in \mathbb{R}^d} v(q) - q \cdot x = -\infty$, on en déduit que :

$$v^{**}(p) = \sup_{x \in v^{\mathbf{s}}} p \cdot x + \inf_{q \in \mathbb{R}^d} (v(q) - q \cdot x) \ . \tag{4.7}$$

Si $v^{\mathbf{s}}$ est vide alors $\mathrm{vex}_{\mathbb{P}}(v) \equiv -\infty$ et $v^{**} \equiv -\infty$ donc la double inégalité est vérifiée.

Nous supposons maintenant que $\emptyset \neq v^{\mathbf{s}} \subsetneq \mathbb{R}^d$. En utilisant (4.7), et en prenant $q = 0$, on obtient, pour tout $p \in \mathbb{P}$, $v^{**}(p) \leq \sup_{x \in v^{\mathbf{s}}} p \cdot x + v(0) = (\mathrm{vex}_{\mathbb{P}}(v))(p) + v(0)$.

De plus, si $x \in v^{\mathbf{s}}$, $\inf_{q \in \mathbb{R}^d} v(p) - q \cdot x \geq 0$ et finalement $v^{**}(p) \geq \sup_{x \in v^{\mathbf{s}}} p \cdot x = (\mathrm{vex}_{\mathbb{P}}(v))(p)$. ∎

Démonstration (théorème 4.3)
Soit $v \in \mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$, l'élément v est homogène puisqu'il est égal à $\mathrm{vex}_{\mathbb{P}}(v)$. En effet, pour tout $p \in \mathbb{P}$ pour tout $\lambda > 0$, $v(\lambda p) = \sup_{x \in v^{\mathbf{s}}} \lambda p(x) = \lambda \sup_{x \in v^{\mathbf{s}}} p(x) = \lambda v(p)$.

De plus, on a soit $v^{\mathbf{s}} = \emptyset$, soit $v^{\mathbf{s}} \neq \emptyset$. D'après la proposition 4.7, $v^{\mathbf{s}} = \emptyset \iff v \equiv -\infty$ et $v^{\mathbf{s}} \neq \emptyset \iff v(p) > -\infty \; \forall p \in \mathbb{P}$. Dans le premier cas, il n'y a plus rien à montrer. Supposons donc que $v(p) > -\infty$ pour tout $p \in \mathbb{P}$ et donc $v(0) = \sup_{x \in v^{\mathbf{s}}} 0 = 0$. D'après le lemme 4.3, on conclut que $v = \mathrm{vex}_{\mathbb{P}}(v) = v^{**}$ et donc v est convexe s.c.i d'après le théorème 4.1.

Maintenant, si $v \equiv -\infty$, alors $v \in \mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$. Supposons, par conséquent, que v soit convexe s.c.i, homogène et $v(0) = 0$, alors pour tout $p \in \mathbb{P}$, $v(p) > -\infty$. En effet, s'il existe $p \in \mathbb{P}$ tel que $v(p) = -\infty$ alors en prenant une suite $(\lambda_n)_n$ tendant vers 0^+, on a, par homogénéité, $v(\lambda_n p) = \lambda_n v(p) = -\infty$ et comme v est s.c.i alors $v(0) = -\infty$ ce qui contredit $v(0) = 0$. On conclut par le théorème 4.1 que $v = v^{**}$ puis par le lemme 4.3 que $v^{**} = \mathrm{vex}_{\mathbb{P}}(v)$ et finalement, $v = \mathrm{vex}_{\mathbb{P}}(v)$. ∎

4.4 Domaines des sous-niveaux et interprétation abstraite

4.4.1 Généralisation de domaines numériques abstraits de l'interprétation abstraite

Nous rappelons l'idée de Sankaranarayanan, Sipma et Manna [SSM05] : fixer un nombre fini de formes linéaires pour calculer des invariants numériques sous la forme de polyèdres. Le fait de fixer les faces du polyèdre répond aux problèmes d'explosion combinatoire du domaine des polyèdres. Dans notre nouveau domaine numérique dit des sous-niveaux, nous acceptons des fonctions aussi bien des formes linéaires que *non-linéaires* comme fonctions de base. De plus, nous ne nous limitons pas au caractère fini de la configuration. Ceci ouvre la voie aux calculs d'invariants convexes généraux en considérant toutes les formes linéaires.

Puisque notre domaine généralise celui de Sankaranarayanan, Sipma et Manna, notre domaine contient le domaine des intervalles en prenant comme configuration $\mathbb{P} = \{\pm e_i,\ i = 1 \ldots, d\}$ où $\{e_i\}_{i=1\ldots,d}$ représente la base duale de $\mathbb{R}^d$. Les $\mathbb{P}$-fonctions support, dans cette configuration, correspondent exactement aux bornes des intervalles. Nous pouvons recoder les zones de Miné [Min04] en tant que $\mathbb{P}$-sous niveau avec comme configuration $\mathbb{P}$ l'ensemble des formes linéaires des différences entre les variables c'est-à-dire $\{x_i - x_j,\ i, j = 0, \ldots, d,\ x_0 = 0\}$. Les octogones sont aussi des $\mathbb{P}$-sous niveaux dans la configuration $\mathbb{P} = \{\pm x_i \pm x_j,\ i, j = 0, \ldots, d,\ x_0 = 0\}$, les formes linéaires de $\mathbb{P}$ représentant bien les différences et sommes de variables. Dans les zones, comme dans les octogones, les $\mathbb{P}$-fonction support sont les *difference bounds matrices* (DBM).

4.4.2 Quelques configurations utiles en interprétation abstraite

Nous allons conclure ce chapitre par donner quelques exemples de configurations utiles pour l'analyse statique de programmes par interprétation abstraite.

Nous appelons *configuration affine tropicale*, la configuration contenant toutes les différences de fonctions tropicalement (au sens max-plus) affines c'est-à-dire de la forme :

$$p(x) = \max(b, \max_{j=1,\dots,d}(a_j + x_j)) - \max(e, \max_{j=1,\dots,d}(c_j + x_j))$$

où $b, e \in \mathbb{R} \cup \{-\infty\}$ et pour tout $j \in \{1, \dots, d\}$, $a_j, c_j \in \mathbb{R} \cup \{-\infty\}$. La fonction $p(x) = -\infty$ est tropicalement affine mais n'est pas utile comme fonctions de base. Cette configuration a été suggérée par Gaubert, Katz et Sergeev dans [GKS10]. Les polyèdres max-plus correspondent aux $\mathbb{P}$-polyèdres de la configuration affine tropicale. Dans, [AGG08b], les polyèdres max-plus sont utilisés pour générer des invariants pour des programmes manipulant des tableaux et des chaînes de caractères.

Nous appelons *configuration quadratique*, une configuration dont toutes les fonctions de base sont quadratiques c'est-à-dire des fonctions de $\mathbb{R}^d$ dans $\mathbb{R}$, $x :\mapsto x^T Ax + b^T x + c$. Si toutes les fonctions quadratiques sont convexes, nous parlons de *configurations quadratiques convexes*. Nous détaillerons le type d'invariants générés dans le chapitre 6 dans le cas d'une configuration quadratique convexe *finie*.

Enfin, nous terminons par une idée (future) : une *configuration géométrique* c'est-à-dire considérer des fonctions de base de la forme :

$$\sum_{k=1}^{K} c_k x_1^{a_{k1}} x_2^{a_{k2}} \dots x_d^{a_{kd}}$$

avec $c_k > 0$, pour tout $k = 1 \dots, K$ et $a_{kj} \in \mathbb{R}$, pour tout $k = 1, \dots, K$ et $j = 1, \dots, d$. Une section entière est consacrée à la programmation géométrique dans [BV04]. Une configuration géométrique peut s'avérer utile pour trouver des invariants lorsque l'arithmétique d'un programme est polynomiale.

Le tableau de la figure 4.17 récapitule quelques ensembles de fonctions de base utiles en interprétation abstraite.

Configurations	Classes d'ensembles	Algorithmes	Utilisation
infinie	Convexes fermés (et autres ?)	Programmation semi-infinie	Itération sur les politiques dynamique chapitre 5
linéaire	polyèdres ; convexes fermés	programmation linéaire	invariants numériques
affine tropicale	convexes tropicaux fermés	programmation linéaire tropicale ?	[AGG08b] [All09, Chapter 7]
quadratique convexe finie	CQ-représentables [TN01, Section 2.3.1]	programmation quadratique	chapitre 6
géométrique		programmation géométrique	arithmétique polynomiale

Figure 4.17 – Quelques ensembles de fonctions de base utiles en interprétation abstraite

CHAPITRE 5

ITÉRATION SUR LES POLITIQUES DYNAMIQUE DANS LE DOMAINE DES SOUS-NIVEAUX

Nous avons présenté dans le chapitre 4 un nouveau domaine numérique abstrait. Nous développons dans ce chapitre une méthode pour calculer des invariants numériques dans ce domaine : afin de borner les variables d'un programme. En se fixant une configuration $\mathbb{P}$, nous construisons, dans un premier temps, une fonction sémantique abstraite $F^\sharp = ({F_1}^\sharp, \ldots, {F_n}^\sharp)$ à partir la correspondance de Galois de la proposition 4.6 ce qui est classique en interprétation abstraite (voir 2.17). La fonction $F^\sharp$ étant croissante sur le treillis complet $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})^n$, pour trouver des bornes précises sur les variables du programme analysé, nous sommes amenés à résoudre le problème :

$$\inf\{v \in \mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})^n \mid F^\sharp(v) \leq v\} \ .$$

Nous cherchons a priori des fonctions $\mathbb{P}$-convexes ce qui nous permet de déduire des ensembles $\mathbb{P}$-convexes. Cependant, l'évaluation de la fonction $F^\sharp$ est équivalente à chercher la valeur optimale d'un problème d'optimisation globale non nécessairement convexe. La construction de la fonction sémantique abstraite nous amène donc à considérer une deuxième fonction sémantique appelée fonction sémantique *relâchée* qui fournit une sur-approximation de la fonction sémantique abstraite classique. Cette fonction sémantique est construite à partir de la théorie de la dualité lagrangienne dans le cadre de l'optimisation semi-infinie. Cette approche nous permet également de développer un cadre théorique pour définir une itération sur les politiques dynamique : les contraintes effectives des problèmes d'optimisation déterminent quelles fonctions de base peuvent être sélectionnées. Par ailleurs, la représentation duale est utilisée pour définir nos politiques. D'un point de vue effectif, l'application porte sur les configurations quadratiques finies qui sera développée au chapitre 6.

Nous présentons dans la section 5.1, une légère modification de la fonction sémantique concrète définie dans le chapitre 2 et nous discutons la construction de la fonction sémantique abstraite dans le domaine abstraits des sous-niveaux. Dans la section 5.2, nous développons la construction de la fonction sémantique relâchée. Dans la section 5.3, nous comparons les deux fonctions sémantiques et exposons les propriétés de la fonction sémantique relâchée. Ensuite, dans la section 5.4, nous discutons des propriétés de la fonction sémantique relâchée. Enfin, à la section 5.5, nous définissons une itération sur les politiques où l'ensemble des politiques est choisi dynamiquement. Nous illustrons ce schéma d'itération sur un exemple simple.

5.1 Fonction sémantique abstraite

Dans le chapitre 2, nous avons défini une sémantique concrète puis une sémantique abstraite définie à partir d'une correspondance de Galois. Dans cette section, nous allons construire une sémantique abstraite dans le domaine numérique des sous-niveaux à partir de la correspondance de Galois donnée par la proposition 4.6.

Nous avons vu au chapitre 2 que chaque coordonnée de la fonction sémantique concrète F^C était essentiellement composée d'opérations constantes, d'union, d'intersections et d'images de sous-ensemble de $\mathbb{R}^d$ par une fonction. Par conséquent, pour construire une fonction sémantique abstraite sur le domaine des sous-niveaux, il suffit de décrire, à partir de la correspondance de Galois de la proposition 4.6, comment se traduisent les unions, intersections et images de sous-ensemble de $\mathbb{R}^d$ par une fonction dans le treillis complets $\overline{\mathbb{R}}^{\mathbb{P}}$ des fonctions de $\mathbb{P}$ dans $\overline{\mathbb{R}}$.

Nous notons toujours n le nombre de points de contrôle. On cherche donc à construire une fonction $F^\sharp$ de $(\overline{\mathbb{R}}^{\mathbb{P}})^n$ dans $(\overline{\mathbb{R}}^{\mathbb{P}})^n$. Dans toute la suite de ce chapitre, l'élément X représente le vecteur d'ensembles $(X_1, \ldots, X_n)$. L'élément v désigne un vecteur d'éléments de $\overline{\mathbb{R}}^{\mathbb{P}}$ $(v_1, \ldots, v_n)$.

5.1.1 Quelques modifications sur la fonction sémantique concrète

Dans ce chapitre, nous utilisons la syntaxe définie 2.2. Pour analyser plus facilement les programmes dans le domaine des sous-niveaux, nous opérons quelques modifications dans la fonction sémantique concrète.

Premièrement, quitte à ajouter des coordonnées dans la fonction sémantiques concrète (ou de manière équivalente ajouter des instructions *skip*), nous supposons, dans ce chapitre, que les opérations d'unions décrites dans le reste de la thèse n'ont que deux opérandes.

Deuxièmement, nous avons vu que les opérations d'intersections venaient des tests et donc des boucles **while**, et des branchements conditionnels **if**, **then** et **else**. Dans le chapitre 2, nous décomposions les tests et les affectations internes des boucles et des branchements conditionnels, afin de permettre une meilleure lisibilité des interprétations des programmes et aussi d'assurer qu'une coordonnée de la fonction sémantique abstraite correspondait à une seule opération d'union, de test ou d'affectation.

L'affectation interne d'une boucle n'est exécutée que si le vecteur de variable satisfait le test d'entrée d'une boucle, c'est-à-dire si le vecteur de variable appartient à l'ensemble $\{y \in \mathbb{R}^d \mid r(y) \leq 0\}$ où r est une fonction (continue) de $\mathbb{R}^d$ dans $\mathbb{R}$. Par conséquent, la fonction sémantique concrète pour l'affectation $x = T(x)$ interne d'une boucle s'écrit $T(X_i \cap \{y \in \mathbb{R}^d \mid r(y) \leq 0\})$ où i est le point de contrôle correspondant à l'union des ensembles des valeurs prises par les variables au premier parcours de la boucle et après chaque tour de boucle.

De même, pour un branchement conditionnel **if**, **then**, **else**, les affectations des branches **then** et **else** ne sont exécutées que si les tests d'entrée sont satisfaits. Nous supposons, quitte à rajouter des coordonnées à la fonction sémantique concrète, qu'il n'y a qu'un seul arc (ij) entrant au point de contrôle j correspondant au branchement conditionnel. En appelant provisoirement $T_{\textbf{then}}$ et $T_{\textbf{else}}$ les affectations respectives des branches **then** et **else**, la fonction sémantique concrète pour ces affectations s'écrit respectivement $T_{\textbf{then}}(X_i \cap \{y \in \mathbb{R}^d \mid r(y) \leq 0\})$ et $T_{\textbf{else}}(X_i \cap \{y \in \mathbb{R}^d \mid -r(y) < 0\})$. Cependant, par la suite nous évaluerons la fonction sémantique concrète en résolvant des problèmes de maximisation. Dans des cas dégénérés (r concave et le premier ensemble non vide), maximiser une fonction continue sur $\{y \in \mathbb{R}^d \mid$

$-r(y) < 0\}$ est équivalent à optimiser sur $\{y \in \mathbb{R}^d \mid -r(y) \leq 0\}$. Dans les cas généraux, le deuxième ensemble est toujours plus grand que le premier, par conséquent maximiser sur le second fournit une sur-approximation, ce qui nous suffit en interprétation abstraite. Nous ne donc considérons que $T_{\textbf{else}}(X_i \cap \{y \in \mathbb{R}^d \mid -r(y) \leq 0\})$. Les fonctions sémantiques concrètes sur les branches **then** et **else** ont par conséquent la même structure et nous n'étudierons que la fonction sémantique concrète uniquement sur la branche **then**.

Remarque 5.1 Nous prenons des fonctions de test r à valeurs dans $\mathbb{R}$ puisque le complémentaire de l'ensemble $\{y \in \mathbb{R}^d \mid r(x) \leq 0\}$ est une union d'ensembles $\{y \in \mathbb{R}^d \mid r_k(x) > 0\}$ dans le cas où r est valeurs dans $\mathbb{R}^m$ avec $m > 1$.

Nous notons l'ensemble de toutes les coordonnées de la fonction sémantique concrète étant une union dans l'ensemble :

$$\mathbb{U} = \{u \in \{1, \ldots, n\} \mid \exists\, j, k \in \{1, \ldots, n\},\ F_u^C(X) = X_j \cup X_k\}\ .$$

A chaque élément $u \in \mathbb{U}$, on associe $\ell_d(u), \ell_g(u)$ qui sont respectivement les coordonnées des opérandes droite et gauche de l'union. De même, nous notons, l'ensemble de toutes les coordonnées de la fonction sémantique concrète représentant une affectation :

$$\mathbb{A} = \{a \in \{1, \ldots, n\} \mid \exists\, l \in \{1, \ldots, n\}, \exists\, T \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d),\ F_a^C(X) = T(X_l)\}\ .$$

A chaque élément $a \in \mathbb{A}$, on associe $\ell(a)$ qui est la coordonnée sur laquelle agit l'affectation. Enfin, nous notons l'ensemble de toutes les coordonnées de la fonction sémantique concrète étant une opération d'intersection :

$$\mathbb{I} = \{\imath \in \{1, \ldots, n\} \mid \exists\, l \in \{1, \ldots, n\}, \exists\, T, r \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d),\ F_i^C(X) = T(X_l \cap \{y \in \mathbb{R}^d \mid r(y) \leq 0\}\ .$$

A chaque élément $\imath \in \mathbb{I}$, on associe $\ell(\imath)$ qui est la coordonnée sur laquelle agit l'affectation interne de la boucle ou du branchement conditionnel.

Dans tout ce chapitre, on fixe une configuration $\mathbb{P}$ pour analyser un programme. Dans toute la suite du chapitre, on note une affectation T et un test r.

Pour illustrer nos définitions, nous prendrons tout au long de ce chapitre le programme donné par la figure 5.1.

5.1.2 Les constantes

Dans ce chapitre, nous supposons que les constantes sont des sous-ensembles D de $\mathbb{R}^d$. De plus, nous supposons que les ensembles D sont $\mathbb{P}$-convexes où $\mathbb{P}$ est la configuration utilisée pour analyser le programme.

Pour tout $i \in \{1, \ldots, n\}$ représentant une constante, la fonction sémantique concrète est donc de la forme $F_i^C(X) = D$. On obtient ainsi l'égalité suivante pour la fonction sémantique abstraite, pour $v \in \overline{\mathbb{R}}^{\mathbb{P}}$:

$$F_i^\sharp(v) = w$$

où w est une fonction indépendante de v et qui vérifie $w^{\mathbf{s}} = D$.

```
u=0;
assume (x,y) in B(0,1);[1]
u=x;
x=y;
y=u;[2]
while [3](x*x+y*y<=1){
      u=x;
      x=2*y+3*x*x;
      y=0.5*y+0.2*exp(u);[4]
      if (x>=0)
            then{
               u=x;
               x=y+1;
               y=u+1;[5]
               }
            else{
               u=x;
               x=y+1;
               y=u*u;[6]
               };
          [7]
   }[8]
```

Figure 5.1 – Programme avec une boucle et branchement conditionnel

Exemple 5.1 (Produit d'intervalles)
Supposons qu'un programme contient une initialisation :

$$assume\ x \in C$$

où l'ensemble C est un produit cartésien d'intervalles c'est-à-dire de la forme :

$$\prod_{i=1}^{d} [a_i, b_i]$$

où, pour tout $i \in \{1, \ldots, d\}$, $a_i \in \mathbb{R}$, $b_i \in \mathbb{R}$ et $a_i \leq b_i$. Considérons maintenant la configuration $\mathbb{P}$ suivante :

$$\{\underline{x_i} : x \mapsto x_i,\ -\underline{x_i},\ p : x \mapsto g(x)\}$$

où p est une fonction continue sur $\mathbb{R}^d$. On suppose $\sup_{x \in C} g(x)$ est connu et donc remplace l'ensemble initial C par l'ensemble D de la forme $w^{\mathbf{s}}$ avec :

$$w(q) = b_i \text{ si } q = \underline{x_i},\ w(p) = -a_i \text{ si } q = -\underline{x_i},\ w(q) = \sup_{x \in C} g(x) \text{ si } q = g\ .$$

Exemple 5.2 (Boule et sphère unité de $\mathbb{R}^d$)
On munit $\mathbb{R}^d$ de la norme euclidienne et considérons $B(0,1)$ la boule unité fermée de $\mathbb{R}^d$ et $S(0,1)$ la sphère unité de $\mathbb{R}^d$. Soient les configurations $\mathbb{P}_1 = \{x \mapsto \|x\|^2,\ x \mapsto -\|x\|^2\}$ et $\mathbb{P}_2 = \{x \mapsto p \cdot x,\ p \in B(0,1)\}$. Supposons qu'un programme contient une initialisation :

$$assume\ x \in C$$

où $C = B(0,1)$ ou $C = S(0,1)$. On peut réécrire $B(0,1)$ comme le $\mathbb{P}_1$-sous niveau $u^{\mathbf{s}}$ avec :

$$u(p) = 1 \text{ si } p = x \mapsto \|x\|^2, \text{ et } u(p) = 0 \text{ si } p = x \mapsto -\|x\|^2\ .$$

On peut également représenter $B(0,1)$ comme le $\mathbb{P}_2$-sous niveau $v^{\mathbf{s}}$ avec :

$$v(p) = \|p\| \text{ pour tout } p \in \mathbb{P}_2\ .$$

La fonction v est $\mathbb{P}$-convexe et continue (en fait convexe semi-continue et finie).

Dans le cas où $C = S(0,1)$, on peut représenter C par le $\mathbb{P}_1$-sous niveau $w^{\mathbf{s}}$ avec :

$$w(x \mapsto \|x\|^2) = 1 \text{ et } w(x \mapsto -\|x\|^2) = -1\ .$$

Appliquons ce résultat au programme de la figure 5.1, au point de contrôle [1], la fonction sémantique abstraite est la boule unité fermée $B_2(0,1)$ et donc le $\mathbb{P}_2$-sous niveau $v_1^{\mathbf{s}}$ est égal à $B_2(0,1)$ si $v_1(p) = \|p\|$ pour tout $p \in \mathbb{P}_2$.

5.1.3 Les unions

Nous rappelons que les opérateurs d'unions n'ont ici que deux opérandes. Ces deux opérandes sont des sous-ensembles de $\mathbb{R}^d$ qui représentent l'ensemble des valeurs possibles à deux points de contrôle distincts. Soit $u \in \mathbb{U}$, la fonction sémantique concrète a pour coordonnée F_u^C définie par l'équation (5.1) :

$$F_u^C(X) = X_{\ell_g(u)} \cup X_{\ell_d(u)}\ . \tag{5.1}$$

Par définition de la correspondance de Galois de la proposition 4.6, la fonction sémantique abstraite pour une union vérifie l'équation (5.2) :

$$F_u^\sharp(v) = (v^{\mathbf{s}}_{\ell_g(u)} \cup v^{\mathbf{s}}_{\ell_d(u)})^\sigma\ . \tag{5.2}$$

Remarque 5.2 Les propriétés des correspondances de Galois 4.5 impliquent que $(v^{\mathbf{s}}_{\ell_g(u)} \sqcup v^{\mathbf{s}}_{\ell_d(u)})^\sigma = (v^{\mathbf{s}}_{\ell_g(u)} \cup v^{\mathbf{s}}_{\ell_d(u)})^\sigma$.

L'évaluation d'une coordonnée de la fonction sémantique abstraite représentant une union pour un élément $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ et pour une fonction de base $q \in \mathbb{P}$ est équivalente au problème 5.1.

Problème 5.1 (Evaluation de l'union)
Résoudre :

$$\text{Max} \quad q(x) \quad \text{s. c} \quad x \in v^{\mathbf{s}}_{\ell_g(u)} \cup v^{\mathbf{s}}_{\ell_d(u)}$$

Exemple 5.3 (Boucle dans le programme de la figure 5.1)
Sur le programme de la figure 5.1, au point de contrôle [3], la boucle **while** entraîne une opération d'union : la fonction sémantique concrète au point de contrôle [3] est :

$$X_3 = X_2 \cup X_7$$

Pour rappel, nous trouvons ces deux coordonnées en remarquant que les moyens d'entrée dans la boucle **while** sont soit, si la boucle **while** n'a jamais été parcourue, de venir du point de contrôle [2], soit, si la boucle **while** a déjà été parcourue, de revenir après l'opération interne de la boucle c'est-à-dire le point de contrôle [7].

Soit une configuration $\mathbb{P}$. Soit $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^8$, les ensembles $v_i^{\mathbf{s}}$ s'entendent donc par rapport à cette configuration. La fonction sémantique abstraite au point de contrôle [3] est donc la fonction qui à $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^8$ associe :

$$F_3^\sharp(v) = (v_2^{\mathbf{s}} \cup v_7^{\mathbf{s}})^\sigma \ .$$

5.1.4 Les affectations

On rappelle qu'une affectation est la donnée d'une application continue T de $\mathbb{R}^d$ dans $\mathbb{R}^d$. On suppose, de plus, que l'application T agit sur unique point de contrôle. Soit $a \in \mathbb{A}$, la fonction sémantique concrète d'une affectation a pour coordonnée F_a^C et est caractérisée par l'équation (5.3) :

$$F_a^C(X) = T(X_{\ell(a)}) \ . \tag{5.3}$$

La composante d'une fonction sémantique abstraite pour une affectation s'écrit :

$$F_a^\sharp(v) = (T(v_{\ell(a)}^{\mathbf{s}}))^\sigma = \sup_{x \in v_{\ell(a)}^{\mathbf{s}}} \mathbf{e}_{T(x)} \ . \tag{PAF}$$

L'évaluation d'une composante de la fonction sémantique abstraite représentant une affectation pour un élément $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ et une fonction de base $q \in \mathbb{P}$ est équivalente à la résolution problème 5.2.

Problème 5.2 (Evaluation d'une affectation)
Résoudre :

$$\text{Max} \quad q(y) \quad \text{s. c} \quad \begin{array}{l} y = T(x) \\ x \in v_{\ell(a)}^{\mathbf{s}} \end{array} \tag{PAF_q}$$

Ainsi, on remarque que la fonction sémantique abstraite d'une affectation est la fonction valeur Val(PAF) qui, à toute fonction de base $q \in \mathbb{P}$, associe la valeur du problème d'optimisation (PAF_q), en d'autres termes, Val(PAF) : $q \mapsto$ Val(PAF_q). Si $v_{\ell(a)}$ prend la valeur $-\infty$ le problème n'est pas réalisable et la fonction sémantique abstraite vaut $-\infty$ partout. Par ailleurs, nous rappelons que $v_{\ell(a)}^{\mathbf{s}}$ est uniquement déterminé par les valeurs de $v_{\ell(a)}$ sur son domaine et donc $x \in v_{\ell(a)}^{\mathbf{s}}$ signifie $p(x) \leq v_{\ell(a)}(p)$ pour tout $p \in \text{dom}(v_{\ell(a)})$ si celui-ci est non vide. Si le domaine de $v_{\ell(a)}$ est vide alors les problèmes PAF_q sont non contraints et la difficulté de résolution du problème dépend uniquement de la configuration $\mathbb{P}$ choisie.

Exemple 5.4 (Affectation au point de contrôle [2] du programme de la figure 5.1) Considérons la configuration suivante $\mathbb{P} = \{(x,y) :\mapsto p_1x + p_2y,\ (p_1,p_2) \in B_2(0,1)\}$ où $B_2(0,1)$ dénote la boule unité fermée de $\mathbb{R}^2$.

L'affectation au point de contrôle [2] agit sur le point de contrôle [1], la fonction sémantique abstraite est donc, pour $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^8$:

$$[F_2^\sharp(v)](q) = \sup\{q_1y + q_2x \mid p \cdot (x,y) \leq v_1(p),\ \forall p \in B_2(0,1)\}\ .$$

Par l'inégalité de Cauchy-Schwarz, comme $v_1^{\mathbf{s}} = B_2(0,1)$, $\sup\{q \cdot (y,x) \mid (x,y) \in v_1^{\mathbf{s}}\} = \|q\|$.

Enfin, on remarque que le problème 5.2 généralise le calcul de la $\mathbb{P}$-enveloppe convexe de fonctions. En effet, le calcul de la $\mathbb{P}$-enveloppe convexe d'un élément $w \in \overline{\mathbb{R}}^{\mathbb{P}}$ pour une fonction de base $p \in \mathbb{P}$ est le problème 5.2 où l'application T est l'identité.

Problème 5.3 (Problème de clôture)

Résoudre :

$$\text{Max} \quad p(x) \quad \text{s. c} \quad x \in w^{\mathbf{s}}$$

Exemple 5.5 (Intervalles)
Prenons $\mathbb{P} = \{(x,y) :\mapsto p_1x + p_2y,\ (p_1,p_2) \in B_2(0,1)\}$. Soit v la fonction $p \in \mathbb{P}$ définie par $v(p) = 1$ si $p \in \{(1,0),(-1,0),(0,1),(0,-1)\}$ et $+\infty$ sinon. Alors $v^{\mathbf{s}}$ est le produit d'intervalles $[-1,1]^2$, la fonction $\text{vex}_{\mathbb{P}}(v)$ est donc la fonction qui à $p = (p_1,p_2)$ associe $\sup\{-p_1-p_2,\ -p_1+p_2,\ p_1-p_2,\ p_1+p_2\}$ puisque $(x,y) \mapsto p_1x+p_2y$ définit une forme linéaire et donc le maximum de ces fonctions sur $v^{\mathbf{s}}$ sont atteints sur les points extrêmes de $v^{\mathbf{s}}$ qui sont les vecteurs $(-1,-1)$, $(-1,1)$, $(1,-1)$ et $(1,1)$. La valeur de $\sup\{-p_1 - p_2,\ -p_1 + p_2,\ p_1 - p_2,\ p_1 + p_2\}$ dépend uniquement des signes de p_1 et p_2.

L'opération de clôture devient plus pathologique si l'on prend une classe trop générale de fonctions $p \in \mathbb{P}$.

Exemple 5.6 (Clôture pathologique)
Prenons, $\mathbb{P} = \{p \in \mathcal{C}(\mathbb{R},\mathbb{R}) \mid p(x) = 0,\ \forall x \notin [0,1],\ p(1/2) \leq 0\}$ et on définit $v(p) = (p(1/2))^2$ pour tout $p \in \mathbb{P}$. Nous cherchons à évaluer la $\mathbb{P}$-enveloppe convexe $\text{vex}_{\mathbb{P}}(v)$. Nous commençons par calculer $v^{\mathbf{s}}$ et nous allons montrer que $v^{\mathbf{s}} = \{0, 1/2, 1\}$. Par continuité des fonctions $p \in \mathbb{P}$, on a $p(0) = p(1) = 0$ d'où $p(0) = p(1) \leq (p(1/2))^2$ pour tout $p \in \mathbb{P}$. De plus, comme $p(1/2) \leq 0$ pour tout $p \in \mathbb{P}$ alors $1/2 \in v^{\mathbf{s}}$. On conclut que $]-\infty,0] \cup \{1/2\} \cup [1,+\infty[\subseteq v^{\mathbf{s}}$. Maintenant, considérons la fonction h nulle en dehors de $[0,1]$ suivante dont le graphe est représenté par la figure 5.2 :

$$h(x) = \begin{cases} x & \text{si } x \in [0,1/4] \\ 1/2 - x & \text{si } x \in [1/4,1/2] \\ x - 1/2 & \text{si } x \in [1/2,3/4] \\ 1 - x & \text{si } x \in [3/4,1] \end{cases}$$

La fonction h appartient à $\mathbb{P}$ et vérifie $h(0) = h(1/2) = h(1) = 0$ d'où $h(x) \leq (h(1/2))^2$ implique que $x \in]-\infty,0] \cup \{1/2\} \cup [1,+\infty[$ d'où $v^{\mathbf{s}} \subseteq]-\infty,0] \cup \{1/2\} \cup [1,+\infty[$. Finalement, comme

$$[\text{vex}_{\mathbb{P}}(v)](p) = \sup\{p(x) \mid x \in v^{\mathbf{s}}\},$$

on conclut que $\text{vex}_{\mathbb{P}}(v)$ est la fonction identiquement nulle.

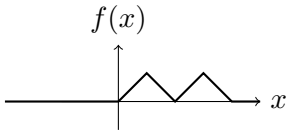


Figure 5.2 – Graphe de la fonction h de l'exemple 5.6

5.1.5 Les intersections

Dans la sous-section 5.1.1, nous avons décrit la construction de la fonction sémantique concrète pour les intersections. Dans le cas des intersections construites à partir une boucle **while**, la coordonnée sur laquelle agit l'affectation interne de la boucle correspond à l'union engendrée par la boucle. Quand l'intersection est construite à partir d'un branchement conditionnel, la coordonnée sur laquelle agit l'opération interne des branches désigne le noeud initial de l'arc entrant au point de contrôle associé au branchement conditionnel. Soit $\imath \in \mathbb{I}$, la fonction sémantique concrète a pour coordonnée $F_\imath^C$ définie par l'équation (5.4) :

$$F_\imath^C(X) = T(X_{\ell(\imath)} \cap \{y \in | \; r(y) \leq 0\}) \;\; . \tag{5.4}$$

On en déduit que la fonction sémantique abstraite pour une intersection vérifie l'équation (PIT) :

$$F_\imath^\sharp(v) = T(v^{\mathbf{s}}_{\ell(\imath)} \cap \{y \in | \; r(y) \leq 0\})^\sigma = \sup_{\substack{x \in v^{\mathbf{s}}_{\ell(\imath)} \\ r(x) \leq 0}} \mathbf{e}_{T(x)} \;\; . \tag{PIT}$$

En conclusion, l'évaluation des coordonnées de la fonction sémantique abstraite représentant une boucle ou un branchement conditionnel, pour un élément $v \in \overline{\mathbb{R}}^{\mathbb{P}}$, pour une fonction de base $q \in \mathbb{P}$, est équivalente à la résolution du problème 5.4.

Problème 5.4 (Evaluation d'une intersection)
Résoudre :

$$\begin{array}{lll} \text{Max} \quad q(y) \quad \text{s. c} & y = T(x) \\ & x \in v^{\mathbf{s}}_{\ell(\imath)} \\ & r(x) \leq 0 \end{array} \tag{PIT_q}$$

Ainsi, on remarque que la fonction sémantique abstraite d'une intersection est la fonction valeur Val(PIT) qui, à toute fonction de base $q \in \mathbb{P}$, associe la valeur du problème d'optimisation (PIT_q), en d'autres termes, Val(PIT) : $q \mapsto$ Val(PIT_q). Si $v_{\ell(\imath)}$ prend la valeur $-\infty$ le problème n'est pas réalisable et la fonction sémantique abstraite vaut $-\infty$ partout. Par ailleurs, nous rappelons que $v^{\mathbf{s}}_{\ell(\imath)}$ est uniquement déterminé par les valeurs de $v_{\ell(\imath)}$ sur son domaine et donc $x \in v^{\mathbf{s}}_{\ell(\imath)}$ signifie $p(x) \leq v_{\ell(\imath)}(p)$ pour tout $p \in \text{dom}(v_{\ell(\imath)})$ dès que $\text{dom}(v_{\ell(\imath)})$ est non vide. Si $\text{dom}(v_{\ell(\imath)})$ est vide, le problème devient un problème d'optimisation sous contrainte qui ne dépend pas de $v_{\ell(\imath)}$. Nous pouvons utiliser la dualité lagrangienne classique [Roc96, section 28] en limitant l'analyse des programmes au cas des tests convexes.

Exemple 5.7 (Affectation interne de la boucle au point de contrôle [3])
Nous reprenons le programme de la figure 5.1] et nous reprenons la configuration suivante $\mathbb{P} = \{(x, y) :\mapsto p_1 x + p_2 y, \;\; (p_1, p_2) \in B_2(0, 1)\}$ de l'exemple 5.4.

L'affectation au point de contrôle [3] agit sur le point de contrôle [2] uniquement si le test d'entrée de boucle est satisfait. Appelons r la fonction définissant le test d'entrée de la boucle : r est la fonction de $\mathbb{R}^2$ dans $\mathbb{R}$ qui à (x, y) associe $r(x, y) = x^2 + y^2 - 1$. La fonction sémantique abstraite est donc, pour $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^8$:

$$[F_3^\sharp(v)](q) = \\ \sup\{q_1(2y + 3x^2) + q_2(0.5y + 0.2\exp(x)) \mid r(x, y) \leq 0,\ p \cdot (x, y) \leq v_2(p),\ \forall p \in B_2(0, 1)\}$$

Exemple 5.8 (Branche conditionnelle then au point de contrôle [5])
Nous reprenons le programme de la figure 5.1] ainsi que la configuration suivante $\mathbb{P} = \{(x, y) :\mapsto p_1 x + p_2 y,\ (p_1, p_2) \in B_2(0, 1)\}$ de l'exemple 5.4.

L'affectation au point de contrôle [5] agit sur le point de contrôle [4] uniquement si le test d'entrée de branche est s'attifait. La fonction sémantique abstraite est donc, pour $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^8$:

$$[F_5^\sharp(v)](q) = \sup\{q_1(y + 1) + q_2(x + 1) \mid x \leq 0,\ p \cdot (x, y) \leq v_2(p),\ \forall p \in B_2(0, 1)\}\ .$$

5.2 Dualité et sémantique relâchée

Les problèmes 5.2, 5.4 sont de même nature. La difficulté des problèmes d'optimisation vient principalement du fait que $\mathbb{P}$ n'a aucune structure particulière et que $\mathbb{P}$ est à priori infini : la fonction objective est une fonction sur $\mathbb{R}^d$ mais le nombre de contrainte est infini, les problèmes 5.2 et 5.4 appartiennent à la classe de problèmes d'optimisation semi-infinie, pour plus de détails sur le sujet, le lecteur peut consulter [GL02, HK93, LS07c, GVRSS08].

En optimisation sous contraintes, une méthode de résolution consiste à résoudre un problème qui a, dans de bons cas, la même valeur que le problème initial mais qui possède toujours de bonnes structures (convexité et semi-continuité) et qui donne toujours une borne valide pour le problème initial. Ce nouveau problème appelé problème dual est construit à partir du problème primal. On "ajoute" les contraintes à la fonction objective par le biais d'une forme bilinéaire définie sur le produit cartésien de l'ensemble des valeurs des fonctions définissant les contraintes et d'un espace bien choisi. Cette méthode est appelé dualité lagrangienne [BS00, Section 2.5.3].

5.2.1 Rappels de dualité

Une méthode classique pour résoudre un problème d'optimisation sous contrainte est d'introduire des vecteurs du dual algébrique ou topologique quand ceux-ci sont identifiables. Le but est d'utiliser la bilinéarité de l'identification pour se ramener à un problème qui a de bonnes structures (convexité et continuité). Lorsque les duaux ne sont pas identifiables, on utilise une notion de dualité plus faible (voir [Sch71, Chapter 4]) : on restreint par cette construction le nombre de formes linéaires à considérer mais on conserve la bilinéarité.

Remarque 5.3 Tous les espaces vectoriels réels disposent d'une base de Hamel. Grâce aux bases de Hamel, on peut toujours identifier le dual algébrique d'un espace vectoriel aux fonctions sur cette base de Hamel dans $\mathbb{R}$ (voir [AB06, Section 5.9]). Cependant, une base de Hamel est souvent difficile à expliciter.

Crochet de dualité

Définition 5.1 (*Crochet de dualité*)

Les espaces vectoriels E et F sont en dualité s'il existe une forme bilinéaire $\langle \cdot, \cdot \rangle$ sur $E \times F$ à valeurs dans $\mathbb{R}$ telle que :

1. $\langle x, y \rangle = 0$, pour tout $y \in F$ implique que $x = 0$.
2. $\langle x, y \rangle = 0$, pour tout $x \in E$ implique que $y = 0$.

On appelle la forme bilinéaire $\langle \cdot, \cdot \rangle$, crochet de dualité entre E et F.

Si E et F sont en dualité, on dira que F est un dual de E ou que E est un dual de F. On notera le crochet de dualité associé $\langle \cdot, \cdot \rangle_{E \times F}$.

Remarque 5.4 Pour deux espaces vectoriels E et F, chaque élément $\langle \cdot, y \rangle$ avec $y \in F$ définit à une forme linéaire sur E et donc appartient au dual algébrique de E (éventuellement au dual topologique de E si elle est continue). Tous les espaces vectoriels en dualité avec E sont des sous ensembles du dual algébrique de E.

Remarque 5.5 Un espace de Banach et son dual topologique sont en dualité. En effet, on peut prendre comme crochet de dualité $\langle x, x' \rangle = x'(x)$ où x' désigne la forme linéaire continue et x le vecteur de l'espace de Banach. Par ailleurs, lorsque le dual topologique est identifiable, le crochet de dualité peut être donné par la forme bilinéaire d'identification.

Exemple 5.9 (Cas général)
Soit E un ensemble quelconque (sans structure particulière), on met en dualité $\mathbb{R}^E$ avec le sous-espace vectoriel des éléments presque nuls. On appelle éléments presque nuls, les vecteurs $\lambda \in \mathbb{R}^E$ dont le support (au sens fonctionnel), c'est-à-dire, l'ensemble $\text{supp}(\lambda) = \{x \in E \mid \lambda(x) \neq 0\}$ est fini ou vide. On note l'espace vectoriel des éléments presque nuls $\mathbb{R}^E_{fini}$.

Ainsi, de manière générale, on met en dualité $\mathbb{R}^E$ et $\mathbb{R}^E_{fini}$ par la forme bilinéaire :

$$(f, \lambda) \mapsto \langle f, \lambda \rangle = \sum_{x \in \text{supp}(\lambda)} f(x)\lambda(x) \ .$$

Exemple 5.10 (Dualité produit)
Soient E, F deux espaces vectoriels. Soient maintenant un espace dual E^* (resp. F^*) de E (resp. F) et notons $\langle \cdot, \cdot \rangle_{E \times E^*}$ (resp. $\langle \cdot, \cdot \rangle_{F \times F^*}$) le crochet de dualité associé. L'espace vectoriel produit $E \times F$ est en dualité avec $E^* \times F^*$ avec pour crochet de dualité $\langle \cdot, \cdot \rangle_{E \times E^*} + \langle \cdot, \cdot \rangle_{F \times F^*}$.

Exemple 5.11 (Cas fini)
Soit E un espace vectoriel de dimension finie d. Dans ce cas, on peut identifier le dual de E à E et le crochet de dualité est le produit scalaire de E. Le crochet de dualité est, par conséquent, défini par l'application de $E \times E$ dans $\mathbb{R}$:

$$(f, \lambda) \mapsto \langle f, \lambda \rangle = \sum_{i=1}^{d} f_i \lambda_i$$

Exemple 5.12 (Cas compact)
L'ensemble des mesures signées forme un outil intéressant pour la programmation semi-infinie. En effet, si nous supposons que E est un espace métrique compact, le théorème de représentation de Riesz-Markov permet d'identifier le dual topologique de $\mathcal{C}(E, \mathbb{R})$ aux mesures boréliennes signées finies [1] sur l'espace métrique compact E. L'ensemble des boréliennes signées finies noté $\mathcal{M}(E)$ est un espace de Banach lorsqu'il est muni de la norme de la variation totale. Le crochet de dualité est donné par l'intégrale de Lebesgue c'est-à-dire, l'application de $\mathcal{C}(E, \mathbb{R}) \times \mathcal{M}(E)$:

$$(f, \lambda) \mapsto \langle f, \lambda \rangle = \int_E f(x) d\lambda(x) \ .$$

Exemple 5.13 (Cas intégrable)
Supposons que $(E, \mathcal{T})$ est un espace mesurable et soit μ une mesure (positive) finie sur $\mathcal{T}$. Pour tout $p \in]1, +\infty[$, on peut munir $\mathrm{L}^p(E, \mathcal{T}, \mu)$ c'est-à-dire l'ensemble des fonctions mesurables telles que $\int_E |f(x)|^p d\mu(x) < +\infty$ de la dualité $\mathrm{L}^p - \mathrm{L}^q$: le dual de $\mathrm{L}^p(E, \mathcal{T}, \mu)$ est $\mathrm{L}^q(E, \mathcal{T}, \mu)$ avec $1/p + 1/q = 1$ et le crochet de dualité est donné par l'intégrale de Lebesgue du produit des fonctions $f \in \mathrm{L}^p$ et $g \in \mathrm{L}^q$.

Dans le cas $p = +\infty$ et $p = 1$, même s'il n'y a de réflexivité, on peut munir $\mathrm{L}^\infty(E, \mathcal{T}, \mu)$ d'une dualité avec $\mathrm{L}^1(E, \mathcal{T}, \mu)$ et vice et versa, le crochet de dualité est toujours donné par l'intégrale de Lebesgue.

5.2.2 Sémantique abstraite par dualité

Nous cherchons à calculer, au pire, une sur-approximation des fonctions valeur Val(PAF) et Val(PIT). On considère donc les éléments de l'espace vectoriel dual qui ont un crochet de dualité positif avec les vecteurs positifs d'un espace vectoriel E muni d'un ordre $\preceq$ (on appelle éléments positifs, les vecteurs $x \in E$ tels que $0 \preceq x$). Notons $E_{\geq 0}$ l'ensemble des vecteurs positifs de E, E^* un ensemble en dualité avec E et soit $\langle \cdot, \cdot \rangle$ le crochet de dualité associé. On appelle *cône dual* de $E_{\geq 0}$ l'ensemble E^*_+ défini par :

$$\{\lambda \in E^* \mid \langle v, \lambda \rangle \geq 0, \ \forall\, v \in E_{\geq 0}\}$$

De plus, nous nous intéressons également aux éléments de E qui agissent positivement sur E^*_+. On appelle le *cône bidual* de $E_{\geq 0}$ l'ensemble $E^{++}_{\geq 0}$ défini par :

$$\{v \in E \mid \langle v, \lambda \rangle \geq 0, \ \forall\, \lambda \in E^*_+\}$$

Soit l'hypothèse suivante :

$$E_{\geq 0} = E^{++}_{\geq 0} \tag{H1}$$

L'hypothèse (H1) dépend évidemment du crochet de dualité choisi. On utilisera l'expression (E, E^*) satisfait l'hypothèse (H1). L'hypothèse (H1) est, par exemple, vérifiée pour les espaces en dualité des exemples 5.9, 5.12 et 5.13, l'ordre utilisé est l'ordre partiel classique sur les fonctions. Si E est un espace de Banach et si E^* désigne le dual topologique de E alors

1. pour des détails sur les mesures signées, le lecteur pourra consulter [Doo93, Chapter 9] ou [Hal74, Sections 28-29]

par des arguments de séparation, (E, E^*) l'hypothèse satisfait l'hypothèse (H1). On notera désormais en abrégé e.v. pour espace vectoriel.

A un élément $w \in \overline{\mathbb{R}}^{\mathbb{P}}$, on associe, si $w > -\infty$ et $\mathrm{dom}(w) \neq \emptyset$, l'ensemble $D(w)$ défini par :

$$\{(Q, E) \mid \emptyset \neq Q \subseteq \mathrm{dom}(w),\ E \text{ s.e.v. de } \mathbb{R}^Q,\ \forall x \in \mathbb{R}^d,\ \mathbf{e}_{x|_Q} \in E,\ w_{|_Q} \in E\} \tag{5.5}$$

Nous considérons de tels ensembles car les problèmes d'optimisation pour calculer la fonction sémantique abstraite 5.2, 5.4 dépendent de la structure des ensembles de contraintes ainsi que des valeurs prises par $v_{\ell(a)}$ et $v_{\ell(\imath)}$. Les ensembles Q dans $D(w)$ représentent les contraintes que l'on va considérer dans les problèmes d'optimisation. Les espaces vectoriels E, dans l'ensemble $D(w)$, déterminent quelle structure algébrique peut être utilisée pour dualiser les contraintes. La figure 5.3 explique l'importance de tels ensembles. On peut remarquer que l'ensemble $D(w)$ n'est jamais vide car il contient toujours $(\mathrm{dom}(w), \mathbb{R}^{\mathrm{dom}(w)})$. Par ailleurs, l'ensemble $D(w)$ contient toujours un couple $(Q, \mathbb{R}^Q)$ où Q est un sous-ensemble fini de $\mathrm{dom}(w)$ et ainsi $\mathbb{R}^Q$ muni d'une norme quelconque est un espace de Banach. Soit (Q, E) un élément de $D(w)$, on munit E de la relation d'ordre des fonctions, et ainsi $E_{\geq 0}$ correspond à $E \cap \mathbb{R}^Q_+$. D'où, $x \in w^{\mathbf{s}}$ équivaut à $(w - \mathbf{e}_x)_{|_{\mathrm{dom}(w)}} \in \mathbb{R}_+^{\mathrm{dom}(w)}$ puis, pour tout espace vectoriel E tel que $(\mathrm{dom}(w), E) \in D(w)$, $x \in w^{\mathbf{s}}$ équivaut à $(w - \mathbf{e}_x)_{|_{\mathrm{dom}(w)}} \in E_{\geq 0}$.

On associe à w l'ensemble $\mathbf{D}(w)$ les couples d'espaces vectoriels (Q, E, E^*) en dualité où (Q, E) est un élément de l'ensemble $D(w)$ et (E, E^*) satisfait l'hypothèse (H1). Plus précisément :

$$\mathbf{D}(w) = \{(Q, E, E^*) \mid (Q, E) \in D(w),\ E \text{ et } E^* \text{ sont en dualité et } (E, E^*) \text{ satisfait (H1)}\}\ .$$

On remarque que l'ensemble $\mathbf{D}(w)$ n'est jamais vide puisque $(\mathrm{dom}(w), \mathbb{R}^{\mathrm{dom}(w)}, \mathbb{R}^{\mathrm{dom}(w)}_{fini})$ appartient à $\mathbf{D}(w)$. De plus, $\mathbf{D}(w)$ contient toujours un élément (Q, E, E^*) tel que E soit un espace de Banach et E^* soit son dual topologique. Soit $(Q, E, E^*) \in \mathbf{D}(w)$, nous notons $\langle \cdot, \cdot \rangle_{Q \times E \times E^*}$ le crochet de dualité associé aux espaces en dualité E, E^*.

Le résultat donné ici est classique en optimisation et permet de réécrire la fonction sémantique abstraite des affectations et des intersections à partir d'une dualité.

Proposition 5.1 (*Sémantique abstraite et bidual*)

- Soient $a \in \mathbb{A}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$. Soit $(\mathrm{dom}(v_{\ell(a)}), E, E^*) \in \mathbf{D}(v_{\ell(a)})$, alors :

$$\sup_{x \in v^{\mathbf{s}}_{\ell(a)}} \mathbf{e}_{T(x)} = \sup_{x \in \mathbb{R}^d} \inf_{\lambda \in E^*_+} \mathbf{e}_{T(x)} + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda \rangle_{\mathrm{dom}(v_{\ell(a)}) \times E \times E^*}$$

- Soient $\imath \in \mathbb{I}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(\imath)} > -\infty$ et $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$. Soit $(\mathrm{dom}(v_{\ell(\imath)}), E, E^*) \in \mathbf{D}(v_{\ell(\imath)})$, alors :

$$\sup_{\substack{x \in v^{\mathbf{s}}_{\ell(\imath)} \\ r(x) \leq 0}} \mathbf{e}_{T(x)} = \sup_{x \in \mathbb{R}^d} \inf_{(\lambda, \mu) \in E^*_+ \times \mathbb{R}_+} \mathbf{e}_{T(x)} + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda \rangle_{\mathrm{dom}(v_{\ell(\imath)}) \times E \times E^*} - \mu r(x)$$

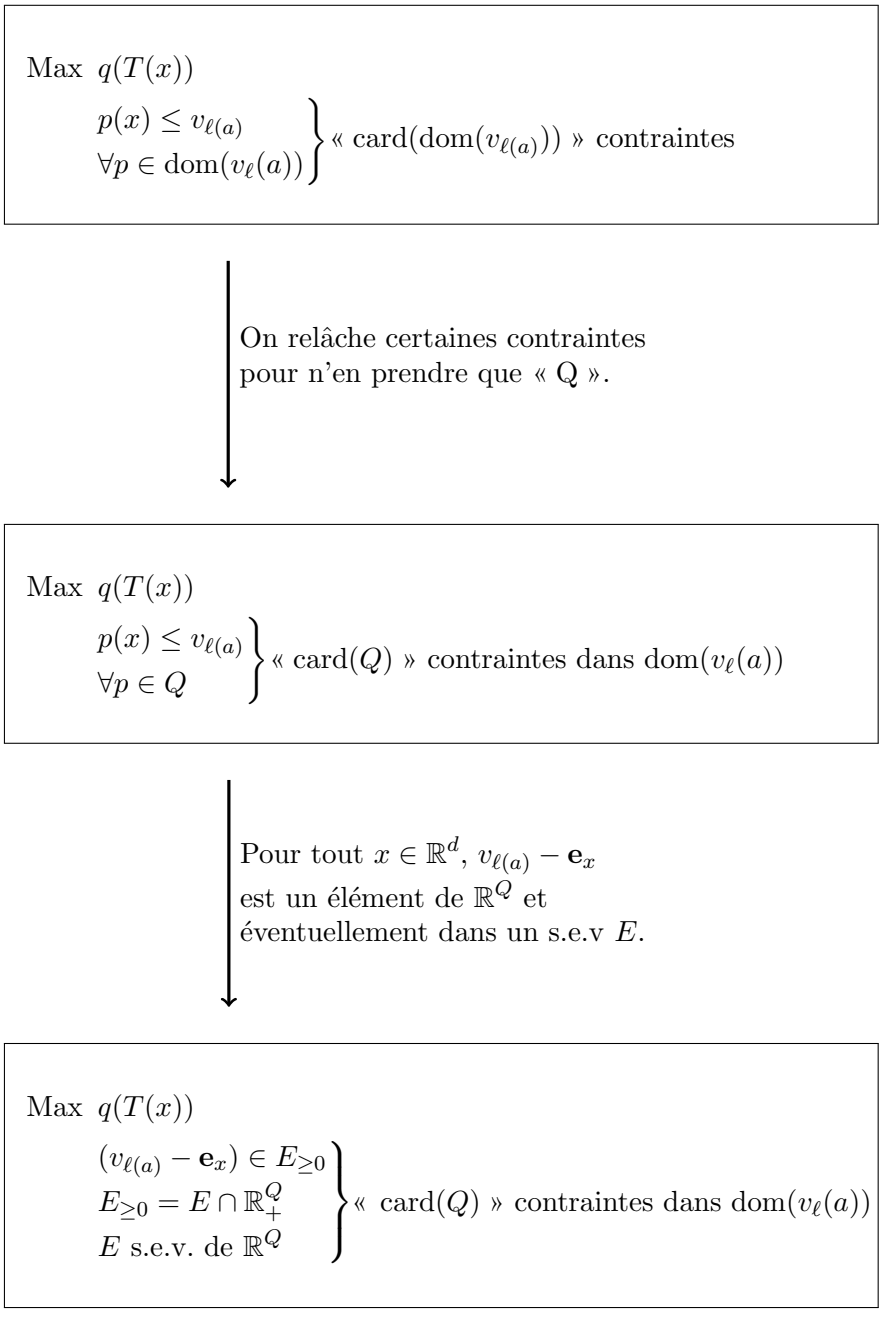


Figure 5.3 – Utilisation possible de l'ensemble (5.5) $D(v_{\ell(a)})$

Nous ne démontrons pas ce résultat qui est un classique en optimisation. Pour une preuve de ce résultat, le lecteur pourra consulter [Roc74]. On déduit de la proposition 5.1 que, pour tout $(\mathrm{dom}(v_{\ell(a)}), E, E^*) \in \mathbf{D}(v_{\ell(a)})$, $(\mathrm{dom}(v_{\ell(\imath)}), F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$:

$$\mathrm{Val}(PAF_q) = \sup_{x\in\mathbb{R}^d} \inf_{\lambda\in E^*_+} q(T(x)) + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda\rangle_{\mathrm{dom}(v_{\ell(a)})\times E\times E^*},$$
$$\mathrm{Val}(PIT_q) = \sup_{x\in\mathbb{R}^d} \inf_{(\lambda,\mu)\in F^*_+\times\mathbb{R}_+} q(T(x)) + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{\mathrm{dom}(v_{\ell(\imath)})\times E\times E^*} - \mu r(x) \ .$$

On aura besoin de définir une fonction sémantique relâchée dont la dualité ne dépend pas de $v_{\ell(a)}$ (resp. $v_{\ell(\iota)}$), par exemple lorsque $v_{\ell(a)}$ (resp. $v_{\ell(\iota)}$) prend la valeur $-\infty$. On introduit l'ensemble $\mathbb{D}(\mathbb{P})$ qui contient tous les couples d'espaces vectoriels (E, E^*) en dualité qui satisfont (H1) tels qu'il existe Q non vide inclus dans $\mathbb{P}$ vérifiant E est un sous-espace vectoriel de $\mathbb{R}^Q$. Plus formellement, on obtient la définition suivante :

$$\mathbb{D}(\mathbb{P}) = \{(Q, E, E^*) \mid \emptyset \neq Q \subseteq \mathbb{P},\ E \text{ s.e.v. de } \mathbb{R}^Q,\ (E, E^*) \text{ satisfait (H1)}\}$$

5.2.3 Affectation et intersection : sémantique relâchée

L'introduction du lagrangien permet d'ajouter les contraintes à la fonction objectif. Cependant, même dans la formulation donnée par la proposition 5.1, le problème n'a pas de bonnes structures. La dualité lagrangienne classique consiste à résoudre un problème qui a de bonnes propriétés. Ce nouveau problème est appelé problème dual et est donné par l'interversion de l'infimum et du supremum dans les égalités de la proposition 5.1. La fonction sémantique relâchée est définie en commutant le supremum et l'infimum de la proposition 5.1. Nous verrons plus tard que cette interversion permet d'avoir de meilleures propriétés sur la fonction sémantique.

Définition 5.2 (*Fonction sémantique relachée d'une affectation*)

Soient $a \in \mathbb{A}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$. Soit maintenant $(Q, E, E^*) \in \mathbf{D}(v_{\ell(a)})$. La fonction sémantique relâchée d'une affectation au point v, $F_a^{Q,E,E^*}(v)$, est l'élément de $\overline{\mathbb{R}}^{\mathbb{P}}$ défini par :

$$F_a^{Q,E,E^*}(v) = \inf_{\lambda \in E^*_+} \sup_{x \in \mathbb{R}^d} \mathbf{e}_{T(x)} + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda \rangle_{Q \times E \times E^*}$$

Si $v_{\ell(a)} \equiv +\infty$, on pose pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$,

$$F_a^{Q,E,E^*}(v) = \sup_{x \in \mathbb{R}^d} \mathbf{e}_{T(x)} \ .$$

S'il existe $p \in \mathbb{P}$ tel que $v_{\ell(a)}(p) = -\infty$, on pose pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$,

$$F_a^{Q,E,E^*}(v) = -\infty \ .$$

Remarque 5.6 (Fonction relâchée standard pour les affectations) Lorsqu'on fixe $q \in \mathbb{P}$, on obtient la valeur suivante, si $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$ et $(Q, E, E^*) \in \mathbf{D}(v_{\ell(a)})$:

$$\left[F_a^{Q,E,E^*}(v)\right](q) = \inf_{\lambda \in E^*_+} \sup_{x \in \mathbb{R}^d} q(T(x)) + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda \rangle_{Q \times E \times E^*} \ .$$

Quand $v_{\ell(a)} \equiv +\infty$, pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$, $[F_a^{Q,E,E^*}(v)](q) = \sup_{x \in \mathbb{R}^d} q(T(x))$, et s'il existe $p \in \mathbb{P}$ tel que $v_{\ell(a)}(p) = -\infty$, pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$, $[F_a^{Q,E,E^*}(v)](q) = -\infty$.

Remarque 5.7 Si $v_{\ell(a)} \equiv +\infty$, la fonction sémantique relâchée coïncide avec la fonction sémantique abstraite. On obtient le même résultat s'il existe $p \in \mathbb{P}$, tel que $v_{\ell(a)}(p) = -\infty$.

Exemple 5.14 (Fonction sémantique relâchée de l'exemple 5.4)
Nous reprenons la configuration $\mathbb{P}$ de l'exemple 5.4 c'est-à-dire l'ensemble $\mathbb{P} = \{(x,y) :\mapsto p_1x + p_2y,\ (p_1,p_2) \in B_2(0,1)\}$. L'ensemble $\mathbb{P}$ s'identifie à la boule unité fermée de $\mathbb{R}^2$. Nous identifions les fonctions de $\overline{\mathbb{R}}^{\mathbb{P}}$ aux fonctions de $\mathbb{R}^2$ dans $\overline{\mathbb{R}}$.

L'affectation du point de contrôle [2] agit sur le point de contrôle [1]. Or d'après l'exemple 5.2, comme $v_1^{\mathbf{s}} = B_2(0,1)$, on a $v_1(p) = 1$, pour tout $p \in \mathbb{P}$, la fonction v_1 est donc continue sur $\mathbb{P} = B_2(0,1)$. Nous utilisons la dualité de l'exemple 5.12. On note $\mathcal{M}$ l'ensemble des mesures signées finies sur $B_2(0,1)$ (pour la tribu borélienne) et nous posons $E = \mathcal{C}(B_2(0,1),\mathbb{R})$, $E^* = \mathcal{M}(B_2(0,1))$ et $(\mathbb{P},E,E^*) \in \mathbf{D}(v_1)$. Le cône dual E_+^* correspond à $\mathcal{M}_+(B_2(0,1))$ l'ensemble des mesures positives finies sur $B_2(0,1)$. Pour cette dualité, pour $q \in \mathbb{P}$, la fonction sémantique relâchée au point de contrôle [2] est donc la fonction :

$$\left[F_2^{\mathbb{P},E,E^*}(v)\right](q) = \inf_{\lambda\in E_+^*} \sup_{(x,y)\in\mathbb{R}^2} q_1y+q_2x+\lambda(B_2(0,1))-x\int_{B_2(0,1)} p_1d\lambda(p)-y\int_{B_2(0,1)} p_2d\lambda(p)\ .$$

Remarquons tout d'abord que :

$$\begin{array}{ll} & \sup_{(x,y)\in\mathbb{R}^2} q_1y + q_2x + \lambda(B_2(0,1)) - x\int_{B_2(0,1)} p_1d\lambda(p) - y\int_{B_2(0,1)} p_2d\lambda(p) < +\infty \\ \text{ssi} & q_1y + q_2x + \int_{B_2(0,1)} p_1d\lambda(p) - y\int_{B_2(0,1)} p_2d\lambda(p) = 0 \end{array}$$

On pose , $q^0 = (q_2,q_1)/\|q\|$ et $\lambda = \|q\|\delta_{q^0}$ (la mesure de Dirac qui charge le vecteur normalisé q^0). On obtient :

$$\begin{array}{ll} & \sup_{(x,y)\in\mathbb{R}^2} q_1y + q_2x + \|q\|\delta_{q^0}(B_2(0,1)) - x\int_{B_2(0,1)} p_1d\lambda(p) - y\int_{B_2(0,1)} p_2d\lambda(p) \\ = & \sup_{(x,y)\in\mathbb{R}^2} q_1y + q_2x + \|q\|\delta_{q^0}(B_2(0,1)) - xq_2d\lambda(p) - yq_1 \\ = & \|q\|\delta_{q^0}(B_2(0,1)) = \|q\| \end{array}$$

On a pour le moment juste montré que $\left[F_2^{\mathbb{P},E,E^*}(v)\right](q) \leq \|q\|$. Nous conclurons plus tard que $\left[F_2^{\mathbb{P},E,E^*}(v)\right](q) = \|q\|$ pour tout $q \in B_2(0,1)$.

Grâce à cet exemple traité de manière compliquée on montre que l'échange des variables x et y effectué entre les points de contrôle [1] et [2] conserve bien la boule unité. Un tel résultat ne pourrait avoir lieu en prenant le domaine des intervalles, des gabarits ou même des polyèdres.

En prenant comme affectation l'identité, on retrouve le calcul de la $\mathbb{P}$-enveloppe convexe. On peut donc évaluer celle-ci par dualité lagrangienne.

Définition 5.3 ($\mathbb{P}$-*enveloppe convexe relachée*)

Soient $w \in \overline{\mathbb{R}}^{\mathbb{P}}$ tel que $w > -\infty$ et $\mathrm{dom}(w) \neq \emptyset$. Soit $(Q, E, E^*) \in \mathbf{D}(w)$. La $\mathbb{P}$-enveloppe convexe relâchée de w est l'élément $\mathrm{vex}_{\mathbb{P}}(w)^{Q,E,E^*}$ défini par :

$$\mathrm{vex}_{\mathbb{P}}(w)^{Q,E,E^*} = \inf_{\lambda \in E^*_+} \sup_{x \in \mathbb{R}^d} \mathbf{e}_x + \langle w - \mathbf{e}_x, \lambda \rangle_{Q \times E \times E^*}$$

Si $w \equiv +\infty$, on pose pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$,

$$\mathrm{vex}_{\mathbb{P}}(w)^{Q,E,E^*} = \sup_{x \in \mathbb{R}^d} \mathbf{e}_x$$

S'il existe $p \in \mathbb{P}$ tel que $w(p) = -\infty$, on pose pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$,

$$\mathrm{vex}_{\mathbb{P}}(w)^{Q,E,E^*} = -\infty$$

Exemple 5.15 ($\mathbb{P}$-enveloppe convexe dans les intervalles)
On revient à l'exemple 5.5. On reprend donc la configuration $\mathbb{P} = \{(x, y) :\mapsto p_1 x + p_2 y,\ (p_1, p_2) \in B_2(0, 1)\}$. On pose $Q = \{(1, 0), (-1, 0), (0, 1), (0, -1)\}$. On introduit l'espace vectoriel $E = \mathbb{R}^Q$ et $E^* = \mathbb{R}^Q$. On prend la fonction v de $\mathbb{P}$ dans $\overline{\mathbb{R}}$ telle que $v(p) = 1$ si $p \in Q$ et $v(p) = +\infty$ si $p \notin Q$. La $\mathbb{P}$-enveloppe convexe relâchée de v est la fonction qui à q associe :

$$\mathrm{vex}_{\mathbb{P}}(v)^{Q,E,E^*}(q) = \inf_{\lambda \in E^*_+} \sup_{(x,y) \in \mathbb{R}^2} q_1 x + q_2 y + \sum_{p \in Q} (1 - (p_1 x + p_2 y))\lambda(p)$$

que l'on peut réécrire :

$$\begin{aligned} \mathrm{vex}_{\mathbb{P}}(v)^{Q,E,E^*}(q) = \inf_{\lambda \in E^*_+} \sup_{(x,y) \in \mathbb{R}^2} & x(q_1 - \lambda(e_1) + \lambda(-e_1)) + y(q_2 - \lambda(e_2) + \lambda(-e_2)) \\ & + \lambda(e_1) + \lambda(-e_1) + \lambda(e_2) + \lambda(-e_2) \end{aligned}$$

où e_1 désigne le vecteur $(1, 0)$ et e_2 désigne le vecteur $(0, 1)$. La valeur $\sup_{(x,y) \in \mathbb{R}^2} x(q_1 - \lambda(e_1) + \lambda(-e_1)) + y(q_2 - \lambda(e_2) + \lambda(-e_2))$ est finie ssi :

$$\begin{cases} q_1 - \lambda(e_1) + \lambda(-e_1) = 0 \\ q_2 - \lambda(e_2) + \lambda(-e_2) = 0 \end{cases}$$

Puisque λ prend des valeurs positives, $q_1 \geq 0$ implique $\lambda(e_1) = q_1$ et $\lambda(-e_1) = 0$ et $q_1 < 0$ implique $\lambda(e_1) = 0$ et $\lambda(-e_1) = -q_1$ et on obtient les mêmes implications pour q_2, $\lambda(e_2)$ et $\lambda(-e_2)$. On retrouve les résultats de l'exemple 5.5 :

$$\mathrm{vex}_{\mathbb{P}}(v)^{Q,E,E^*}(q) = \sup\{-p_1 - p_2,\ -p_1 + p_2,\ p_1 - p_2,\ p_1 + p_2\}$$

Ce résultat découle en fait du théorème de dualité forte en programmation linéaire que nous exposerons plus tard.

Exemple 5.16 ($\mathbb{P}$-enveloppe convexe pathologique)
Nous reprenons la configuration et la fonction de l'exemple 5.6 : $\mathbb{P} = \{p \in \mathcal{C}(\mathbb{R}, \mathbb{R}) \mid p(x) = 0,\ \forall x \notin [0, 1], p(1/2) \leq 0\}$ et on définit $v(p) = p(1/2)^2$ pour tout $p \in \mathbb{P}$. Nous avions montré

que la fonction $\text{vex}_{\mathbb{P}}(v)$ était identiquement nulle. Nous considérons maintenant le crochet de dualité entre $\mathbb{R}^{\mathbb{P}}$ et $\mathbb{R}^{\mathbb{P}}_{fini}$. Nous calculons donc, en posant $E = \mathbb{R}^{\mathbb{P}}$ et $E^* = \mathbb{R}^{\mathbb{P}}_{fini}$, $\text{vex}_{\mathbb{P}}(v)^{\mathbb{P},E,E^*}$. Soit $p \in \mathbb{P}$, par définition,

$$\text{vex}_{\mathbb{P}}(v)^{\mathbb{P},E,E^*}(p) = \inf_{\lambda \in E^*_+} \sup_{x\in[0,1]} p(x) + \sum_{q\in\mathbb{P}} \lambda(q)\left((q(1/2))^2 - q(x)\right)$$

les sommes sont finies car λ est non nulle sur un nombre fini d'élément $q \in \mathbb{P}$. Pour tout $p \in \mathbb{P}$, la fonction $x \mapsto \max(p(x), 0)$ appartient à $\mathbb{P}$. Considérons la fonction λ telle que $\lambda(q)$ vaut 1 si $q = \max(p, 0)$ et 0 sinon. On obtient $\text{vex}_{\mathbb{P}}(v)^{\mathbb{P},E,E^*}(p) \leq \sup_{x\in[0,1]} p(x) + \max(p(1/2), 0))^2 - \max(p(x), 0) = \sup_{x\in[0,1]} p(x) - \max(p(x), 0) = 0$.

Comme pour les affectations, on définit la fonction sémantique relâchée d'une intersection en intervertissant le supremum et l'infimum dans la proposition 5.1.

Définition 5.4 (*Fonction sémantique relachée d'une intersection*)

Soient $\imath \in \mathbb{I}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(\imath)} > -\infty$ et $\text{dom}(v_{\ell(\imath)}) \neq \emptyset$. Soit maintenant $(Q, E, E^*) \in \mathbf{D}(v_{\ell(\imath)})$. La fonction sémantique relâchée d'une intersection au point v, $F_\imath^{Q,E,E^*}$, est l'élément de $\overline{\mathbb{R}}^{\mathbb{P}}$ défini par :

$$F_\imath^{Q,E,E^*}(v) = \inf_{(\lambda,\mu)\in E^*_+\times\mathbb{R}_+} \sup_{x\in\mathbb{R}^d} \mathbf{e}_{T(x)} + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{Q\times E\times E^*} - \mu r(x) \ .$$

Si $v_{\ell(\imath)} \equiv +\infty$, on pose pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$,

$$F_\imath^{Q,E,E^*}(v) = \inf_{\mu\in\mathbb{R}_+} \sup_{x\in\mathbb{R}^d} \mathbf{e}_{T(x)} - \mu r(x) \ .$$

S'il existe $p \in \mathbb{P}$ tel que $v_{\ell(\imath)}(p) = -\infty$, on pose pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$,

$$F_\imath^{Q,E,E^*}(v) = -\infty \ .$$

Remarque 5.8 (Fonction relâchée standard pour les intersections) Lorsqu'on fixe $q \in \mathbb{P}$, on obtient la valeur suivante :

$$\left[F_\imath^{Q,E,E^*}(v)\right](q) = \inf_{(\lambda,\mu)\in E^*_+\times\mathbb{R}_+} \sup_{x\in\mathbb{R}^d} q(T(x)) + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{Q\times E\times E^*} - \mu r(x) \ .$$

Quand $v_{\ell(\imath)} \equiv +\infty$, pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$, $[F_\imath^{Q,E,E^*}(v)](q) = \inf_{\mu\in\mathbb{R}_+} \sup_{x\in\mathbb{R}^d} q(T(x)) - \mu r(x)$ et s'il existe $p \in \mathbb{P}$ tel que $v_{\ell(\imath)}(p) = -\infty$, pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$, $[F_\imath^{Q,E,E^*}(v)](q) = -\infty$.

Remarque 5.9 S'il existe $p \in \mathbb{P}$, tel que $v_{\ell(\imath)}(p) = -\infty$, la fonction sémantique relâchée coïncide avec la fonction sémantique abstraite.

Exemple 5.17 (Fonction sémantique relâchée de l'exemple 5.7)
Nous reprenons la configuration $\mathbb{P}$ de l'exemple 5.4 c'est-à-dire l'ensemble $\mathbb{P} = \{(x,y) :\mapsto p_1x + p_2y,\ (p_1,p_2) \in B_2(0,1)\}$. L'ensemble $\mathbb{P}$ s'identifie à la boule unité fermée de $\mathbb{R}^2$. Nous identifions les fonctions de $\overline{\mathbb{R}}^{\mathbb{P}}$ aux fonctions de $\mathbb{R}^2$ dans $\overline{\mathbb{R}}$.

Si v_2 est continue sur $\mathbb{P}$, on peut prendre la dualité $E = \mathcal{C}(B_2(0,1),\mathbb{R})$, $E^* = \mathcal{M}(B_2(0,1)$ et $(\mathbb{P},E,E^*) \in \mathbf{D}(v_2)$. Pour cette dualité, pour $q \in \mathbb{P}$, la fonction sémantique relâchée au point de contrôle [3] est par conséquent la fonction :

$$\left[F_3^{Q,E,E^*}(v)\right](q) = \inf_{\substack{\lambda\in E_+^* \\ \mu\in\mathbb{R}_+}} \sup_{(x,y)\in\mathbb{R}^2} q_1(2y+3x^2) + q_2(0.5y + 0.2\exp(x)) + \mu(1-x^2-y^2) + \int_{B_2(0,1)} v_2(p)d\lambda(p) - x\int_{B_2(0,1)} p_1 d\lambda(p) - y\int_{B_2(0,1)} p_2 d\lambda(p)\ .$$

Si v_2 n'est pas continue sur $\mathbb{P}$ ne prend jamais la valeur $-\infty$ et $\mathrm{dom}(v_2)$ est non vide, on peut prendre la dualité $E = \mathbb{R}^{\mathrm{dom}(v_2)}$, $E^* = \mathbb{R}_{fini}^{\mathrm{dom}(v_2)}$ ainsi $(\mathrm{dom}(v_2),E,E^*) \in \mathbf{D}(v_2)$. Remarquons que E_+^* correspond aux fonctions à valeurs positives ou nulles qui sont non nulles sur un nombre fini d'éléments $p \in \mathbb{P}$. Pour cette dualité, pour $q \in \mathbb{P}$, la fonction sémantique relâchée au point de contrôle [3] est par conséquent la fonction :

$$\left[F_3^{\mathrm{dom}(v_2),E,E^*}(v)\right](q) = \inf_{\substack{\lambda\in E_+^* \\ \mu\in\mathbb{R}_+}} \sup_{(x,y)\in\mathbb{R}^2} q_1(2y+3x^2) + q_2(0.5y + 0.2\exp(x)) + \sum_{\substack{\mathrm{supp}\,\lambda \\ p\in\mathrm{dom}(v_2)}} \lambda(p)\left(v_2(p) - p\cdot(x,y)\right) + \mu(1-x^2-y^2)\ .$$

Problèmes duaux

Si, pour tout $a \in \mathbb{A}$, $v_{\ell(a)} > -\infty$ et pour tout $\imath \in \mathbb{I}$, $v_{\ell(\imath)} > -\infty$, la fonction sémantique relâchée est la fonction identiquement égale à $-\infty$. Lorsque ces fonctions sont identiquement égales à $+\infty$, les problèmes d'optimisation à résoudre pour calculer la fonction sémantique abstraite ou relâchée ne dépendent pas de $v_{\ell(a)}$ ou $v_{\ell(\imath)}$.

Pour tout $a \in \mathbb{A}$, calculer la fonction sémantique relâchée au point $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)})$ est non vide revient à résoudre la famille de problèmes de minimisation 5.5 dont les contraintes ne dépendent pas du degré de liberté $v_{\ell(a)}$ mais uniquement de la dualité choisie. Fixons donc, pour un élément $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$, un élément $(Q,E,E^*) \in \mathbf{D}(v_{\ell(a)})$.

Problème 5.5 (Problème dual d'une affectation)
Résoudre le problème de minimisation :

$$\underset{\lambda^q\in E_+^*}{\mathrm{Min}}\ [\psi_a(q)](\lambda^q) \qquad (DAFq)$$

où $[\psi_a(q)](\lambda^q) = \underset{x\in\mathbb{R}^d}{\mathrm{Max}}\ q(T(x)) + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda^q\rangle_{Q\times E\times E^*}$

On pourrait ajouter une contrainte supplémentaire sur l'ensemble des éléments duaux aux vecteurs. En effet, il faudrait considérer uniquement les vecteurs duaux $\lambda^q \in E_+^*$ tels que :

$$[\psi_a(q)]\,(\lambda^q) < +\infty \ .$$

Le problème dual d'un problème de calcul de $\mathbb{P}$-enveloppe convexe est donc le problème 5.5 où l'application T est l'identité.

De même, pour tout $\imath \in \mathbb{I}$, calculer la fonction relâchée au point $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ satisfaisant $v_{\ell(\imath)} > -\infty$ et $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$ revient à résoudre la famille de problèmes de minimisation 5.6 dont les contraintes ne dépendent uniquement que de dualité choisie. Fixons donc, pour un élément $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(\imath)} > -\infty$ et $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$, un élément $(Q, E, E^*) \in \mathbf{D}(v_{\ell(\imath)})$.

Problème 5.6 (Problème dual d'une intersection)

Résoudre le problème de minimisation :

$$\underset{(\lambda^q,\mu^q)\in E^*_+\times\mathbb{R}_+}{\mathrm{Min}} [\psi_\imath(q)]\,(\lambda^q,\mu^q) \qquad (DITq)$$

où $[\psi_\imath(q)]\,(\lambda^q,\mu^q) = \underset{x\in\mathbb{R}^d}{\mathrm{Max}}\ q(T(x)) + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda^q\rangle_{Q\times E\times E^*} - \mu^q r(x)$

On pourrait ajouter une contrainte supplémentaire sur l'ensemble des éléments duaux aux vecteurs. En effet, il faudrait considérer uniquement les vecteurs duaux $(\lambda^q, \mu^q) \in E^*_+ \times \mathbb{R}_+$ tels que :

$$[\psi_\imath(q)]\,(\lambda^q,\mu^q) < +\infty \ .$$

Proposition 5.2 (*Problèmes duaux convexes*)

Les problèmes 5.5 et 5.6 sont des problèmes de minimisation convexe.

Nous démontrons ce résultat classique pour la commodité du lecteur.

Démonstration

Ce résultat est essentiellement dû au fait que les fonctions $\lambda \mapsto \langle v_{\ell(a)} - \mathbf{e}_x, \lambda\rangle_{Q\times E\times E^*}$ et $(\lambda,\mu) \mapsto \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{Q\times E\times E^*} - \mu r(x)$ sont des fonctions linéaires et donc convexes par conséquent le supremum pris sur tout $\mathbb{R}^d$ définit également des fonctions convexes d'où la convexité des fonctions ψ_a et $\psi_\imath$. On conclut que les problèmes sont convexes en remarquant que les ensembles E^*_+ et $\mathbb{R}_+$ sont des ensembles convexes. ∎

5.2.4 Sémantique relâchée des unions

Le problème 5.4 est assez compliqué à résoudre puisque contrairement à l'intersection, nous ne pouvons pas concaténer les contraintes. Nous ne tentons pas de le résoudre par des méthodes directes. Nous cherchons simplement une borne valide, c'est-à-dire une fonction sur $\overline{\mathbb{R}}^{\mathbb{P}}$ qui sur-approxime la fonction sémantique abstraite. On utilise la proposition 5.3.

Proposition 5.3 (*Encadrement de l'union*)

Pour tout $v, w \in \overline{\mathbb{R}}^{\mathbb{P}}$:

$$\sup(\text{vex}_{\mathbb{P}}(v), \text{vex}_{\mathbb{P}}(w)) \leq (v^\star \cup w^\star)^\dagger \leq \text{vex}_{\mathbb{P}}(\sup(v, w)) \ .$$

Si, de plus, $v, w \in \text{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ alors :

$$(v^\star \cup w^\star)^\dagger = \sup(v, w) \ . \tag{5.6}$$

Démonstration
Soient $v, w \in \overline{\mathbb{R}}^{\mathbb{P}}$. Par monotonie de l'application $u \mapsto u^\star$, $v^\star \subseteq v^\star \cup w^\star$ et $w^\star \subseteq v^\star \cup w^\star$ et par monotonie de l'application $C \mapsto C^\dagger$, on obtient, $(v^\star)^\dagger \leq (v^\star \cup w^\star)^\dagger$ et $(w^\star)^\dagger \leq (v^\star \cup w^\star)^\dagger$, on conclut que $\sup(\text{vex}_{\mathbb{P}}(v), \text{vex}_{\mathbb{P}}(w)) \leq (v^\star \cup w^\star)^\dagger$. Soit $x \in v^\star \cup w^\star$, on a donc $p(x) \leq v(p)$ pour tout $p \in \mathbb{P}$ ou $p(x) \leq w(p)$ pour tout $p \in \mathbb{P}$, on conclut que $p(x) \leq \sup(v, w)(p)$ pour tout $p \in \mathbb{P}$ d'où, $x \in \sup(v, w)^\star$. Finalement, par monotonie de $C \mapsto C^\dagger$, on conclut que $(v^\star \cup w^\star)^\dagger \leq \text{vex}_{\mathbb{P}}(\sup(v, w))$.

Si $v, w \in \text{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$, $\sup(\text{vex}_{\mathbb{P}}(v), \text{vex}_{\mathbb{P}}(w)) = \sup(v, w)$ et comme $\text{vex}_{\mathbb{P}}(\sup(v, w)) \leq \sup(v, w)$, on conclut d'après l'inégalité précédente que $(v^\star \cup w^\star)^\dagger = \sup(v, w)$. ∎

Remarque 5.10 Pour tout $v \in \overline{\mathbb{R}}^{\mathbb{P}}$, $\text{vex}_{\mathbb{P}}(v)$ est toujours plus petite que v, on déduit que $(v^\star \cup w^\star)^\dagger \leq \sup(v, w)$.

A présent, nous pouvons définir la sémantique relâchée pour l'union toujours en s'assurant que celle-ci majore la fonction sémantique abstraite.

Définition 5.5 (*Fonction sémantique relachée pour l'union*)

Soit $u \in \mathbb{U}$. La coordonnée de la fonction sémantique relâchée est la fonction $F_u^{\mathcal{R}} : (\overline{\mathbb{R}}^{\mathbb{P}})^n \mapsto \overline{\mathbb{R}}^{\mathbb{P}}$ définie par :

$$F_u^\vee(v) = \sup(v_{\ell_g(u)}, v_{\ell_d(u)}) \ .$$

Exemple 5.18 (Fonction sémantique relâchée de l'union)
Sur le programme de la figure 5.1, au point de contrôle [3], la boucle **while** entraîne une opération d'union : la fonction sémantique concrète au point de contrôle [3] est :

$$X_3 = X_2 \cup X_7$$

Pour rappel, nous trouvons ces deux coordonnées en remarquant que les moyens d'entrée dans la boucle **while** sont soit, si la boucle **while** n'a jamais été parcourue, de venir du point de contrôle [2], soit, si la boucle **while** a déjà été parcourue, de revenir après l'opération interne de la boucle c'est-à-dire le point de contrôle [7].

Soit une configuration $\mathbb{P}$. Soit $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^8$, les ensembles $v_i^{\mathbf{s}}$ s'entendent donc par rapport à cette configuration. La fonction sémantique abstraite au point de contrôle [3] est donc la fonction qui à $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^8$ associe :

$$F_u^\vee(v) = \sup(v_2, v_7) \ .$$

D'après l'équation (5.6), si les opérandes de l'union $v_{\ell_g(u)}$ et $v_{\ell_d(u)}$ sont des fonctions $\mathbb{P}$-convexes, la fonction sémantique abstraite d'une union est égale à la fonction sémantique relâchée :

$$F_u^\sharp(v) = F_u^\vee(v) \ .$$

5.3 Egalités entre sémantiques abstraites et relâchées

Dans cette section, nous étudions les configurations et les conditions pour lesquelles nous avons égalité entre la fonction sémantique abstraite et la fonction sémantique relâchée. Pour l'union, nous avons vu que si les opérandes étaient des éléments $\mathbb{P}$-convexes alors $F_u^\vee = F_u^\sharp$. Pour les affectations et les intersections, nous devrons utiliser des techniques d'analyse convexe et d'optimisation pour obtenir l'égalité entre la fonction sémantique relâchée et la fonction sémantique abstraite. Le théorème d'égalité découlera du théorème dit de dualité forte : la valeur du problème primal coïncide avec la valeur duale.

5.3.1 Affectations concaves et tests convexes

Pour établir un résultat d'égalité, on utilise un résultat de dualité forte en programmation semi-infinie de Shapiro [Sha05]. Soient $K \subseteq E$ un cône de l'espace vectoriel E, f une fonction de $\mathbb{R}^d$ dans $\mathbb{R} \cup \{-\infty\}$ et $G : \mathbb{R}^d \mapsto E$. On s'intéresse au problème de maximisation suivant :

$$\text{Max} \ \ f(x) \ \ \text{s.c} \ \ G(x) \in K \tag{P}$$

On note $\text{Sol}(P)$ l'ensemble des solutions du problème (P) et $\text{Val}(P)$ la valeur du problème (P). Soit E^* un dual de E et $\langle \cdot, \cdot \rangle$ le crochet de dualité. On pose $L(x, \lambda) = f(x) + \langle G(x), \lambda \rangle$ et notons K^+ le cône dual de K. On a $\text{Val}(P) = \sup_{x \in \mathbb{R}^d} \inf_{\lambda \in K^+} L(x, \lambda)$ et on pose $\text{Val}(D) = \inf_{\lambda \in K^+} \sup_{x \in \mathbb{R}^d} L(x, \lambda)$.

Proposition 5.4 (*Proposition 3.4 [Sha05]*)

Supposons $\text{Sol}(P)$ non vide et borné. De plus, si le cône bidual de K^{++} est égal à K et si pour tout $\lambda \in K^+$, $x \mapsto L(x, \lambda)$ est une fonction concave semi-continue supérieurement telle qu'il existe x^λ vérifiant $L(x^\lambda, \lambda) > -\infty$ alors $\text{Val}(P) = \text{Val}(D)$.

Dans notre contexte, la fonction f est une fonction $q \in \mathbb{P}$, la fonction G est la fonction de $\mathbb{R}^d$ dans un espace vectoriel F tel que $(\text{dom}(v_{\ell(a)}), F) \in D(v_{\ell(a)})$, $v_{\ell(a)} - \mathbf{e}_x$ dans le cas d'une affectation et la fonction de $\mathbb{R}^d$ dans un espace vectoriel H tel que $(\text{dom}(v_{\ell(\imath)}), F) \in D(v_{\ell(\imath)})$, $v_{\ell(\imath)} - \mathbf{e}_x$ dans le cas d'une intersection. De plus, le cône K est $F_{\geq 0}$ (pour rappel $F \cap \mathbb{R}_+^{\text{dom}(v_{\ell(a)})}$) dans le cas d'une affectation et $H_{\geq 0}$ (pour rappel $H \cap \mathbb{R}_+^{\text{dom}(v_{\ell(\imath)})}$) dans le cas d'une intersection. Le lagrangien L correspond soit à $q(T(x)) + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda \rangle_{\text{dom}(v_{\ell(a)}) \times F \times F^*}$ soit à $q(T(x)) + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda \rangle_{\text{dom}(v_{\ell(\imath)}) \times H \times H^*} - \mu r(x)$ où $(\text{dom}(v_{\ell(a)}), F, F^*) \in \mathbf{D}(v_{\ell(a)})$ et $(\text{dom}(v_{\ell(\imath)}), H, H^*) \in \mathbf{D}(v_{\ell(\imath)})$.

Théorème 5.1 (*Egalité entre fonction sémantique abstraite et relachée*)

On note T_a l'application associée à la coordonnée $a \in \mathbb{A}$ (resp. $T_\imath$ pour $\imath \in \mathbb{I}$). On note $r_\imath$ le test associé à la coordonnée $\imath \in \mathbb{I}$.

Soient $a \in \mathbb{A}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$. Soit $q \in \mathbb{P}$ tel que l'ensemble $\mathrm{Sol}(PAF_q)$ soit non vide et borné. Si, pour tout $p \in \mathrm{dom}(v_{\ell(a)})$, p est convexe et si $q \circ T_a$ est concave alors pour tout $(\mathrm{dom}(v_{\ell(a)}), E, E^*) \in \mathbf{D}(v_{\ell(a)})$:

$$\left[F_a^\sharp(v)\right](q) = \left[F_a^{\mathrm{dom}(v_{\ell(a)}),E,E^*}(v)\right](q) \ .$$

Soient $\imath \in \mathbb{I}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(\imath)} > -\infty$ et $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$. Soit $q \in \mathbb{P}$ tel que l'ensemble $\mathrm{Sol}(PIT_q)$ soit non vide et borné. Supposons que $r_\imath$ est une fonction convexe. Si, pour tout $p \in \mathrm{dom}(v_{\ell(\imath)})$, p est convexe et si $q \circ T_\imath$ est concave alors pour tout $(\mathrm{dom}(v_{\ell(\imath)}), F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$:

$$\left[F_\imath^\sharp(v)\right](q) = \left[F_\imath^{\mathrm{dom}(v_{\ell(\imath)}),F,F^*}(v)\right](q) \ .$$

Soit $u \in \mathbb{U}$. Si, $v_{\ell_g(u)}$ et $v_{\ell_d(u)}$ sont des fonctions $\mathbb{P}$-convexes alors :

$$F_u^\sharp(v) = F_u^\vee(v) \ .$$

Démonstration

Soient $\imath \in \mathbb{I}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$. Soit $q \in \mathbb{P}$ tel que $\mathrm{Sol}(PIT_q)$ est non vide est borné. Soit $(\mathrm{dom}(v_{\ell(\imath)}), F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$. Comme $\mathrm{Sol}(PIT_q)$ est non vide alors $v^{\mathbf{s}}_{\ell(\imath)} \cap r_\imath^{-1}(\mathbb{R}_-)$ est non vide. Par ailleurs $q \circ T_\imath$ est concave et à valeur finie donc continue sur $\mathbb{R}^d$. De plus, comme tout $p \in \mathrm{dom}(v_{\ell(\imath)})$ est convexe alors, pour tout $(\lambda, \mu) \in F^*_+ \times \mathbb{R}_+$, $x \mapsto q(T_\imath x) + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{\mathrm{dom}(v_{\ell(\imath)})\times F\times F^*} - \mu r_\imath(x)$ est concave. En effet, si, pour tout $p \in \mathrm{dom}(v_{\ell(\imath)})$, p est convexe alors $\mathbf{e}_{tx+(1-t)x'}(p) \leq t\mathbf{e}_x(p) + (1-t)\mathbf{e}_{x'}(p)$ pour tout $p \in \mathrm{dom}(v_{\ell(\imath)})$, pour tout $x, x' \in \mathbb{R}^d$ et pour tout $t \in [0,1]$; on en déduit que, pour tout $x, x' \in \mathbb{R}^d$ et pour tout $t \in [0,1]$, $v_{\ell(\imath)} - \mathbf{e}_{tx+(1-t)x'} - (v_{\ell(\imath)} - t\mathbf{e}_x + (1-t)\mathbf{e}_{x'}) \geq 0$ puis, par définition de F et F^*_+, $\langle v_{\ell(\imath)} - \mathbf{e}_{tx+(1-t)x'} - (v_{\ell(\imath)} - t\mathbf{e}_x + (1-t)\mathbf{e}_{x'}), \lambda\rangle_{\mathrm{dom}(v_{\ell(\imath)})\times F\times F^*} \geq 0$ pour tout $x, x' \in \mathbb{R}^d$, pour tout $t \in [0,1]$ et pour tout $\lambda \in F^*_+$; enfin $\langle v_{\ell(\imath)} - \mathbf{e}_{tx+(1-t)x'}, \lambda\rangle_{\mathrm{dom}(v_{\ell(\imath)})\times F\times F^*} \geq \langle v_{\ell(\imath)} - t\mathbf{e}_x + (1-t)\mathbf{e}_{x'}, \lambda\rangle_{\mathrm{dom}(v_{\ell(\imath)})\times F\times F^*}$ pour tout $x, x' \in \mathbb{R}^d$, pour tout $t \in [0,1]$ et pour tout $\lambda \in F^*_+$. En outre, $x \mapsto q(T_\imath x) + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{\mathrm{dom}(v_{\ell(\imath)})\times F\times F^*} - \mu r_\imath(x)$ à valeurs finies donc continue sur $\mathbb{R}^d$. On conclut par la proposition 3.4 de [Sha05] (proposition 5.4 ici). On utilise le même raisonnement pour les affectations. Pour les unions, le résultat découle de l'équation (5.6). ■

Les deux fonctions sémantiques coïncident également lorsque $v_{\ell(a)}$ et $v_{\ell(\imath)}$ prennent la valeur $-\infty$. Lorsque l'on s'intéresse aux affectations, les deux fonctions sémantiques coïncident dès que $v_{\ell(a)}$ est identiquement égale à $+\infty$. Pour les intersections, l'hypothèse des tests convexes est essentielle (mais pas nécessaire) pour assurer l'égalité des deux fonctions sémantiques : ce résultat est assuré par le théorème de dualité forte de la maximisation sous un nombre fini de contraintes.

Exemple 5.19 (Retour à l'exemple 5.14)

Nous revenons à l'exemple 5.14 pour achever la preuve. Le problème :

$$\mathrm{Max}\{q_1 y + q_2 x \mid p \cdot (x,y) \leq v_1(p), \ \forall p \in B_2(0,1)\}$$

est un problème de maximisation d'une fonction continue sur un ensemble compact, l'ensemble des solutions est donc non vide et borné. Les fonctions $(x, y) \mapsto p_1 x + p_2 y$ sont linéaires et donc convexes et la fonction $(x, y) \mapsto q_1 y + q_2 x$ est linéaire donc concave et v_1 ne prend pas la valeur $-\infty$ et est finie. D'après le théorème 5.1, il y a égalité entre la fonction sémantique relâchée définie à l'exemple 5.14 et la fonction sémantique abstraite définie à l'exemple 5.4.

5.3.2 Configurations bornées

Les configurations couramment utilisées en analyse statique contiennent la base duale. Par exemple, dès que les bornes des intervalles sont finies, nous manipulons des ensembles compacts convexes ce qui assure que les problèmes d'optimisation à résoudre sur ces ensembles ont des valeurs finies et des solutions optimales convexes compactes non vides. On peut généraliser cette idée en prenant une configuration $\mathbb{P}$ qui, dès qu'un degré de liberté est fini sur certaines fonctions de base, assure que le $\mathbb{P}$-sous niveau est compact.

Définition 5.6 (*Configuration bornée*)

On dit que $\mathbb{P}$ est une configuration bornée si :

1. $\mathbb{P}$ est un sous-ensemble de $\mathcal{C}(\mathbb{R}^d, \mathbb{R})$ l'ensemble des fonctions continues de $\mathbb{R}^d$ dans $\mathbb{R}$.
2. Il existe $B \subseteq \mathbb{P}$ tel que, pour tout $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ vérifiant $v(p) > -\infty$ pour tout $p \in \mathbb{P}$, $B \subseteq \mathrm{dom}(v)$ et $v^{\mathbf{s}} \neq \emptyset$, $v^{\mathbf{s}}$ est une partie bornée de $\mathbb{R}^d$.

Par la suite, lorsque $\mathbb{P}$ est une configuration bornée, on notera $\mathcal{B}(\mathbb{P})$, l'ensemble suivant :

$$\{B \subseteq \mathbb{P} \mid (B \subseteq \mathrm{dom}(v),\ v^{\mathbf{s}} \neq \emptyset) \implies v^{\mathbf{s}} \text{ borné}\}$$

Exemple 5.20 (Configuration bornée classique d'interprétation abstraite)
Toute configuration contenant $\{\pm e_i^\star\}$ où $\{e_i^\star\}$ est la base duale de $\mathbb{R}^d$ est une configuration bornée. En d'autres termes, les domaines des intervalles, zones de Miné et gabarits de Sankaranarayanan et al sont des configurations bornées.

Exemple 5.21 (Configurations quadratiques 1)
Toute configuration contenant une fonction $q : x \mapsto x^T L x$ où L est une matrice définie positive est une configuration bornée.

Exemple 5.22 (Configurations quadratiques 2)
Toute configuration contenant des fonctions q_i de la forme $q_i : x \mapsto x^T L^i x$ où L^i sont des matrices diagonales telles que $L^i_{kk} > 0$ si $k = i$ et 0 sinon.

Soit une configuration bornée $\mathbb{P}$, et soient $a \in \mathbb{A}$, $B \in \mathcal{B}(\mathbb{P})$ et un élément v dans $(\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $B \subseteq \mathrm{dom}(v_{\ell(a)})$. Si $v^{\mathbf{s}}_{\ell(a)} \neq \emptyset$ alors :

$$\forall\, p \in \mathbb{P},\ \exists\, x \in v^{\mathbf{s}}_{\ell(a)},\ \text{tel que } p(Tx) = \left[F_a^\sharp(v)\right](p)$$

et, de plus, pour tout $p \in \mathbb{P}$, l'ensemble des $x \in v^{\mathbf{s}}_{\ell(a)}$ qui satisfait l'égalité est compact. On obtient le même résultat sur les intersections dès que $v^{\mathbf{s}}_{\ell(i)} \cap r^{-1}(\mathbb{R}_-) \neq \emptyset$.

Nous déduisons immédiatement le corollaire suivant :

Corollaire 5.1 (*Dualité forte et configurations bornées*)

On note T_a l'application associée à la coordonnée $a \in \mathbb{A}$ (resp. $T_\imath$ pour $\imath \in \mathbb{I}$). On note $r_\imath$ le test associé à la coordonnée $\imath \in \mathbb{I}$. Soit $\mathbb{P}$ est une configuration bornée.

Soient $a \in \mathbb{A}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$ et $v^{\mathbf{s}}_{\ell(a)} \neq \emptyset$. On suppose qu'il existe $B' \in \mathcal{B}(\mathbb{P})$ tel que $B' \subseteq \mathrm{dom}(v_{\ell(a)})$ et que pour tout $p \in \mathrm{dom}(v_{\ell(a)})$, p soit convexe. Soit $q \in \mathbb{P}$ telle que $q \circ T_a$ est concave alors pour tout $(\mathrm{dom}(v_{\ell(a)}), E, E^*) \in \mathbf{D}(v_{\ell(a)})$:

$$\left[F^\sharp_a(v)\right](q) = \left[F_a^{\mathrm{dom}(v_{\ell(a)}),E,E^*}(v)\right](q) \ .$$

Soient $\imath \in \mathbb{I}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$ et $v^{\mathbf{s}}_{\ell(\imath)} \cap (r_\imath)^{-1}(\mathbb{R}_-) \neq \emptyset$. On suppose qu'il existe $B \in \mathcal{B}(\mathbb{P})$ tel que $B \subseteq \mathrm{dom}(v_{\ell(\imath)})$ et que pour tout $p \in \mathrm{dom}(v_{\ell(\imath)})$, p soit convexe. Supposons, de plus, que $r_\imath$ est une fonction convexe. Soit $q \in \mathbb{P}$ telle que $q \circ T_\imath$ est concave alors pour tout $(\mathrm{dom}(v_{\ell(\imath)}), F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$:

$$\left[F^\sharp_\imath(v)\right](q) = \left[F_\imath^{\mathrm{dom}(v_{\ell(\imath)}),F,F^*}(v)\right](q) \ .$$

5.3.3 Problèmes d'optimisation à contraintes finies

Dans les résultats précédents, nous n'avons émis aucune hypothèse sur la cardinalité de $\mathbb{P}$ mais les résultats énoncés plus tôt restent vrais si $\mathrm{dom}(v_{\ell(a)})$ ou $\mathrm{dom}(v_{\ell(\imath)})$ sont des ensembles finis. Dans cette sous-section, nous allons revenir à des problèmes de maximisation avec un nombre fini de contraintes. Ce type de résultats peut s'appliquer aux domaines classiquement utilisés en interprétation abstraite qui, en plus de la finitude des contraintes, utilisent la linéarité des fonctions de base $p \in \mathbb{P}$.

Théorème 5.2 (*Problèmes à domaines finis*)

On note T_a l'application linéaire associé à la coordonnée $a \in \mathbb{A}$ (resp. $T_\imath$ pour $\imath \in \mathbb{I}$). On note $r_\imath$ le test associé à la coordonnée $\imath \in \mathbb{I}$.

Soient $a \in \mathbb{A}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$. Supposons que, pour tout $p \in \mathrm{dom}(v_{\ell(a)})$, p est linéaire et que $\mathrm{dom}(v_{\ell(a)})$ est un ensemble fini. Si $q \circ T_a$ est linéaire et $v^{\mathbf{s}}_{\ell(a)} \neq \emptyset$ alors pour tout $(\mathrm{dom}(v_{\ell(a)}), E, E^*) \in \mathbf{D}(v_{\ell(a)})$:

$$\left[F^\sharp_a(v)\right](q) = \left[F_a^{\mathrm{dom}(v_{\ell(a)}),E,E^*}(v)\right](q) \ .$$

Soient $\imath \in \mathbb{I}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(\imath)} > -\infty$ et $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$. Supposons que, pour tout $p \in \mathrm{dom}(v_{\ell(\imath)})$, p est linéaire et que $\mathrm{dom}(v_{\ell(\imath)})$ est un ensemble fini. On suppose que $r_\imath$ est affine. Si $q \circ T_\imath$ est linéaire et $v^{\mathbf{s}}_{\ell(\imath)} \cap r_\imath^{-1}(\mathbb{R}_-) \neq \emptyset$ alors pour tout $(\mathrm{dom}(v_{\ell(\imath)}), F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$:

$$\left[F^\sharp_\imath(v)\right](q) = \left[F_\imath^{\mathrm{dom}(v_{\ell(\imath)}),F,F^*}(v)\right](q) \ .$$

Exemple 5.23 (Retour à l'exemple 5.15)
L'ensemble Q de l'exemple 5.15 est un ensemble fini et pour tout $q \in Q$, $x \mapsto q \cdot x$ définit une forme linéaire. De plus, on a choisi une fonction telle que $\mathrm{dom}(v) = Q$ et comme dans

le calcul de la $\mathbb{P}$-enveloppe convexe, T_a correspond à l'identité et que pour tout $p \in \mathbb{P}$, p est linéaire alors d'après le théorème 5.2, la fonction $\text{vex}_{\mathbb{P}}(v)^{Q,E,E^*}$ coïncide avec la $\mathbb{P}$-enveloppe convexe de v.

La finitude des ensembles $\text{dom}(v_{\ell(a)})$ et $\text{dom}(v_{\ell(\imath)}) \neq \emptyset$ ainsi que la linéarité des éléments de ces ensembles entraîne la finitude de l'ensemble des solutions des problèmes duaux. Nous énonçons ce résultat sous la forme de la proposition suivante :

Proposition 5.5 (*Configurations linéaires finies et vecteurs duaux*)

Sous les hypothèses du théorème 5.2, l'ensemble des solutions des problèmes ($DAFq$) et ($DITq$) sont finis.

Le théorème 5.2 découle du théorème de dualité forte de la programmation linéaire. La proposition 5.5 découle du théorème de Krein-Milman et du théorème de séparation qui affirme d'une part qu'un ensemble convexe compact est l'enveloppe convexe de ses points extrêmes et d'autre part qu'une forme linéaire atteint son maximum sur un convexe compact sur un point extrême. Dans le cas des polytopes convexes, l'ensemble des points extrêmes est fini d'où le résultat. Le théorème 5.2 et la proposition 5.5 sont à la base de l'itération sur les politiques dans le domaine des gabarits linéaires de Sankaranarayanan développée par Taly et al [GGTZ07].

5.4 Propriétés de la sémantique relâchée des affectations et des intersections

L'interversion du supremum et de l'infimum dans la proposition 5.1 implique des propriétés supplémentaires à la fonction sémantique relâchée. Dans un premier temps, on peut réécrire la fonction sémantique relâchée comme un infimum de fonctions affines sur les degrés de liberté v.

5.4.1 Sur-approximation sûre de la sémantique abstraite

L'évaluation de la fonction sémantique abstraite par dualité est un bon outil puisque la fonction sémantique relâchée majore, au sens fonctionnel, la fonction sémantique abstraite. Ce théorème correspond au très connu théorème de dualité faible en optimisation. En interprétation abstraite, on parle de sur-approximation sûre de la sémantique abstraite.

Théorème 5.3 (*Sur-approximation sûre et fonction sémantique relâchée*)

Soient $a \in \mathbb{A}$ et $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\text{dom}(v_{\ell(a)}) \neq \emptyset$. Soient $\imath \in \mathbb{I}$ et $w \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $w_{\ell(\imath)} > -\infty$ et $\text{dom}(w_{\ell(\imath)}) \neq \emptyset$. Pour tout $(Q, E, E^*) \in \mathbf{D}(v_{\ell(a)})$ et pour tout $(Q', F, F^*) \in \mathbf{D}(w_{\ell(\imath)})$ alors :

$$F_a^\sharp(v) \leq F_a^{Q,E,E^*}(v) \leq \sup_{x \in \mathbb{R}^d} \mathbf{e}_{Tx}$$

$$\text{et} \quad F_\imath^\sharp(w) \leq F_\imath^{Q',F,F^*}(w) \leq \inf_{\mu \in \mathbb{R}_+} \sup_{x \in \mathbb{R}^d} \mathbf{e}_{Tx} - \mu r(x)$$

Malgré le côté classique, nous donnons une démonstration de ce résultat simple mais important.

Démonstration
Nous montrons la proposition pour les intersections, la démonstration pour les affectations est exactement la même. Soit $\imath \in \mathbb{I}$ et $w \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $w_{\ell(\imath)} > -\infty$ et $\mathrm{dom}(w_{\ell(\imath)}) \neq \emptyset$. Soit $(Q', F, F^*) \in \mathbf{D}(w_{\ell(\imath)})$. La deuxième inégalité vient uniquement du fait que :

$$F_i^{Q',F,F^*}(w) \leq \inf_{\mu\in\mathbb{R}_+} \sup_{x\in\mathbb{R}^d} \mathbf{e}_{Tx} + \langle w_{\ell(\imath)} - \mathbf{e}_x, 0\rangle_{Q'\times F\times F^*} - \mu r(x) \ .$$

Le vecteur 0 étant le vecteur nul de $\mathbb{R}^{Q'}$.

Montrons maintenant la première inégalité. Si $v^{\mathbf{s}}_{\ell(\imath)} \cap r^{-1}(\mathbb{R}_-) = \emptyset$, alors $F_i^\sharp(v) \equiv -\infty$ d'où l'inégalité. Maintenant, on suppose que $v^{\mathbf{s}}_{\ell(\imath)} \cap r^{-1}(\mathbb{R}_-) \neq \emptyset$. Soit $(Q', F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$. Soient $x \in v^{\mathbf{s}}_{\ell(\imath)} \cap r^{-1}(\mathbb{R}_-)$ et $(\lambda, \mu) \in F^*_+ \times \mathbb{R}_+$. On a, par définition de $F^*_+ \times \mathbb{R}_+$, $\mathbf{e}_{T(x)} \leq \mathbf{e}_{T(x)} + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{Q'\times F\times F^*} - \mu r(x)$ d'où :

$$F_i^\sharp(v) \leq \sup_{\substack{x\in v^{\mathbf{s}}_{\ell(\imath)}\\ r(x)\leq 0}} \mathbf{e}_{T(x)} + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{Q'\times F\times F^*} - \mu r(x) \leq \sup_{x\in\mathbb{R}^d} \mathbf{e}_{T(x)} + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{Q'\times F\times F^*} - \mu r(x)$$

Or $F_i^\sharp(v)$ ne dépend pas du couple (λ, μ), on conclut que :

$$F_i^\sharp(v) \leq \inf_{(\lambda,\mu)\in F^*_+\times\mathbb{R}_+} \sup_{x\in\mathbb{R}^d} \mathbf{e}_{T(x)} + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda\rangle_{Q'\times F\times F^*} - \mu r(x) \leq F_i^{Q',F,F^*}(v) \ .$$

■

Le théorème 5.3 permet également de conclure que, pour l'exemple 5.14 la fonction sémantique relâchée vaut la norme de $[F_2^{\mathbb{P},E,E^*}(v)](q) = \|q\|$ pour tout $q \in \mathbb{P}$. En effet, dans l'exemple 5.14, nous avons montré que $[F_2^{\mathbb{P},E,E^*}(v)](q) \leq \|q\|$, de plus, nous avions montré précédemment que, dans l'exemple 5.4, $[F_2^\sharp(v)](q) = \|q\|$ pour tout $q \in \mathbb{P}$.

Le théorème 5.3 peut s'appliquer au calcul de $\mathrm{vex}_{\mathbb{P}}(v)^{Q,F,F^*}$.

Proposition 5.6 (*Approximation sûre de l'enveloppe $\mathbb{P}$-convexe*)

Soient $v \in \overline{\mathbb{R}}^{\mathbb{P}}$ tel que $v > -\infty$ et $\mathrm{dom}(v) \neq \emptyset$. Soit $(\mathrm{dom}(v), F, F^*) \in \mathbf{D}(v)$ alors :

$$\mathrm{vex}_{\mathbb{P}}(v) \leq \mathrm{vex}_{\mathbb{P}}(v)^{Q,F,F^*} \ .$$

Soit l'ensemble $\mathbf{B}(v) = \{f_p \in \mathbb{R}^{\mathrm{dom}(w)},\ p \in \mathrm{dom}(v) \mid f_p(q) = 1 \text{ si } q = p,\ 0 \text{ si } q \neq p\}$. S'il existe une application i de $\mathbf{B}(v)$ dans F^*_+ telle que, pour tout $g \in \mathbf{B}(v)$, pour tout $x \in \mathbb{R}^d$:

$$\sum_{q\in\mathrm{dom}(v)} (v(q) - q(x))g(q) = \langle v - \mathbf{e}_x, i(g)\rangle_{\mathrm{dom}(v)\times F\times F^*} \tag{5.7}$$

alors :

$$\mathrm{vex}_{\mathbb{P}}(v)^{Q,F,F^*} \leq v \ .$$

Démonstration
L'inégalité $\text{vex}_{\mathbb{P}}(v) \leq \text{vex}_{\mathbb{P}}(v)^{\text{dom}(v),F,F^*}$ découle directement du théorème 5.3. Pour prouver l'inégalité $\text{vex}_{\mathbb{P}}(v)^{\text{dom}(v),F,F^*} \leq v$ il faut montrer que pour tout $p \in \mathbb{P}$, $\text{vex}_{\mathbb{P}}(v)^{\text{dom}(v),F,F^*}(p) \leq v(p)$. Remarquons, tout d'abord, que si $p \notin \text{dom}(v)$, l'inégalité est vraie. Soient $p \in \text{dom}(v)$, $x \in \mathbb{R}^d$. Considérons $f_p \in \mathbf{B}(v)$ et i telle que (5.7) soit vraie alors :

$$\text{vex}_{\mathbb{P}}(v)^{\text{dom}(v),F,F^*} \leq \sup_{x\in\mathbb{R}^d} p(x) + \langle v - \mathbf{e}_x, i(f_p)\rangle_{\text{dom}(v)\times F\times F^*} = \sup_{x\in\mathbb{R}^d} p(x) + v(p) - p(x) = v(p)$$

c'est-à-dire l'inégalité souhaitée. ∎

Par la suite, on notera pour un élement v de $\overline{\mathbb{R}}^{\mathbb{P}}$ qui ne prend pas la valeur $-\infty$ et dont le domaine n'est pas vide, $\mathbf{B}(v) = \{f_p \in \mathbb{R}^{\text{dom}(w)},\ p \in \text{dom}(v) \mid f_p(q) = 1 \text{ si } q = p,\ 0 \text{ si } q \neq p\}$. On notera, de plus, $F^*_+ >_i \mathbf{B}(v)$dès qu'il existe une application i de $\mathbf{B}(v)$ dans F^*_+ telle que, pour tout $g \in \mathbf{B}(v)$, pour tout $x \in \mathbb{R}^d$, (5.7) soit vraie.

5.4.2 Croissance de la sémantique relâchée

La fonction sémantique relâchée des unions est croissante en tant que supremum d'applications croissantes :

Proposition 5.7 (*Croissance de la sémantique relâchée des unions*)

La fonction $v \mapsto F^{\vee}_u$ est croissante sur $(\overline{\mathbb{R}}^{\mathbb{P}})^n$.

Démonstration
Soit $v, w \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tels que $v \geq w$. Soit $u \in \mathbb{U}$, on a $v_{\ell_g(u)} \geq w_{\ell_g(u)}$ et $v_{\ell_d(u)} \geq w_{\ell_d(u)}$ et on conclut que $\sup(v_{\ell_g(u)}, v_{\ell_d(u)}) \geq \sup(w_{\ell_g(u)}, w_{\ell_d(u)})$. ∎

Nous allons établir un premier résultat de croissance pour la fonction sémantique relâchée dans des cas très particulier. Ce résultat découle uniquement du théorème 5.3.

Corollaire 5.2 (*Cas particulier de croissance de la fonction sémantique relâchée*)

Soient $j \in \mathbb{A}\cup\mathbb{I}$, $v, w \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tels que $v_{\ell(j)} \geq w_{\ell(j)}$. S'il existe $p \in \mathbb{P}$ tel que $w_{\ell(j)}(p) = -\infty$ alors pour tout $(Q_w, F, F^*) \in \mathbb{D}(\mathbb{P})$

$$F_j^{Q_v,E,E^*}(v) \geq F_j^{Q_w,F,F^*}(w)$$

quelque soit $(Q_v, E, E^*) \in \mathbb{D}(\mathbb{P})$ s'il existe $p \in \mathbb{P}$, $v_{\ell(j)}(p) = -\infty$ ou si $v_{\ell(j)} \equiv +\infty$ et quelque soit $(Q_v, E, E^*) \in \mathbf{D}(v_{\ell(j)})$ si $v_{\ell(j)} > -\infty$ et si $\text{dom}(v_{\ell(j)}) \neq \emptyset$.

Si $v_{\ell(j)} \equiv +\infty$ alors pour tout $(Q_v, F, F^*) \in \mathbb{D}(\mathbb{P})$:

$$F_j^{Q_v,E,E^*}(v) \geq F_j^{Q_w,F,F^*}(w)$$

quelque soit $(Q_w, E, E^*) \in \mathbb{D}(\mathbb{P})$ si $w_{\ell(j)} \equiv +\infty$ et quelque soit $(Q_w, E, E^*) \in \mathbf{D}(w_{\ell(j)})$ si $w_{\ell(j)} > -\infty$ et si $\text{dom}(w_{\ell(j)}) \neq \emptyset$.

Nous allons nous intéresser à des propriétés de croissance. On va montrer que sous certaines conditions, $v_{\ell(a)} \geq w_{\ell(a)}$ implique que $F_a^{Q,E,E^*}(v) \geq F_a^{Q'F,F^*}(w)$ pour $a \in \mathbb{A}$, $(Q, E, E^*) \in \mathbf{D}(v_{\ell(a)})$ et $(Q', F, F^*) \in \mathbf{D}(w_{\ell(a)})$. On va montrer des résultats similaires pour $\imath \in \mathbb{I}$. La conservation de l'ordre découle d'une propriété sur les ensembles duaux. Nous introduisons une définition qui nous permettra d'établir des résultats de croissance de la fonction sémantique relâchée.

Définition 5.7 (*Hypothèse de croissance*)

Soient deux triplets $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$ et $(Q', F, F^*) \in \mathbb{D}(\mathbb{P})$. On dit que (Q', F, F^*) satisfait l'hypothèse de croissance par rapport à un triplet $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$ s'il existe une application $i : E^*_+ \mapsto F^*_+$ telle que, pour tout w tels que $(Q, E, E^*) \in \mathbf{D}(w)$ et $(Q, F, F^*) \in \mathbf{D}(w)$, pour tout $\lambda \in E^*_+$:

$$\langle w, \lambda \rangle_{Q \times E \times E^*} = \langle w, i(\lambda) \rangle_{Q' \times F \times F^*} \ .$$

On note dans ce cas $(Q, F, F^*) \ggg_i (Q, E, E^*)$.

Exemple 5.24 (Hypothèse de croissance dans le cas fini)
Soient Q, Q' deux parties finies non vides de $\mathbb{P}$. Supposons que $Q \subsetneq Q'$ et considérons $E = \mathbb{R}^Q$ et $F = \mathbb{R}^{Q'}$ et on choisit $E^* = \mathbb{R}^Q$ et $F = \mathbb{R}^{Q'}$. Les ensembles E^*_+ et F^*_+ correspondent respectivement aux ensembles $\mathbb{R}^Q_+$ et $\mathbb{R}^{Q'}_+$. En prenant l'application $i : E^*_+ \mapsto F^*_+$ définie, pour $\lambda \in E^*_+$ et $q \in Q'$ par :

$$[i(\lambda)](q) = \begin{cases} \lambda(q) & \text{si } q \in Q \\ 0 & \text{si } q \notin Q \end{cases}$$

On obtient finalement, pour $w \in \mathbb{R}^{Q'}$ (et donc $w_{|Q} \in \mathbb{R}^Q$) et pour $\lambda \in E^*_+$:

$$\sum_{q \in Q} w(q)\lambda(q) = \sum_{q \in Q} w(q)\lambda(q) + \sum_{q \in Q' \setminus Q} w(q) \times 0 = \sum_{q \in Q'} w(q)[i(\lambda)](q)$$

Finalement, on conclut que $(Q', F, F^*) \ggg_i (Q, E, E^*)$.

Exemple 5.25 (Hypothèse de croissance dans le cas compact)
Suppsons que $\mathbb{P}$ est un compact métrique. Soient Q une partie finie de $\mathbb{P}$ et Q' une partie fermée telles que $Q \subsetneq Q'$. Considérons $E = \mathbb{R}^Q$ et $F = \mathcal{C}(Q', \mathbb{R})$, on choisit $E^* = \mathbb{R}^Q$ et F^* l'ensemble des mesures signées finies sur Q'. Les ensembles E^*_+ et F^*_+ correspondent respectivement aux ensembles $\mathbb{R}^Q_+$ et à l'ensemble des mesures positives finies sur Q'. A $\gamma \in E^*_+$, on associe, $i(\gamma)$, la mesure définie par $\sum_{q \in Q} \gamma(q)\delta_q$ où δ_q représente la masse de Dirac en q. Ainsi pour $w \in \mathcal{C}(Q', \mathbb{R})$ (et donc $w_{|Q} \in \mathbb{R}^Q$) et pour tout $\gamma \in E^*_+$:

$$\sum_{q \in Q} w(q)\gamma(q) = \sum_{p \in Q} w(p)\gamma(p)\delta_q(p) = \int_{q \in Q'} w(q) d[i(\gamma)](q)$$

Finalement, on conclut que $(Q', F, F^*) \ggg_i (Q, E, E^*)$.

Proposition 5.8 (*Croissance de la fonction sémantique relâchée*)

Soient $j \in \mathbb{A} \cup \mathbb{I}$, $v, w \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tels que $v_{\ell(j)} \geq w_{\ell(j)}$, $\mathrm{dom}(v_{\ell(j)}) \neq \emptyset$ et $w_{\ell(j)} > -\infty$. Soient $(Q_v, E, E^*) \in \mathbf{D}(v_{\ell(j)})$ et $(Q_w, F, F^*) \in \mathbf{D}(w_{\ell(j)})$. Si $(Q_v, E, E^*) \in \mathbf{D}(w_{\ell(j)})$ et il existe une application i telle que $(Q_w, F, F^*) \ggg_i (Q_v, E, E^*)$ alors :

$$F_j^{Q_v, E, E^*}(v) \geq F_j^{Q_w, F, F^*}(w)$$

Démonstration
Soient $\imath \in \mathbb{I}$, $v, w \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$. Supposons que $v_{\ell(j)} \geq w_{\ell(j)}$, $\mathrm{dom}(v_{\ell(j)}) \neq \emptyset$ et $w_{\ell(j)} > -\infty$. Soient $(Q_v, E, E^*) \in \mathbf{D}(v_{\ell(j)})$ et $(Q_w, F, F^*) \in \mathbf{D}(w_{\ell(j)})$. Comme $(Q_v, E, E^*) \in \mathbf{D}(w_{\ell(j)})$ et par définition de E^*_+, on a, pour tout $\lambda \in E^*$, $\langle v_{\ell(\imath)}, \lambda \rangle_{Q_v \times E \times E^*} \geq \langle w_{\ell(\imath)}, \lambda \rangle_{Q_v \times E \times E^*}$ d'où :

$$\begin{array}{rl} & \inf\limits_{\lambda \in E^*_+} \inf\limits_{\mu \in \mathbb{R}_+} \sup\limits_{x \in \mathbb{R}^d} \mathbf{e}_{Tx} + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda \rangle_{Q_v \times E \times E^*} - \mu r(x) \\ \geq & \inf\limits_{\lambda \in E^*_+} \inf\limits_{\mu \in \mathbb{R}_+} \sup\limits_{x \in \mathbb{R}^d} \mathbf{e}_{Tx} + \langle w_{\ell(\imath)} - \mathbf{e}_x, \lambda \rangle_{Q_v \times E \times E^*} - \mu r(x) \end{array}$$

Maintenant par hypothèse de croissance alors il existe $i : E^*_+ \mapsto F^*_+$ telle que :

$$\begin{array}{rl} & \inf\limits_{\lambda \in E^*_+} \inf\limits_{\mu \in \mathbb{R}_+} \sup\limits_{x \in \mathbb{R}^d} \mathbf{e}_{Tx} + \langle w_{\ell(\imath)} - \mathbf{e}_x, \lambda \rangle_{Q_v \times E \times E^*} - \mu r(x) \\ = & \inf\limits_{\lambda \in E^*_+} \inf\limits_{\mu \in \mathbb{R}_+} \sup\limits_{x \in \mathbb{R}^d} \mathbf{e}_{Tx} + \langle w_{\ell(\imath)} - \mathbf{e}_x, i(\lambda) \rangle_{Q_w \times F \times F^*} - \mu r(x) \\ = & \inf\limits_{\substack{\lambda' \in F^*_+ \\ \lambda' = i(\lambda) \\ \lambda \in E^*_+}} \inf\limits_{\mu \in \mathbb{R}_+} \sup\limits_{x \in \mathbb{R}^d} \mathbf{e}_{Tx} + \langle w_{\ell(\imath)} - \mathbf{e}_x, \lambda' \rangle_{Q_w \times F \times F^*} - \mu r(x) \\ \geq & \inf\limits_{\lambda' \in F^*_+} \inf\limits_{\mu \in \mathbb{R}_+} \sup\limits_{x \in \mathbb{R}^d} \mathbf{e}_{Tx} + \langle w_{\ell(\imath)} - \mathbf{e}_x, \lambda' \rangle_{Q_w \times F \times F^*} - \mu r(x) \\ = & F_j^{Q_w, F, F^*}(w) \end{array}$$

d'où l'inégalité. ∎

5.4.3 Concavité des fonctions sémantiques relâchées

Nous terminons cette section sur les propriétés sur la fonction sémantique relâchée. Nous montrons que si on considère uniquement $(\mathbb{R}^{\mathbb{P}})^n$, les fonctions sémantiques relâchées possèdent de bonnes propriétés de convexité.

Proposition 5.9 (*Convexité de la fonction sémantique relâchée des unions*)
Pour tout $u \in \mathbb{U}$, pour tout $p \in \mathbb{P}$, l'application de $(\mathbb{R}^{\mathbb{P}})^n$ dans $\mathbb{R}^{\mathbb{P}}$, $v \mapsto \left[F_u^{\vee}(v)\right](p)$ est convexe en tant que supremum d'applications linéaires.

La fonction sémantique relâchée pour les affectations $v \mapsto F_a^{Q,E,E^*}(v)$ et les intersections $v \mapsto F_i^{Q',F,F^*}(v)$ est l'infimum de fonctions affines sur l'ensemble des éléments $v \in (\mathbb{R}^{\mathbb{P}})^n$ pour lesquels $(Q, E, E) \in \mathbf{D}(v_{\ell(a)})$ et $(Q', F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$. Nous énonçons ce résultat sous la forme du lemme 5.1.

Lemme 5.1 (*Reformulation*)

La fonction sémantique relâchée admet une deuxième formulation :

– Soient $a \in \mathbb{A}$, $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$. Soit $(Q, E, E^*) \in \mathbf{D}(v_{\ell(a)})$, alors :

$$F_a^{Q,E,E^*}(v) = \inf_{\lambda \in E_+^*} \langle v_{\ell(a)}, \lambda \rangle_{Q \times E \times E^*} + \mathbb{V}_a^{Q,E,E^*}(\lambda) \ .$$

où $\mathbb{V}_a^{Q,E,E^*}(\lambda) = \sup_{x \in \mathbb{R}^d} \mathbf{e}_{T(x)} - \langle \mathbf{e}_x, \lambda \rangle_{Q \times E \times E^*}$.

– Soient $\imath \in \mathbb{I}$, $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(\imath)} > -\infty$ et $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$. Soit $(Q', F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$, alors :

$$F_\imath^{Q',F,F^*}(v) = \inf_{(\lambda,\mu) \in F_+^* \times \mathbb{R}_+} \langle v_{\ell(\imath)}, \lambda \rangle_{Q' \times F \times F^*} + \mathbb{V}_\imath^{Q',F,F^*}(\lambda, \mu) \ .$$

où $\mathbb{V}_\imath^{Q',F,F^*}(\lambda, \mu) = \sup_{x \in \mathbb{R}^d} \mathbf{e}_{T(x)} - \langle \mathbf{e}_x, \lambda \rangle_{Q' \times F \times F^*} - \mu r(x)$.

Si $v_{\ell(a)}$ (resp. $v_{\ell(\imath)}$) est identiquement égale à $+\infty$ alors, on peut poser pour tout $(Q, E, E^*) \in \mathbb{D}(\mathbb{P})$, $F_a^{Q,E,E^*}(v) = \mathbb{V}_a^{Q,E,E^*} = \sup_{x \in \mathbb{R}^d} \mathbf{e}_{T_a(x)}$ (resp. $F_\imath^{Q',F,F^*}(v) = \mathbb{V}_\imath^{Q',F,F^*} = \inf_{\mu \in \mathbb{R}_+} \sup_{x \in \mathbb{R}^d} \mathbf{e}_{T(x)} - \mu r(x)$). Dans le cas où $v_{ell(a)}$ prend la valeur $-\infty$, on pose $F_a^{Q,E,E^*}(v) = \mathbb{V}_a^{Q,E,E^*} \equiv -\infty$ et de même pour les intersections.

Le lemme 5.1 implique comme corollaire que les fonctions sémantiques relâchées $v \mapsto F_a^{Q,E,E^*}(v)$ et $v \mapsto F_\imath^{Q',F,F^*}(v)$ sont des fonctions concaves en $v \in (\mathbb{R}^{\mathbb{P}})^n$ pour lesquels $(Q, E, E^*) \in \mathbf{D}(v_{\ell(a)})$ et $(Q', F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$.

5.4.4 Propriété de sélection

La propriété de sélection consiste à pouvoir choisir une politique à chaque itération de l'algorithme d'itération sur les politiques. Dans notre cas, comme les politiques sont données par des multiplicateurs de Lagrange, il suffit de prouver l'existence de solutions pour le problème dual pour prouver qu'il y a propriété de sélection. Cependant, le fait que les solutions primales existe ne suffit pas à prouver que les solutions duales existent, il faut ajouter une hypothèse en plus : la condition de qualification de Slater.

La propriété de sélection est, dans le cadre du domaine des sous-niveaux, assurée par un théorème d'atteinte duale, c'est-à-dire que, pour tout $a \in \mathbb{A}$, pour tout $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$, pour tout $(Q, E, E^*) \in \mathbf{D}(v_{\ell(a)})$, pour tout $p \in \mathbb{P}$, il existe $\lambda^p \in E_+^*$ tel que :

$$\left[F_a^{Q,E,E^*}(v)\right](p) = \langle v_{\ell(a)}, \lambda^p \rangle_{Q \times E \times E^*} + \left[\mathbb{V}_a^{Q,E,E^*}(\lambda^p)\right](p)$$

et, pour tout $\imath \in \mathbb{I}$, pour tout $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(\imath)} > -\infty$ et $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$, pour tout $(Q', F, F^*) \in \mathbf{D}(v_{\ell(\imath)})$, pour tout $p \in \mathbb{P}$, il existe $(\lambda^p, \mu^p) \in F_+^* \times \mathbb{R}_+$ tel que :

$$\left[F_\imath^{Q',F,F^*}(v)\right](p) = \langle v_{\ell(\imath)}, \lambda^p \rangle_{\mathrm{dom}(v_{\ell(\imath)})} + \left[\mathbb{V}_\imath^{Q',F,F^*}(\lambda^p, \mu^p)\right](p)$$

Nous utilisons ici la condition de Slater dans le cas où Q est une sous-partie non vide compacte de $\mathbb{P}$ muni d'une distance et $v_{\ell(a)}$ et $v_{\ell(\imath)}$ sont des fonctions continues sur Q. Si Q

est une partie finie, on munit $\mathbb{R}^Q$ de la dualité finie. Pour prouver l'existence de solutions duales, nous allons nous servir du résultat 5.4 qui assure l'existence d'un minimiseur pour des fonctions définies sur le dual topologique d'un espace de Banach.

Théorème 5.4 (*Existence de solutions*)

Soit H un espace de Banach. On suppose que H', son dual topologique, est un espace de Banach pour une certaine norme $\|\cdot\|_{H'}$. Soit $f : H' \mapsto]-\infty, +\infty]$ une fonction telle que :
- Il existe $x \in H'$, $f(x) \in \mathbb{R}$;
- f est semi-continue inférieurement pour la topologie faible-$*$;
- f est coercive pour la norme $\|\cdot\|_{H'}$.

Alors f atteint son minimum sur H'.

Nous énonçons la condition de Slater dans le cas Q compact. Dans notre cadre, puisqu'on s'intéresse à $\mathcal{C}(Q, \mathbb{R})$, le dual topologique que l'on considère est l'espace vectoriel des mesures signées finies pour la tribu borélienne sur Q. On munit l'espace des mesures signées finies de la norme de la variation totale $\|\cdot\|_{VT}$ qui est un espace de Banach pour cette norme. De plus, pour toute mesure positive μ, on a $\|\mu\|_{VT} = \int_Q d\mu$.

Définition 5.8 (*Condition de Slater*)

Le problème 5.2 satisfait la condition de Slater, s'il existe $x \in \mathbb{R}^d$ tel que $(v_{\ell(a)} - \mathbf{e}_x)(p) > 0$ pour tout $p \in Q$.

Le problème 5.4 satisfait la condition de Slater, s'il existe $x \in \mathbb{R}^d$ tel que $(v_{\ell(\imath)} - \mathbf{e}_x)(p) > 0$ pour tout $p \in Q'$ et $-r(x) > 0$.

Dès qu'il y a compacité, on peut trouver choisir une constante ε strictement positive qui minore chaque $(v_{\ell(a)} - \mathbf{e}_x)(p)$ uniformément sur Q dès que $(v_{\ell(a)} - \mathbf{e}_x)(p) > 0$ pour tout $p \in Q$. Ce résultat reste vrai avec $v_{\ell(\imath)}$ et $-r$.

Proposition 5.10 (*Propriété de Slater dans le cas compact*)

On suppose que Q est un espace topologique compact et $w \in \mathcal{C}(Q, \mathbb{R})$. Soit $x \in \mathbb{R}^d$ tel que $(w - \mathbf{e}_x)(p) > 0$, pour tout $p \in Q$ et supposons que $\mathbf{e}_x \in \mathcal{C}(Q, \mathbb{R})$. Alors, il existe $\varepsilon > 0$ et tel que $(w - \mathbf{e}_x)(p) > \varepsilon$ pour tout $p \in Q$.

Démonstration (proposition 5.10)
. Soit $x \in \mathbb{R}^d$ tel que $w - \mathbf{e}_x > 0$, on a donc :

$$Q = (w - \mathbf{e}_x)^{-1}(]0, +\infty[) \subseteq \bigcup_{\varepsilon > 0} \{p \in Q \mid (w - \mathbf{e}_x)^{-1}(]\varepsilon, +\infty[)\} \ .$$

Par hypothèse de continuité de $w - \mathbf{e}_x$, la famille $\left\{(w - \mathbf{e}_x)^{-1}(]\varepsilon, +\infty[)\right\}_{\varepsilon > 0}$ forme un recouvrement d'ouverts de Q. Par compacité de Q,

$$Q \subseteq \bigcup_{\substack{\varepsilon_i > 0 \\ i=1,\dots,n}} (w - \mathbf{e}_x)^{-1}(]-\varepsilon_i, +\infty[) \ .$$

En posant $\varepsilon = \min_{i=1,\dots,n} \varepsilon_i$, on conclut qu'il existe $\varepsilon > 0$ tel que $w - \mathbf{e}_x > \varepsilon$. ∎

La condition de qualification de Slater implique que les fonctions duales ψ_a et $\psi_\imath$ des problèmes respectifs 5.5 et 5.6 sont coercives c'est-à-dire qu'elles vérifient :

$$\lim_{\|\lambda^q\|\to+\infty} [\psi_a(q)](\lambda^q) = \lim_{\max(\|\lambda^q\|,|\mu^q|)\to+\infty} [\psi_\iota(q)](\lambda^q,\mu^q) = +\infty$$

De plus, la coercivité des fonctions est équivalente au fait que les sous-niveaux de la fonction duale sont bornés. On peut énoncer les résultats suivants :

Théorème 5.5 (*Propriété de sélection*)

Supposons que le problème 5.2 satisfait la condition de Slater et que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$. Soit $(Q, E, E') \in \mathbf{D}(v_{\ell(a)})$ tel que E soit un espace de Banach et soit E' le dual topologique de E. S'il existe $\lambda^q \in E'_+$ tel que $[\psi_a(q)](\lambda^q) < +\infty$ alors il existe $\lambda_0^q \in E'_+$:

$$\left[F_a^{Q,E,E'}(v)\right](q) = \sup_{x\in\mathbb{R}^d} q(Tx) + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda_0^q\rangle_{Q\times E\times E'}$$

Supposons que le problème 5.4 satisfait la condition de Slater et que $v_{\ell(\iota)} > -\infty$ et $\mathrm{dom}(v_{\ell(\iota)}) \neq \emptyset$. Soit $(Q', F, F') \in \mathbf{D}(v_{\ell(\iota)})$ tel que F soit un espace de Banach et F' le dual topologique de F. S'il existe $(\lambda^q, \mu^q) \in F'_+ \times \mathbb{R}_+$, tel que $[\psi_\iota(q)](\lambda^q,\mu^q) < +\infty$ alors il existe $(\lambda_0^q, \mu_0^q) \in F'_+ \times \mathbb{R}_+$:

$$\left[F_\iota^{Q',F,F'}(v)\right](q) = \sup_{x\in\mathbb{R}^d} q(Tx) + \langle v_{\ell(\iota)} - \mathbf{e}_x, \lambda_0^q\rangle_{Q'\times F\times F'} - \mu_0^q r(x)$$

Proposition 5.11 (*Compacité des politiques*)

On reprend les hypothèses du théorème 5.5.

S'il existe $\lambda^q \in F'_+$ tel que $[\psi_a(q)](\lambda^q) < +\infty$ alors l'ensemble $\mathrm{Sol}(DAFq)$ est un ensemble compact pour la topologie faible-$*$ et non vide.

De même, s'il existe $\lambda^q \in F'_+$ et $\mu^q \in \mathbb{R}_+$ tels que $[\psi_\iota(q)](\lambda^q,\mu^q) < +\infty$ alors l'ensemble $\mathrm{Sol}(DITq)$ est un ensemble compact pour la topologie faible-$*$ et non vide.

La coercivité est simple à vérifier en calculant des limites mais la propriété est en réalité utilisé pour prouver que l'ensemble des vecteurs duaux réalisables est un ensemble borné (pour la topologie forte). En effet, une fonction f d'un espace de Banach sur $\mathbb{R}$ est coercive si et seulement si pour tout $\alpha \in \mathbb{R}$, le sous-niveau $S_\alpha(f)$ est borné. Nous allons vérifier que les fonctions duales $\psi_a(\lambda^q, q)$ et $\psi_\iota(\lambda^q, \mu^q, q)$ satisfont les hypothèses du théorème 5.4 et le théorème 5.5 sera démontré.

Démonstration (Théorème 5.5)

Soit $a \in \mathbb{A}$. Pour tout $x \in \mathbb{R}^d$, la fonction $\lambda \mapsto p\circ T(x) + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda\rangle_{Q\times E\times E^*}$ est affine sur E^* donc convexe. De plus, par définition de la topologie faible-$*$, la fonction $\lambda \mapsto p\circ T(x) + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda\rangle_{Q\times E\times E^*}$ est continue pour la topologie faible-$*$ donc semi-continue inférieurement pour cette topologie. On conclut, par la propriété 4.1, que ψ_a est convexe et semi-continue inférieurement pour la topologie faible-$*$ en tant que supremum de fonctions convexe semi-continue inférieurement.

Soient $x \in \mathbb{R}^d$ tel que $v_{\ell(a)} - \mathbf{e}_x > \varepsilon$ et soit $\lambda \in E^*_+$,

$$\langle v_{\ell(a)} - \mathbf{e}_x, \lambda\rangle_{Q\times E\times E^*} = \int_Q v_{\ell(a)}(p) - p(x) d\lambda(p) \geq \varepsilon \int_Q d\lambda(p) = \varepsilon\|\lambda\|_{VT}$$

d'où $\varepsilon\|\lambda\|_{VT}$ tend vers $+\infty$ quand $\|\lambda\|_{VT}$ tend vers $+\infty$.

Dans le cas Q fini, l'intégrale est une somme sur Q et la norme $\|\lambda\|$ est la norme 1. Puisque $\mathbb{R}^Q$ définit un espace vectoriel de dimension finie alors toutes les normes sont équivalentes et donc $\|\lambda\|$ tend vers $+\infty$ pour la norme 1 implique $N(\lambda)$ pour toute autre norme N.

En conclusion, ψ_a est convexe, semi-continue inférieurement pour la topologie faible-$*$ et coercive pour la norme de E^*, on conclut l'existence de λ_0^q par le théorème 5.4. ∎

Démonstration (Proposition 5.11)
La fonction ψ_a est semi-continue inférieurement pour la topologie faible-$*$ et coercive pour la topologie forte. De plus, par hypothèse, par le théorème 5.5 Sol($DAFq$) est non vide.

En outre, Sol($DAFq$) $= S_{\psi(\lambda_0,v)}(\psi_a)$ pour tout $\lambda_0 \in$ Sol($DAFq$) donc Sol($DAFq$) est convexe fermé pour la topologie faible-$*$en tant que sous-niveau d'une fonction faible-$*$ s.c.i. convexe L'ensemble Sol($DAFq$) est borné en tant que sous-niveau de fonction coercive.

Par le théorème de Banach-Alaoglu, comme Sol($DAFq$) est borné et fermé pour la topologie faible-$*$ alors Sol($DAFq$) est faible-$*$ compact. ∎

La condition de qualification de Slater va être utilisée plus tard dans ce chapitre et dans le chapitre suivant puisque nous nous placerons dans le cas de configurations finies. Nous introduisons une nouvelle notation qui nous permettra de construire notre schéma d'itération. Soient $j \in \mathbb{A} \cup \mathbb{I}$ et un élément $v_{\ell(j)}$ tel que $v_{\ell(j)} > -\infty$ et $\mathrm{dom}(v_{\ell(j)}) \neq \emptyset$. Soit $(Q_j, E_j, E_j^*) \in \mathbf{D}(v_{\ell(j)})$ et soit $q \in \mathbb{P}$, on note Sol$(Q_j, E_j, E_j^*, v_{\ell(j)}, q)$ l'ensemble des vecteurs duaux $\lambda^q \in (E_j)_+^*$ tel que :

$$\left[F_j^{Q_j,E_j,E_j^*}(v)\right](q) = \langle v_{\ell(j)}, \lambda^p\rangle_{Q_j\times E_j\times E_j^*} + \left[\mathbb{V}_a^{Q_j,E_j,E_j^*}(\lambda^p)\right](p)$$

si $j \in \mathbb{A}$ et, l'ensemble des couples $(\lambda^q, \mu^q) \in (E_j)_+^* \times \mathbb{R}_+$ tels que :

$$\left[F_i^{Q',F,F^*}(v)\right](p) = \langle v_{\ell(\imath)}, \lambda^p\rangle_{\mathrm{dom}(v_{\ell(\imath)})} + \left[\mathbb{V}_i^{Q',F,F^*}(\lambda^p, \mu^p)\right](p)$$

si $j \in \mathbb{I}$. Les ensembles Sol$(Q_j, E_j, E_j^*, v_{\ell(j)}, q)$ peuvent être éventuellement vides. Si $v_{\ell(j)} \equiv +\infty$, Sol$(Q_j, E_j, E_j^*, v_{\ell(j)}, q) = \{0\}$ pour tout $q \in \mathbb{P}$ si $j \in \mathbb{A}$ et Sol$(Q_j, E_j, E_j^*, v_{\ell(j)}, q) = \{0, \mu^q\}$ si $j \in \mathbb{A}$ et si μ^q satisfait :

$$\inf_{\mu\in\mathbb{R}_+}\sup_{x\in\mathbb{R}^d} q(x) - \mu r(x) = \sup_{x\in\mathbb{R}^d} q(x) - \mu^q r(x)$$

Enfin, s'il existe $p \in \mathbb{P}$ tel que $v_{\ell(a)}(p) = -\infty$, on pose Sol$(Q_j, E_j, E_j^*, v_{\ell(j)}, q) = (E_j)_+^*\backslash\{0\}$ si $j \in \mathbb{A}$ et $((E_j)_+^*\backslash\{0\}) \times \{0\}$ si $j \in \mathbb{I}$.

5.5 Schéma d'itération sur les politiques à dualité dynamique

Dans cette section, nous entamons une discussion sur les méthodes itératives pour trouver une solution, ou du moins une sur-approximation valide du problème :

Problème 5.7 (Plus petit invariant de la fonction sémantique abstraite)

$$\inf\{v \in (\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}}))^n \mid F^\sharp(v) \leq v\} \qquad \text{(INV)}$$

La fonction $F^\sharp$ désigne la fonction dont toutes les coordonnées sont $F_i^\sharp$. La solution de (INV) est le plus petit point fixe de $F^\sharp$ puisque cette fonction est croissante pour l'ordre produit et nous avons montré que $\mathrm{Vex}_{\mathbb{P}}(\overline{\mathbb{R}}^{\mathbb{P}})$ est un treillis complet. Cependant, nous ne savons pas évalué la fonction $F^\sharp$, nous utilisons donc les fonctions sémantiques relâchées pour trouver une sur-approximation valide de la solution du problème (INV). En effet, d'après le théorème 5.3, pour un élément $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ fixé et pour une coordonnée $j \in \mathbb{A} \cup \mathbb{I}$, $F_j^{Q_j,E_j,E_j^*}(v) \leq v_j$, implique que $F_j^\sharp(v) \leq v_j$ quelque soit $(Q_j, E_j, E_j^*) \in \mathbf{D}(v_{\ell(j)})$.

5.5.1 Définition des politiques

Nous allons définir une politique à partir d'une famille d'espaces vectoriels E_j donnés. Chaque E_j représente un sous-espace vectoriel d'un ensemble $\mathbb{R}^{Q_j}$ où Q_j est un sous-ensemble non vide de $\mathbb{P}$ pour une coordonnée $j \in \mathbb{A} \cup \mathbb{I}$. Nous supposons que pour tout $j \in \mathbb{A} \cup \mathbb{I}$, $\mathbf{e}_x \in E_j$ pour tout $x \in \mathbb{R}^d$. Nous nous intéressons également à l'ensemble :

$$\mathbf{C}(Q_j, E_j) = \{v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n \mid v_{\ell(j)} > -\infty,\ Q_j \subseteq \mathrm{dom}(v_{\ell(j)}),\ v_{\ell(j)_{|Q_j}} \in E_j\}\ .$$

De manière équivalente, l'ensemble $\mathbf{C}(Q_j, E_j)$ désigne tous les vecteurs $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tels que $(Q_j, E_j) \in D(v_{\ell(j)})$. On pose $\mathbf{C}(Q_j, E_j) = (\overline{\mathbb{R}}^{\mathbb{P}})^n$ si $j \notin \mathbb{A} \cup \mathbb{I}$. On complète l'ensemble $\mathbf{C}(Q_j, E_j)$ en ajoutant les vecteurs $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tels que $v_{\ell(j)}(p) = -\infty$ pour un certain $p \in \mathbb{P}$ ainsi que les vecteurs $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ dont la coordonnée $v_{\ell(j)}$ est identiquement égale à $+\infty$. On obtient ainsi l'ensemble $\overline{\mathbf{C}}(Q_j, E_j)$:

$$\overline{\mathbf{C}}(Q_j, E_j) = \mathbf{C}(Q_j, E_j) \cup \{v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n \mid \exists p \in \mathbb{P},\ v_{\ell(j)}(p) = -\infty\} \cup \{v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n \mid\ v_{\ell(j)} \equiv +\infty\}\ .$$

Par la suite, on notera $\overrightarrow{Q}$ une famille de sous-ensembles $(Q_j)_{j\in\mathbb{A}\cup\mathbb{I}}$ et $\overrightarrow{E}$ pour désigner une famille d'ensembles $(E_j)_{j\in\mathbb{A}\cup\mathbb{I}}$ telle que E_j est un sous-espace vectoriel de $\mathbb{R}^{Q_j}$. De plus, on note $\mathbf{C}(\overrightarrow{Q}, \overrightarrow{E})$ l'ensemble suivant :

$$\bigcap_{j\in\mathbb{A}\cup\mathbb{I}} \mathbf{C}(Q_j, E_j)\ .$$

De même, on définit $\overline{\mathbf{C}}(\overrightarrow{Q}, \overrightarrow{E})$ comme l'ensemble suivant :

$$\bigcap_{j\in\mathbb{A}\cup\mathbb{I}} \overline{\mathbf{C}}(Q_j, E_j)\ .$$

Conformément à la définition 3.4, on définit une politique comme une fonction qui collecte tous les vecteurs duaux disponibles à un certain point de contrôle. Nous allons procéder par étapes. On associe pour tout $j \in \mathbb{A} \cup \mathbb{I}$, à une famille (E_j) de sous-espaces vectoriels de $\mathbb{R}^{Q_j}$, une famille $\overrightarrow{E^*}$ d'espaces vectoriels E_j^* telle que chaque E_j^* est en dualité avec E_j et chaque couple (E_j, E_j^*) satisfait (H1).

Etape 1 : Soient $a \in \mathbb{A}$, $v \in \mathbf{C}(\overrightarrow{Q}, \overrightarrow{E})$ et $q \in \mathbb{P}$. On définit l'ensemble $\Lambda_a(v, q)$ par :

$$\{\lambda^q \in (E_a)_+^* \mid [\psi_a(q)]\,(\lambda^q) < +\infty\}$$

Pour rappel, la fonction $[\psi_a(q)](\lambda^q) = \sup_{x\in\mathbb{R}^d} q(T_a(x)) + \langle v_{\ell(a)} - \mathbf{e}_x, \lambda^q\rangle_{Q_a\times E_a\times E_a^*}$ et où T_a désigne l'affectation au point de contrôle a. Si $v_{\ell(a)} \equiv +\infty$, on pose $\Lambda_a(v,q) = \{0_{E_a^*}\}$ pour tout $q \in \mathbb{P}$. S'il existe $p \in \mathbb{P}$ tel que $v_{\ell(a)}(p) = -\infty$, on pose $\Lambda_a(v,q) = E_a^*$ pour tout $q \in \mathbb{P}$.
On définit également l'ensemble $\Lambda_\imath(v,q)$ pour $\imath \in \mathbb{I}$, $v \in \mathbf{C}(\overrightarrow{Q},\overrightarrow{E})$ et $q \in \mathbb{P}$ par :

$$\{(\lambda^q,\mu^q) \in (E_j)_+^* \times \mathbb{R}_+ \mid [\psi_\imath(q)](\lambda^q,\mu^q) < +\infty\}$$

Pour rappel, la fonction $[\psi_\imath(q)](\lambda^q,\mu^q) = \sup_{x\in\mathbb{R}^d} q(T_\imath(x)) + \langle v_{\ell(\imath)} - \mathbf{e}_x, \lambda^q\rangle_{Q_\imath\times E_\imath\times E_\imath^*} - \mu^q r_\imath$ et où $T_\imath$ désigne l'affectation au point de contrôle $\imath$. Si $v_{\ell(\imath)} \equiv +\infty$, on pose $\Lambda_\imath(v,q) = \{(0_{E_\imath^*},\mu^q) \mid [\psi_\imath(q)](0,\mu^q) < +\infty\}$ pour tout $q \in \mathbb{P}$. S'il existe $p \in \mathbb{P}$ tel que $v_{\ell(\imath)}(p) = -\infty$, on pose $\Lambda_\imath(v,q) = ((E_\imath^*)\backslash\{0\}) \times \{0\}$ pour tout $q \in \mathbb{P}$.
Ces deux ensembles peuvent être éventuellement vides et dans ce cas la fonction sémantique relâchée $F_a^{Q_a,E_a,E_a^*}(v)$ (resp.$F_\imath^{Q_\imath,E_\imath,E_\imath^*}(v)$) vaut identiquement $+\infty$.
Enfin, on pose pour $k \in \mathbb{U}$ et pour tout $v \in \overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E})$ et tout $p \in \mathbb{P}$, $\Lambda_k(v,q) = \{\vee\}$ et pour $k \notin \mathbb{U}\cup\mathbb{A}\cup\mathbb{I}$, $\Lambda_k(v,q) = \{\sharp\}$ ce qui signifie que la fonction sémantique relâchée est la fonction $F_u^\vee$ pour $u \in \mathbb{U}$ et $F_k{}^\sharp$ si $k \notin \mathbb{U}\cup\mathbb{A}\cup\mathbb{I}$.

Etape 2 : On note $\Sigma = \{1,\ldots,n\}$. Soient $k \in \Sigma$ et $v \in \overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E})$. On appelle fonction de sélection de $\mathbb{P}$ sur $\bigcup_{q\in\mathbb{P}} \Lambda_k(v,q)$, une fonction $\rho_{k,v}$ telle que :

$$\rho_{k,v}(p) \in \Lambda_k(v,p),\ \forall\, p \in \mathbb{P}\ .$$

On note $\Lambda_k(v)$ l'ensemble des fonctions de sélections $\rho_{v,k}$ de $\mathbb{P}$ sur $\bigcup_{q\in\mathbb{P}} \Lambda_k(v,q)$.

Etape 3 : Maintenant fixons $k \in \Sigma$, on définit de la même manière, les fonctions de sélections ρ_k de $\overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E})$ sur $\bigcup_{v\in\overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E})} \Lambda_k(v)$. Ces fonctions de sélection ρ_k satisfont :

$$\rho_k(w) \in \Lambda_k(w), \forall\, w \in \overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E})\ .$$

Pour $k \in \Sigma$, on note Λ_k l'ensemble des fonctions de sélection ρ_k de $\overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E})$ sur $\bigcup_{v\in\overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E})} \Lambda_k(v)$.

Etape 4 : Une fonction de sélection de Σ dans $\bigcup_{k\in\Sigma} \Lambda_k$ est appelée une *politique*. Une politique π est donc une fonction de Σ dans $\bigcup_{k\in\Sigma} \Lambda_k$ telle que :

$$\pi(i) := \pi_i \in \Lambda_i\ \forall\, i \in \Sigma\ .$$

En résumé, nous obtenons la définition suivante :

Définition 5.9 (*Politiques*)

Une politique est une fonction de sélection qui à chaque point de contrôle $i \in \{1, \dots, n\}$, à chaque fonction $v \in \overline{\mathbf{C}}(\overrightarrow{Q}, \overrightarrow{E})$ et chaque fonction de base $p \in \mathbb{P}$ associe son ensemble de vecteurs duaux disponibles. Plus formellement, une politique π est une fonction de Σ à valeurs dans :

$$\bigcup_{k\in\Sigma}\left(\overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E}) \mapsto \bigcup_{v\in\overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E})}\left(\mathbb{P}\mapsto\bigcup_{q\in\mathbb{P}}\Lambda_k(v,q)\right)\right)$$

telle que, pour tout $i \in \Sigma$, $\pi_i : \overline{\mathbf{C}}(\overrightarrow{Q}, \overrightarrow{E}) \mapsto \bigcup_{v\in\overline{\mathbf{C}}(\overrightarrow{Q},\overrightarrow{E})}\left(\mathbb{P}\mapsto\bigcup_{q\in\mathbb{P}}\Lambda_i(v,q)\right)$, pour tout $w \in \overline{\mathbf{C}}(\overrightarrow{Q}, \overrightarrow{E})$, $\pi_i(w) : \mathbb{P} \mapsto \bigcup_{q\in\mathbb{P}} \Lambda_i(w, q)$ et finalement, pour tout $p \in \mathbb{P}$, $[\pi_i(w)](p) \in \Lambda_i(w, p)$.

On note $\Pi(\overrightarrow{Q}, \overrightarrow{E}, \overrightarrow{E^*})$ l'ensemble des politiques.

Remarque 5.11 On notera $[\pi_i(w, p)]$ à la place de $[\pi_i(w)](p)$.

En conclusion, pour une famille $\overrightarrow{Q}$ de sous-ensembles $(Q_j)_{j\in\mathbb{A}\cup\mathbb{I}}$ non vide de $\mathbb{P}$ et une famille $\overrightarrow{E}$, de sous-espaces vectoriels $(E_j)_{j\in\mathbb{A}\cup\mathbb{I}}$ (s.e.v de $\mathbb{R}^{Q_j}$), on obtient que :

$$F^{\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}} = \inf_{\pi\in\Pi\left(\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}\right)} F(\pi,\cdot) \tag{5.8}$$

La notation de l'équation (5.8) signifie que si $u \in \mathbb{U}$, $F_u^{\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}} = F_u^{\vee}$ et donc $\pi_u \equiv \vee$. Maintenant si $i \notin \mathbb{U}\cup\mathbb{A}\cup\mathbb{I}$ et donc $F_i^{\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}}$ est une constante alors $F_i^{\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}} = F_i^{\sharp}$ et $\pi_i \equiv \sharp$. Les cas intéressants sont les cas $a \in \mathbb{A}$ et $\imath \in \mathbb{I}$. La fonction $F^{\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}}$ s'applique uniquement sur les éléments $v \in \overline{\mathbf{C}}(\overrightarrow{Q}, \overrightarrow{E})$ ce qui signifie que pour tout $a \in \mathbb{A}$:

- Soit $(Q_a, E_a, E_a^*) \in \mathbf{D}(v_{\ell(a)})$ et dans ce cas, pour tout $p \in \mathbb{P}$, $\pi_a(v, p)$ est un vecteur dual de E_a^* et :

$$(F_a(\pi_a, v))(p) = \langle v_{\ell(a)}, \pi_a(v, p)\rangle_{Q_a\times E_a\times E_a^*} + [\mathbb{V}_a^{Q_a,E_a,E_a^*}(\pi_a(v, p))](p) \ .$$

- Soit $v_{\ell(a)}$ prend la valeur $-\infty$ et dans ce cas, $F_a^{\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}}(v) \equiv -\infty$ et ceci indépendamment de E_a^*. Cependant quand celui-ci est fixé, on s'intéresse qu'aux éléments non nuls de E_a^* qui agissent positivement sur les vecteurs positifs de E_a.
- Soit $v_{\ell(a)} \equiv +\infty$ et dans ce cas $F_a^{\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}}(v) = \sup_{x\in\mathbb{R}^d} \mathbf{e}_x$ et ceci indépendamment de E_a^*. On choisira néanmoins le seul vecteur admissible le vecteur nul de E_a^*.

et pour tout $\imath \in \mathbb{I}$:

- Soit $(Q_\imath, E_\imath, E_\imath^*) \in \mathbf{D}(v_{\ell(\imath)})$ et dans ce cas pour tout $p \in \mathbb{P}$, $\pi_\imath(v, p)$ est un vecteur dual de $E_\imath^*$ et :

$$(F_\imath(\pi_\imath(v, p), v))(p) = \langle v_{\ell(\imath)}, \pi_\imath(v, p)\rangle_{Q_\imath\times E_\imath\times E_\imath^*} + [\mathbb{V}_\imath^{Q_\imath,E_\imath,E_\imath^*}(\pi_\imath(v, p))](p) \ .$$

– Soit $v_{\ell(\imath)}$ prend la valeur $-\infty$ et dans ce cas, $F_\imath^{\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}}(v) \equiv -\infty$ et ceci indépendamment de $E_\imath^*$. Cependant quand celui-ci est fixé, on s'intéresse qu'aux éléments non nuls de $E_\imath^*$ qui agissent positivement sur les vecteurs positifs de $E_\imath$.
– Soit $v_{\ell(\imath)} \equiv +\infty$ et dans ce cas $F_\imath^{\overrightarrow{Q},\overrightarrow{E},\overrightarrow{E^*}}(v) = \inf_{\mu\in\mathbb{R}_+} \sup_{x\in\mathbb{R}^d} \mathbf{e}_x - \mu r(x)$ et ceci indépendamment de $E_\imath^*$. On choisira néanmoins le seul vecteur admissible le vecteur nul de $E_\imath^*$.

5.5.2 Itération sur les politiques dynamique dans le domaine des sous-niveaux

Le schéma que nous décrivons ici, n'est qu'un algorithme de principe et un algorithme plus pratique sera présenté au chapitre 6.

Le choix a priori complexe de la dualité et des espaces vectoriels à chaque étape vient en réalité du fait qu'on souhaite avoir des bornes de plus en plus précises et le seul résultat que nous avons montré concernant la croissance est la proposition 5.8. Nous cherchons donc à satisfaire les hypothèses de la proposition. De plus, nous voulons résoudre un problème simple dans à chaque étape c'est-à-dire à politique fixé. Dans le chapitre suivant dans le cas $\mathbb{P}$ fini, nous verrons que le problème du calcul :

$$\inf\{v \in \overline{\mathbf{C}}(\overrightarrow{Q^k}, \overrightarrow{E^k}) \mid F(\pi^k, v) \leq v\}$$

peut se résoudre grâce à un programme linéaire. Pour pouvoir résoudre de tels problèmes, il faut déterminer une politique π^k et pour la déterminer il faut savoir si cette politique existe d'où la condition : $\mathrm{Sol}(Q_j^{k+1}, E_j^{k+1}, (E_j^{k+1})^*, v_{\ell(j)}^{k+1}, p) \neq \emptyset$.

Enfin, le critère d'arrêt de la boucle **répéter** est, si v^{k+1} et v^k prennent des valeurs finies, une distance sur $(\mathbb{R}^{\mathbb{P}})^n$. Ici nous avons étendu cette distance aux fonctions $\overline{\mathbb{R}}^{\mathbb{P}}$, tout en évitant les problèmes d'indétermination. Bien évidemment, la suite générée par le schéma peut être infinie mais nous montrons grâce au théorème 5.6 qu'à chaque pas du schéma, nous gagnons en précision.

Théorème 5.6 (*Décroissance de la suite* $(v_k)_{k\geq 0}$)

La suite générée $(v^k)_{k\geq 0}$ par l'algorithme 5 est décroissante.

Nous allons démontrer ce théorème grâce au lemme suivant :

Lemme 5.2 (*Propriétés de la suite* $(v_k)_{k\geq 0}$)

Soit $k \in \mathbb{N}$, les assertions suivantes sont vraies :

1. Soit il existe $j \in \mathbb{A} \cup \mathbb{I}$, $(Q_j^k, E_j^k, (E_j^k)^*) \notin \mathbf{D}(v_{\ell(j)}^{k+1})$ et l'algorithme s'arrête soit $v^{k+1} \in \overline{\mathbf{C}}(\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}})$;
2. Si $\mathrm{Sol}(Q_j^{k+1}, E_j^{k+1}, (E_j^{k+1})^*, v_{\ell(j)}^{k+1}) \neq \emptyset$, $F(\pi^{k+1}, v^{k+1}) \leq v^{k+1}$.

En conclusion, $v^{k+1} \in \{v \in \overline{\mathbf{C}}(\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}}) \mid F(\pi^{k+1}, v) \leq v\}$.

Démonstration (Lemme 5.2)

1. Soit $j \in \mathbb{A} \cup \mathbb{I}$. S'il existe $p \in \mathbb{P}$ tel que $v_{\ell(j)}^{k+1}(p) = -\infty$ ou si $v_{\ell(j)}^{k+1} \equiv +\infty$ alors $v^{k+1} \in \overline{\mathbf{C}}(Q_j^{k+1}, E_j^{k+1})$ quelque soit Q_j^{k+1} et E_j^{k+1}. S'il existe $j \in \mathbb{A}\cup\mathbb{I}$, $(Q_j^k, E_j^k, (E_j^k)^*) \notin \mathbf{D}(v_{\ell(j)}^{k+1})$ alors l'algorithme s'arrête. Supposons que, pour tout $p \in \mathbb{P}$, $v_{\ell(j)}^{k+1}(p) > -\infty$

Algorithme 5: Itération sur les politiques dans le domaine des sous-niveaux

Entrées : La fonction sémantique abstraite $F^\sharp$
Sorties : Un post-point fixe de la fonction sémantique asbtraite $F^\sharp$
$\forall\, j \in \mathbb{A} \cup \mathbb{I}$, choisir $(Q_j^0, E_j^0, E_j^{0*}) \in \mathbb{D}(\mathbb{P})$ tel que $\forall\, x \in \mathbb{R}^d$, $\mathbf{e}_{x|_{Q_j^0}} \in E_j^0$;
Choisir $\pi^0 \in \Pi(\overrightarrow{Q^0}, \overrightarrow{E^0}, \overrightarrow{E^{0*}})$ et $\gamma > 0$;
$k = 0$, $\forall\, i \in \{1, \ldots, n\},\ v_i^0 \equiv +\infty$;
répéter
NEW $\quad \forall\, j \in \mathbb{A} \cup \mathbb{I}$, $q \in \mathbb{P}$, calculer $\mathbb{V}_j(\pi_i^k) = \left[\mathbb{V}_j(\pi_j^k(q))\right](q)$;
$\quad$ Calculer $v^{k+1} = \inf\{v \in \overline{\mathbf{C}}(\overrightarrow{Q^k}, \overrightarrow{E^k}) \mid F(\pi^k, v) \leq v\}$ dans $(\overline{\mathbb{R}}^{\mathbb{P}})^n$;
$\quad$ **pour tous les** $j \in \mathbb{A} \cup \mathbb{I}$ **faire**
$\quad\quad$ **si** $\forall\, p \in \mathbb{P}$, $v_{\ell(j)}^{k+1}(p) > -\infty$ *et* $\mathrm{dom}(v_{\ell(j)}^{k+1}) \neq \emptyset$ **alors**
$\quad\quad\quad$ **si** $(Q_j^k, E_j^k, (E_j^k)^*) \in \mathbf{D}(v_{\ell(j)}^{k+1})$ **alors**
$\quad\quad\quad\quad$ choisir Q_j^{k+1}, E_j^{k+1} et $(E_j^{k+1})^*$ tels que $(Q_j^{k+1}, E_j^{k+1}, (E_j^{k+1})^*) \in \mathbf{D}(v_{\ell(j)}^{k+1})$ et tels qu'il existe une application i : $(Q_j^k, E_j^k, (E_j^k)^*) \lll_i (Q_j^{k+1}, E_j^{k+1}, (E_j^{k+1})^*)$
$\quad\quad\quad$ **sinon**
$\quad\quad\quad\quad$ Choisir $\forall\, j \in \{1 \ldots, n\}$, $(\mathrm{dom}(v_j^{k+1}), F_j, F_j^*) \in \mathbf{D}(v_j^{k+1})$ tel qu'il existe une application i telle que $F_j >_i \mathbf{B}(v_j^{k+1})$ puis retourner $\mathrm{vex}_{\mathbb{P}}(v_j^{k+1})^{\mathrm{dom}(v_j^{k+1}), F_j, F_j^*}$;
$\quad\quad$ **sinon**
$\quad\quad\quad$ $Q_j^{k+1} = Q_j^k$, $E^{k+1} = E^k$ et $(E^{k+1})^* = (E^k)^*$;
$\quad$ **si** $\forall\, j \in \mathbb{A} \cup \mathbb{I}$, $\forall\, p \in \mathbb{P}$, $\mathrm{Sol}(Q_j^{k+1}, E_j^{k+1}, (E_j^{k+1})^*, v_{\ell(j)}^{k+1}, p) \neq \emptyset$ **alors**
$\quad\quad$ Evaluer $F^{\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}}, \overrightarrow{(E^{k+1})^*}}(v^{k+1})$;
$\quad\quad$ prendre $\pi^{k+1} \in \Pi(\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}}, \overrightarrow{E^{k+1*}})$ tel que $F^{\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}}, \overrightarrow{(E^{k+1})^*}}(v^{k+1}) = F(\pi^{k+1}, v^{k+1})$, retourner à l'étape NEW;
$\quad$ **sinon**
$\quad\quad$ Choisir $\forall\, j \in \{1 \ldots, n\}$, $(\mathrm{dom}(v_j^{k+1}), F_j, F_j^*) \in \mathbf{D}(v_j^{k+1})$ tel qu'il existe une application i telle que $F_j >_i \mathbf{B}(v_j^{k+1})$ puis retourner $\mathrm{vex}_{\mathbb{P}}(v_j^{k+1})^{\mathrm{dom}(v_j^{k+1}), F_j, F_j^*}$;
jusqu'à $\sup_{i \in \{1, \ldots, n\}} \left(\min \left(1, \sup_{\{p \in \mathbb{P} \mid v_i^{k+1}(p) > -\infty,\ v_i^k(p) < +\infty\}} |v_i^{k+1}(p) - v_i^k(p)| \right) \right) \leq \gamma$;

et $\mathrm{dom}(v_{\ell(j)}) \neq \emptyset$ et que pour tout $j \in \mathbb{A} \cup \mathbb{I}$, $(Q_j^k, E_j^k, (E_j^k)^*) \in \mathbf{D}(v_{\ell(j)}^{k+1})$. Par définition de l'algorithme, $(Q_j^{k+1}, E_j^{k+1}, (E_j^{k+1})^*) \in \mathbf{D}(v_{\ell(j)}^{k+1})$ ceci implique que $v_{\ell(j)}^{k+1} \in \mathbf{C}(Q_j^{k+1}, E_j^{k+1})$. Finalement, comme j est arbitraire alors $v^{k+1} \in \overline{\mathbf{C}}(\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}})$.

2. Soit $w \in \mathbf{C}(\overrightarrow{Q^k}, \overrightarrow{E^k})$, et soit $\overrightarrow{(E^k)^*}$ tels que pour tout $j \in \mathbb{A} \cup \mathbb{I}$, $(Q_j^k, E_j^k, E_j^{k*}) \in \mathbb{D}(\mathbb{P})$. On a pour tout $j \in \mathbb{A} \cup \mathbb{I}$, $(Q_j^k, E_j^k, E_j^{k*}) \in \mathbf{D}(w_j)$. Maintenant soit $\pi^k \in \Pi(Q^k, E^k, (E^k)^*)$ et supposons que $F(\pi^k, w) \leq w$ alors par définition de la borne inférieure $v^{k+1} \leq w$, par la proposition 5.8, $F^{\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}}, \overrightarrow{E^{k+1*}}}(v^k) \leq F^{\overrightarrow{Q^k}, \overrightarrow{E^k}, \overrightarrow{E^{k*}}}(w)$. Par définition des politiques, on a : $F^{\overrightarrow{Q^k}, \overrightarrow{E^k}, \overrightarrow{E^{k*}}}(w) \leq F(\pi^k, w)$ et par hypothèse $F(\pi^k, w) \leq w$. Or comme $\mathrm{Sol}(Q_j^{k+1}, E_j^{k+1}, (E_j^{k+1})^*, v_{\ell(j)}^{k+1}, p) \neq \emptyset$, il existe $\pi^{k+1} \in \Pi(\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}}, \overrightarrow{(E^{k+1})^*})$ telle que $F^{\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}}, \overrightarrow{(E^{k+1})^*}}(v^{k+1}) = F(\pi^{k+1}, v^{k+1})$ et on conclut que $F(\pi^{k+1}, v^{k+1}) \leq F(\pi^k, w) \leq w$ pour tout $v \in \mathbf{C}(\overrightarrow{Q^k}, \overrightarrow{E^k})$ tel que $F(\pi^k, w) \leq w$ et par définition de la borne inférieure $F(\pi^{k+1}, v^{k+1}) \leq v^{k+1}$. ∎

Démonstration (Théorème 5.6)
Le lemme 5.2, nous permet de conclure en utilisant encore une fois la définition de la borne inférieure. D'après le lemme 5.2, pour tout $k \geq 0$, $v^{k+2} \leq v^{k+1}$ car $v^{k+1} \in \{v \in \overline{\mathbf{C}}(\overrightarrow{Q^{k+1}}, \overrightarrow{E^{k+1}}) \mid F(\pi^{k+1}, v) \leq v\}$. Comme par définition, $v_i^0 \equiv +\infty$ pour tout $i \in \{1, \ldots, n\}$ alors $(v^k)_{k\geq 0}$ est décroissante. ∎

Exemple 5.26 (Une simple utilisation du schéma 5)
Considérons le programme de la figure 5.4 :

```
assume (x,y) in B(0,1);[1]
while [2](x*x+y*y<=0.5){
    x=x+y;
    y=y+1;[3]
}
```

Figure 5.4 – Un simple programme avec une boucle

Nous nous intéressons aux valeurs prises par les variables x et y aux points de contrôle 1, 2 et 3 et nous considérons la configuration $\mathbb{P} = \{(x, y) \mapsto p_1 x + p_2 y,\ p = (p_1, p_2) \in B(0, 1)\}$. La fonction sémantique abstraite de ce programme est la fonction qui à $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^3$ et pour $p \in \mathbb{P}$ associe $F^\sharp$ définie par :

$$\begin{array}{lcl} [F_1{}^\sharp(v)](p) & = & \|q\| \\ [F_2{}^\sharp(v)](p) & = & (v_1^{\mathbf{s}} \cup v_3^{\mathbf{s}})^\sigma(p) \\ [F_3{}^\sharp(v)](p) & = & \sup\limits_{\substack{(x,y)\in v_2^{\mathbf{s}} \\ x^2+y^2\leq 0.5}} p_1(x+y) - p_2(y+1) \end{array}$$

Initialisons le schéma par le domaine des intervalles, c'est-à-dire prenons, au point de contrôle 3 :

$$Q_3^0 = \{(x, y) \mapsto x,\ (x, y) \mapsto -x,\ (x, y) \mapsto y,\ (x, y) \mapsto -y\}$$

et ainsi $\overrightarrow{Q^0} = Q_3^0$ et prenons $E_3^0 = \mathbb{R}^{Q_3^0}$ et $(E_3^0)^* = \mathbb{R}^{Q_3^0}$ ce qui signifie $\overrightarrow{E^0} = E_3^0$ et $\overrightarrow{(E^0)^*} = (E_3^0)^*$). En absence d'ambiguïté, nous notons $\pi^0(p)$ à la place de $\pi^0(v,p)$. Nous cherchons à déterminer une politique initiale $\pi^0(p) = (\pi_1^0(p), \pi_2^0(p), \pi_3^0(p))$. Comme les points de contrôle 1 et 2 représentent respectivement une constante et une union, ces fonctions sont connues. Nous nous intéressons au point de contrôle 3, la politique au point de contrôle 3, $\pi_3^0(p)$, est un élément de $(E_3^0)^* \times \mathbb{R}_+$. Nous notons $\lambda^0(p)$ la première coordonnée et $\mu^0(p)$ la deuxième coordonnée de l'élement $\pi_3^0(p)$. Pour simplifier les notations, e_1 désigne le vecteur $(1,0)$, e_2 le vecteur $(0,1)$.

Soient $q \in \mathbb{P}$, $\lambda(q)$ une fonction de Q_3^0 dans $\mathbb{R}^+$ et $\mu(q)$ un réel positif :
$[\mathbb{V}_3^{Q_3^0,E_3^0,(E_3^0)^*}(\pi^0(q))](q) =$

$$\sup_{(x,y)\in\mathbb{R}^2} \begin{array}{l} q_1(x+y) + q_2(y+1) + \mu^0(q)(\frac{1}{2} - x^2 - y^2) - x[\lambda^0(q)](e_1) + x[\lambda^0(q)](-e_1) \\ -y[\lambda^0(q)](e_2)v_2(e_2) + y[\lambda^0(q)](-e_2) \end{array}$$

ou encore :
$[\mathbb{V}_3^{Q_3^0,E_3^0,(E_3^0)^*}(\pi^0(q))](q) =$

$$\sup_{(x,y)\in\mathbb{R}^2} \begin{array}{l} q_2 + \frac{1}{2}\mu^0(q) + x(q_1 - [\lambda^0(q)](e_1) + [\lambda^0(q)](-e_1)) \\ +y(q_1 + q_2 - [\lambda^0(q)](e_2) + [\lambda^0(q)](-e_2)) - \mu^0(q)(x^2+y^2) \end{array}$$

La fonction $(x,y) \mapsto q_2 + \frac{1}{2}\mu^0(q) + x(q_1 - [\lambda^0(q)](e_1) + [\lambda^0(q)](-e_1)) + y(q_1 + q_2 - [\lambda^0(q)](e_2) + [\lambda^0(q)](-e_2)) - \mu^0(q)(x^2+y^2)$ est une fonction quadratique concave, pour calculer son supremum sur $\mathbb{R}^2$, il suffit de trouver en quels points la différentielle s'annule.

Commençons par choisir une politique initiale qui ne prend en compte que la contrainte due au test c'est-à-dire, nous considérons $[\lambda^0(q)](e_1) = \lambda^0(q)](-e_1) = \lambda^0(q)](e_2) = \lambda^0(q)](-e_2) = 0$ et $\mu^0(q) > 0$. Cette heuristique correspond à l'heuristique de l'itération sur les politiques dans le domaine des intervalles de la définition 3.10. On obtient ainsi la nouvelle fonction :

$$[\mathbb{V}_3^{Q_3^0,E_3^0,(E_3^0)^*}(\pi^0(q))](q) = \sup_{(x,y)\in\mathbb{R}^2} q_2 + \frac{1}{2}\mu^0(q) + q_1 x + y(q_1+q_2) - \mu^0(q)(x^2+y^2)$$

et ainsi on conclut que :

$$[\mathbb{V}_3^{Q_3^0,E_3^0,(E_3^0)^*}(\pi^0(q))](q) = q_2 + \frac{1}{2}\mu^0(q) + \frac{q_1^2 + (q_1+q_2)^2}{4\mu^0(q)}$$

Prenons donc, par exemple, pour tout $q \in \mathbb{P}$, $\mu^0(q) = 1$ alors, $[\mathbb{V}_3^{Q_3^0,E_3^0,(E_3^0)^*}(\pi^0(q))](q) = q_2 + \frac{1}{2} + \frac{q_1^2 + (q_1+q_2)^2}{4}$.

Finalement la politique initiale au point de contrôle 3 est le couple composé de la fonction constante $[\lambda^0(q)](e_1) = [\lambda^0(q)](-e_1) = [\lambda^0(q)](e_2) = [\lambda^0(q)](-e_2) = 0$ ainsi que de l'élément $\mu^0(q) = 1$. En considérant une telle politique initiale, nous obtenons :

$$[\mathbb{V}_3^{Q_3^0,E_3^0,(E_3^0)^*}(\pi^0(q))](q) = q_2 + \frac{1}{2} + \frac{q_1^2 + (q_1+q_2)^2}{4} \quad .$$

Nous devons à présent trouver le plus petit élément :

$$v = (v_1, v_2, v_3) \in \overline{\mathbb{R}}^{\mathbb{P}} \times \overline{\mathbb{R}}^{\mathbb{P}} \times \mathbf{C}(Q_3^0, E_3^0)$$

tel que :

$$\|q\| \leq v_1(q),\ \forall q \in \mathbb{P}$$

et

$$\sup(v_1, v_3)(q) \leq v_2(q), \forall q \in \mathbb{P}$$

et

$$0\times v_2(e_1)+0\times v_2(-e_1)+0\times v_2(e_2)+0\times v(-e_2)+q_2+\frac{1}{2}+\frac{q_1^2+(q_1+q_2)^2}{4} \leq v_3(q), \forall q=(q_1,q_2)\in\mathbb{P} \tag{5.9}$$

La solution de ce système est donc :

$$v = (v_1, v_2, v_3) \in \overline{\mathbb{R}}^{\mathbb{P}} \times \overline{\mathbb{R}}^{\mathbb{P}} \times \mathbf{C}(Q_3^0, E_3^0)$$

tel que :

$$v_1^1(q) = \|q\|,\ \forall q \in \mathbb{P}$$

et

$$v_2^1(q) = \sup(\|q\|, q_2 + \frac{1}{2} + \frac{q_1^2+(q_1+q_2)^2}{4})), \forall q \in \mathbb{P}$$

et

$$v_3^1(q) = q_2 + \frac{1}{2} + \frac{q_1^2+(q_1+q_2)^2}{4}, \forall q=(q_1,q_2)\in\mathbb{P}$$

L'information importante ici est l'invariant de boucle trouvé, on sait donc qu'au point de contrôle 2, $q_1x + q_2y \leq \sup(\|q\|, q_2 + \frac{1}{2} + \frac{q_1^2+(q_1+q_2)^2}{4}))$ quelque soit $q \in \mathbb{P}$. On obtient en même temps une information de type "intervalles" :

$$-1 \leq x \leq 1 \text{ et } -1 \leq y \leq \frac{7}{4},$$

une information de type "zones" :

$$x - y \leq \sqrt{2} \text{ et } y - x \leq \sqrt{2}\frac{5+4\sqrt{2}}{8}\ .$$

On obtient de la même manière des informations de type "octogones" et "gabarits linéaires". On remarque qu'également, la fonction v_2^1 est une fonction continue de $\mathbb{P}$ dans $\mathbb{R}$, et ainsi nous pouvons prendre :

$$Q_3^1 = \mathbb{P},\ E_3^1 = \mathcal{C}(\mathbb{P}, \mathbb{R})$$

ainsi que la dualité entre $\mathbb{P}$ et l'ensemble $(E_3^1)^*$ des mesures boréliennes finies signées sur $\mathbb{P}$. D'après l'exemple 5.25, nous savons qu'il existe $i : \mathbb{R}_+^{Q_3^0} \mapsto (E_3^1)_+^*$, telle que :

$$(Q_3^1, E_3^1, (E_3^1)^*) \ggg_i (Q_3^0, E_3^0, (E_3^0)^*)\ .$$

La fonction sémantique relâchée au point de contrôle 3 devient alors :

$$\begin{aligned}[F_3^{\overrightarrow{Q^1},\overrightarrow{E^1},\overrightarrow{E^1}}(v^1)](p) &= \inf_{\lambda\in(E_3^1)_+^*}\inf_{\mu\in\mathbb{R}_+}\sup_{(x,y)\in\mathbb{R}^2} p_1(x+y) - p_2(y+1) + \mu(1/2 - x^2 - y^2) \\ &+ \int_{q\in\mathbb{P}} v_2^1(q)d\lambda(q) - x\int_{q\in\mathbb{P}} q_1 d\lambda(q) - y\int_{q\in\mathbb{P}} q_2 d\lambda(q)\end{aligned}$$

On, a de plus, la condition de Slater puisque :

$$q_1 \times 0 + q_2 \times 0 = 0 < \|q\| \leq v_2(q) \ \forall\, q \in \mathbb{P} \text{ non nul}$$
$$\text{et } 0^2 + 0^2 < \frac{1}{2}$$

et donc il existe π^1 telle que $F_3^{Q_3^1, E_3^1, (E_3^1)^*}(v^1) = F_3(\pi^1, v^1)$. De plus, $\mathbb{P}$ est une configuration bornée puisqu'elle contient les intervalles, $(v_2^1)^{\mathbf{s}} \neq \emptyset$ et $\text{dom}(v_2^1) = \mathbb{P}$, donc $F_3{}^\sharp(v^1)(p)$ est fini pour tout $p \in \mathbb{P}$. Enfin, comme chaque $p \in \mathbb{P}$ est linéaire, l'affectation au point de contrôle 3 est affine et le test convexe alors $F_3{}^\sharp(v^1) = F_3^{Q_3^1, E_3^1, (E_3^1)^*}(v^1)$.

On peut utiliser un solveur de programmation semi-infinie pour calculer π_3^1 comme, par exemple, **SIPAMPL** [2] [VFG04, VF06]. Des résultats sur la discrétisation basé sur le théorème d'Helly [BTBIR79, HK93, Sha09] des problèmes d'optimisation semi-infinie montrent que les mesures solutions sont en réalité à support fini, c'est-à-dire des sommes finies positivement pondérées de masses de Dirac et l'intégrale est en fait le plus souvent une simple somme.

Une fois la politique π_3^1 calculée, on doit calculer $[\mathbb{V}_3^{Q_3^1, E_3^1, (E_3^1)^*}(\pi^1(q))](q)$, pour tout $q \in \mathbb{P}$ puis nous devons trouver le plus petit élément :

$$v = (v_1, v_2, v_3) \in \overline{\mathbb{R}}^{\mathbb{P}} \times \overline{\mathbb{R}}^{\mathbb{P}} \times \mathbf{C}(Q_3^1, E_3^1)$$

tel que :

$$\|q\| \leq v_1(q), \ \forall q \in \mathbb{P}$$

et

$$\sup(v_1, v_3)(q) \leq v_2(q), \forall q \in \mathbb{P}$$

et

$$\int_{p \in \mathbb{P}} v_2(p) d[\lambda^1(q)](p) + [\mathbb{V}_3^{Q_3^1, E_3^1, (E_3^1)^*}(\pi^1(q))](q) \leq v_3(q), \forall q = (q_1, q_2) \in \mathbb{P}$$

On peut prendre, puisque v_3 est constrainte à être une fonction continue, une mesure finie γ et on considère une fonction objective de la forme :

$$\sum_{i=1,2,3} \int_{p' \in \mathbb{P}} v_i(p') d\gamma(p')$$

et conserver les contraintes précédentes, on obtient un programme linéaire en dimension infinie.

2. Développé par I. Vaz, voir http://www.norg.uminho.pt/aivaz/sipampl.html

CHAPITRE 6

CONFIGURATIONS QUADRATIQUES ET RELAXATION DE SHOR

Ce chapitre est une partie détaillée de l'article [AGG10, Section 3]. L'algorithme d'itérations sur les politiques défini au chapitre 5 sont construits sur des espaces abstraits. Nous appliquons dans ce chapitre, cet algorithme à un ensemble *fini* de fonctions de base $\mathbb{P}$ particulier : des fonctions quadratiques. Un exemple de fonctions de base quadratique peut être donné par la fonction de Lyapunov d'un système linéaire convergeant. Pour des fonctions de base quadratique, évaluer la fonction sémantique abstraite se ramène à résoudre un problème de maximisation quadratique éventuellement non-convexe. Cependant, il existe des techniques de relaxation convexe des problèmes quadratiques non-convexes comme la relaxation de Shor. La relaxation de Shor coïncide avec la fonction relâchée lagrangienne et donc avec notre fonction sémantique relâchée. Par ailleurs, la relaxation de Shor consiste à évaluer la fonction relâchée en résolvant un programme d'optimisation semi-definie. Le fait important est qu'il existe des méthodes rapides permettant de résoudre des problème d'optimisation semi-definie. En effet, une solution ε-optimale (l'image de la solution est au plus ε de la valeur optimale) peut être calculée en temps polynomial par la méthode de l'ellipsoïde [GLS88] ou par la méthode des points intérieurs [NN94](lorsqu'il existe une matrice strictement réalisable). La complexité polynomiale annoncée ici fait référence au modèle des bits (machine de Turing) [GJ79] pour la méthode de l'ellipoïde et uniquement [1] pour le modèle des nombres réels pour la méthode des points intérieurs bien que cette méthode soit plus efficace en pratique (voir le survol de Pardalos et Ramana [PR97]).

La première section 6.1 est un rappel d'optimisation quadratique non-convexe ainsi qu'un rappel sur la relaxation de Shor. La deuxième section 6.2 constitue une application nouvelle de l'optimisation quadratique : la résolution de problème de point fixe en interprétation abstraite.

6.1 Optimisation quadratique

Dans la sous-section 6.1.1, nous rappelons quelques résultats basiques d'optimisation quadratique ainsi que quelques notations. Dans la sous-section 6.1.2, nous introduisons la re-

1. Le problème de la polynomialité de la méthode des points intérieurs sur le modèle des bits est un problème ouvert voir Pardalos et Ramana [PR97, Open Problem 9.2]

laxation de Shor et enfin nous terminons sur la sous-section 6.1.3 sur des résultats numériques d'approximation ainsi que des résultats de dualité forte pour des problèmes d'optimisation quadratique non-convexe.

6.1.1 Rappels et notations

Nous rappelons qu'une forme quadratique est une fonction f de $\mathbb{R}^d$ dans $\mathbb{R}$ de la forme :

$$f(x) = x^T Ax + x^T b + c$$

où une matrice A symétrique de taille $d \times d$, un vecteur b de $\mathbb{R}^d$ et un réel c. Pour une forme quadratique f, on notera A_f, b_f et c_f respectivement sa matrice associée, son vecteur associé et son scalaire associé. Une forme quadratique f est dite *homogène* si $b_f = 0$ et $c_f = 0$. Dans notre cas, on s'intéressera aux templates quadratiques à scalaire nul ($c_f = 0$).

On dit qu'un problème d'optimisation (maximisation ici) est quadratique s'il est de la forme :

$$\begin{array}{ll} \text{Max} & f_0(x) \\ \text{s. c} & f_i(x) \leq 0 \\ & \forall\, i = 1, \ldots, m \end{array} \tag{QP}$$

où $f_0, f_1, \ldots, f_m$ sont des formes quadratiques sur $\mathbb{R}^d$. En remplaçant l'opérateur Max par Min, on obtient un problème de minimisation quadratique.

Pour commencer, nous faisons quelques rappels sur les matrices définies. Soit une matrice A symétrique de taille $d \times d$: A est semi-définie positive si $x^T Ax \geq 0$ pour tout $x \in \mathbb{R}^d$; A est définie positive si $x^T Ax > 0$ pour tout $x \neq 0$; A est semi-définie négative si $x^T Ax \leq 0$ pour tout $x \in \mathbb{R}^d$; A est définie négative si $x^T Ax < 0$ pour tout $x \neq 0$; s'il existe $x \in \mathbb{R}^d$ et $y \in \mathbb{R}^d$ tels que $x^T Ax < 0$ et $y^T Ay > 0$, on dit que la matrice est indéfinie. On note respectivement, S_d^+, S_d^{++}, S_d^- et S_d^{--} le cône convexe des matrices symétriques semi-définies positives, positives, semi-définies négatives et négatives. On munit également l'ensemble des matrices symétriques S_d de l'ordre de Loewner défini par $A \preceq B \iff B - A \in S_d^+$. La notation $A \succeq 0$ signifie que A est semi-définie positive et on note $\succ$, $\preceq$ et $\prec$ pour définie positive, semi-définie négative et définie négative. On munit S_d du produit scalaire de Frobenius i.e le produit scalaire $\langle \cdot, \cdot \rangle_F$ défini sur S_d par $\langle A, B \rangle_F = \sum_{i,j=1}^{d} A_{ij} B_{ij}$.

Revenons aux fonctions $f_0, f_1, \ldots, f_m$, ces fonctions sont quadratiques, la convexité et la concavité dans ce cas sont déterminées par la positivité des matrices $A_0, A_1, \ldots, A_m$ associées. En effet, une fonction quadratique est convexe (concave) si et seulement sa matrice associée est semi-définie positive (négative). Lorsque A_0 est semi-définie positive et $A_1, \ldots, A_m$ sont semi-définies négatives, le problème de maximisation P est facile à résoudre, dans les autres cas, le problème est *NP-dur* [Vav90]. Une manière de donner une borne sur la valeur du problème est de calculer la valeur du problème dual qui a de bonnes propriétés. Dans le cas quadratique, on peut évaluer facilement la valeur du problème dual.

6.1.2 Relaxation de Shor

La relaxation de Shor est détaillée dans [TN01, Sho87]. Nous donnons les détails de la relaxation de Shor qui, dans le cadre des fonctions quadratiques, est une réécriture de la

fonctionnelle duale de la dualité lagrangienne comme un problème d'optimisation semi-définie. Par la suite, nous utiliserons la relaxation de Shor pour evaluer la fonction sémantique relâchée.

On se donne un problème de maximisation quadratique P. Nous rappelons que la valeur du problème dual de P pour la dualité lagrangienne est donnée par l'équation (6.1).

$$\inf_{\lambda \in \mathbb{R}^m_+} \sup_{x \in \mathbb{R}^d} f_0(x) - \sum_{i=1}^{m} \lambda_i f_i(x) \tag{6.1}$$

Une première remarque simple est de voir qu'à λ fixé, la fonction définie à l'équation (6.1) est un problème de maximisation sans contrainte. Par ailleurs, la valeur du problème non contraint est fini si et seulement si il existe $\eta \in \mathbb{R}$ tel que, pour tout $x \in \mathbb{R}^d$,

$$f_0(x) - \sum_{i=1}^{m} \lambda_i f_i(x) \leq \eta \ . \tag{6.2}$$

On introduit la matrice $A(\lambda)$, le vecteur $B(\lambda)$ et le scalaire $C(\lambda)$ définis par :

$$A(\lambda) = A_{f_0} - \sum_{i=1}^{m} \lambda_i A_{f_i}, \ B(\lambda) = b_{f_0} - \sum_{i=1}^{m} \lambda_i b_{f_i} \text{ et } C(\lambda) = c_{f_0} - \sum_{i=1}^{m} \lambda_i c_{f_i} \ .$$

Ainsi, on peut réécrire l'équation (6.2), pour tout $x \in \mathbb{R}^d$,

$$x^T A(\lambda) x + B(\lambda) x + C(\lambda) - \eta \leq 0 \ .$$

Maintenant en écrivant que $\sup_{x \in \mathbb{R}^d} x^T A(\lambda) x + B(\lambda) x + C(\lambda) \leq \eta$ où $\eta \in \mathbb{R}$ est équivalent à la définie positivité de la matrice :

$$x^T A(\lambda) x + B(\lambda) x + C(\lambda) - \eta = \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{pmatrix} C(\lambda) - \eta & \frac{1}{2} B(\lambda) \\ \frac{1}{2} B(\lambda)^T & A(\lambda) \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}$$

on conclut que $\sup_{x \in \mathbb{R}^d} x^T A(\lambda) x + B(\lambda) x + C(\lambda) - \eta \leq 0$ est équivalent au fait que la matrice

$$\begin{pmatrix} C(\lambda) - \eta & \frac{1}{2} B(\lambda) \\ \frac{1}{2} B(\lambda)^T & A(\lambda) \end{pmatrix}$$

soit semi-définie négative. Trouver exactement le supremum dans l'équation 6.1 revient à prendre η le plus petit possible, donc on conclut que la valeur de l'équation 6.1 coïncide avec la valeur du problème d'optimisation semi-definie :

$$\underset{\substack{\lambda \in \mathbb{R}^m_+ \\ \eta \in \mathbb{R}}}{\text{Min}} \ \eta \quad \text{s. c} \quad \begin{pmatrix} C(\lambda) - \eta & \frac{1}{2} B(\lambda) \\ \frac{1}{2} B(\lambda)^T & A(\lambda) \end{pmatrix} \preceq 0 \tag{Shor}$$

6.1.3 Relaxations de problème d'optimisation quadratique et bornes de qualité

Nous proposons un bref survol des travaux effectués sur la relaxation de problèmes d'optimisation quadratique par un problème de programmation semi définie. Plus précisément, nous distinguons deux types de travaux : l'étude de la qualité de la relaxation, c'est-à-dire, des travaux pour lesquels les auteurs cherchent à évaluer le rapport entre la valeur du problème quadratique et celle du problème SDP relâché ; les travaux portant sur les problèmes quadratiques dont la relaxation SDP est exacte.

Dans la sous-section 6.1, nous avons introduit la relaxation de Shor. On peut également directement relâcher un problème quadratique sans faire intervenir de vecteurs duaux. On peut réécrire une fonction quadratique à l'aide du produit scalaire de Frobenius. Pour une fonction quadratique f on a $x^T A_f x + b_f^T x + c_f = \langle Q_f, X\rangle_F$ où $Q_f = \begin{pmatrix} c_f & (1/2)b_f^T \\ (1/2)b_f & A_f \end{pmatrix}$ et $X = \begin{pmatrix} 1 \\ x \end{pmatrix}\begin{pmatrix} 1 \\ x \end{pmatrix}^T$. Ainsi on peut réécrire un problème de maximisation quadratique sous la forme :

$$\begin{array}{ll} \text{Max} & \langle Q_{f_0}, X\rangle_F \\ \text{s. c} & \langle Q_{f_i}, X\rangle_F \leq 0 \\ & \forall\, i = 1, \ldots, m \\ & X_{11} = 1,\ X \in S_d^+,\ \text{rg}(X) = 1 \end{array} \tag{QP'}$$

La notation rg désignant le rang d'une matrice. Ce problème n'est pas convexe à cause de la contrainte de rang. En relâchant la contrainte de rang, on obtient un problème convexe qui est le problème dual de (Shor).

$$\begin{array}{ll} \text{Max} & \langle Q_{f_0}, X\rangle_F \\ \text{s. c} & \langle Q_{f_i}, X\rangle_F \leq 0 \\ & \forall\, i = 1, \ldots, m \\ & X_{11} = 1,\ X \in S_{d+1}^+ \end{array} \tag{ShorD}$$

Si (ShorD) admet une matrice positive strictement réalisable i.e. $X \in S_d^{++}$ telle que $\langle Q_{f_i}, X\rangle_F < 0$ pour tout $i = \{1, \ldots, m\}$ et $X_{11} = 1$, alors les valeurs de (ShorD) et (Shor) coïncident.

Qualité de la relaxation SDP

Les travaux d'études sur la qualité de la relaxation (ShorD) pour des problèmes quadratiques ont été initiés par le travail de Goemans et Williamson [GW95]. Leur motivation était tout autre puisque Goemans et Williamson souhaitaient garantir une solution réalisable du problème de coupe maximale ($MAXCUT$) pas trop éloignée de la solution optimale. Goemans et Williamson ont exprimé le problème de la coupe maximale comme un problème quadratique en nombre entier. Le problème quadratique a une structure particulière : $m = d$, les matrices A_{f_i} sont de la forme $e^i(e^i)^T$ où e^i désigne la base canonique de $\mathbb{R}^d$, $b_{f_0} = b_{f_i} = 0$, $c_{f_i} = -1$ et $A_{f_0} \in S_d^+$, ses éléments hors diagonaux étant strictement négatifs et la somme de chaque ligne est nulle. Goemans et Williamson prouvent grâce à des arguments de type stochastiques que, pour ce problème particulier,

$$\text{Val}(ShorD) \geq \text{Val}(QP) \geq (0.8781)\,\text{Val}(ShorD)\ .$$

Dans [Ye99a], Ye s'intéresse aux problèmes de maximisation (QP) homogénéisés c'est-à-dire lorsque la fonction objective q satisfait $q(\alpha x) = \alpha^2 q(x)$ pour $\alpha \in \mathbb{R}$ et avec contraintes de boîte : de la forme $x \in [-1,1]^d$. La classe des problèmes quadratiques homogénéisés avec contraintes de boîte contient également les problèmes dont la fonction objective est de la forme $x^T Q x + b^T x$ puisqu'il suffit d'ajouter une variable $t \in [-1,1]$ pour le rendre homogène (prendre $x^T Q x + t b^T x$ comme fonction objective). En reprenant notre notation, les conditions du problème sont les suivantes : les matrices A_{f_i} ont pour entrée 1 pour la coordonnée (i,i) et 0 ailleurs, $b_{f_0} = b_{f_i} = 0$, $c_{f_0} = 0$, $c_{f_i} = -1$ et enfin la matrice A_{f_0} est symétrique indéfinie ou symétrique semidéfinie positive. Ye montre alors que :

$$\text{Val}(ShorD) \geq \text{Val}(QP') \geq (\pi/2)\,\text{Val}(ShorD) \ .$$

Dans [Ye99b], Ye montre que ce résultat demeure vrai si on ajoute des contraintes d'égalités quadratiques. Par ailleurs, Ye dans [Ye99a, Ye99b] utilisent des techniques développées par Nesterov décrites dans [Nes97, Nes98].

Egalité entre problème quadratique et relaxation SDP

Une deuxième classe de travaux intéressante est l'étude des problèmes quadratiques qui se résolvent de manière exacte par la relaxation SDP (ShorD). On peut citer notamment le travail de Zhang [Zha00]. Zhang s'intéresse à des problèmes de maximisation quadratique avec une unique contrainte du type $x^2 := (x_1^2, \ldots, x_d^2) \in F$ où F est un sous-ensemble convexe de $\mathbb{R}^d$ et où la fonction objective est homogène (i.e $b_{f_0} = 0$ et $c_{f_0} = 0$), la contrainte $x^2 \in F$ se transformant en une contrainte $(X_{22}, \ldots, X_{d+1\,d+1}) \in F$ dans le problème SDP (ShorD). Zhang [Zha00, Theorem 2] montre que, lorsque tous les termes hors diagonale de la matrice Q_{f_0} sont positifs ou nuls alors $\text{Val}(QP) = \text{Val}(ShorD)$. Dans le [Zha00, Theorem 4], Zhang étend son précédent résultat aux matrices Q_{f_0} telles qu'il existe $\sigma \in \{-1,1\}^{d+1}$ satisfaisant $Q_{f_0}(i,j)\sigma_i\sigma_j \geq 0$ pour tout i,j, $i \neq j$ (Zhang appelle cette condition, la condition de *almost OD-nonnegative*). Par ailleurs, Zhang montre qu'on peut calculer des solutions optimales de QP à partir des solutions optimales de (ShorD).

Kim et Kojima [KK03] ont étendu le travail de Zhang en montrant que $\text{Val}(QP) = \text{Val}(ShorD)$ demeure vrai dès que toutes les matrices des contraintes Q_{f_i} et Q_{f_0} ont des éléments hors diagonale positifs ou nuls, ou dès que la famille des $(Q_{f_i})_{i=\{1\ldots,m\}}$ vérifie une condition *almost OD-nonnegative* uniforme.

Plusieurs travaux [Mor93, SW95, BTT96, Pol98] font mention du cas particulier des problèmes quadratiques avec une unique contrainte (i.e. $m = 1$) : s'il existe $\gamma \geq 0$ tel que $A_{f_0} - \gamma A_{f_1} \in S_-^d$ alors $\text{Val}(QP) = \text{Val}(ShorD)$. Beck a généralisé ce résultat aux problèmes *quadratiques matriciels* réels dans [Bec07] puis, dans [Bec09] aux problèmes *quadratiques matriciels* complexes.

Optimisation sur le cône du second ordre et relaxation de problèmes quadratiques

Il existe un autre type de relaxation appelé optimisation sur le cône du second ordre[2]. Cette dénomination fait référence au cône de Lorentz, c'est-à-dire, le cône défini par $\{(x,t) \in \mathbb{R}^d \times \mathbb{R}_+ \mid \|x\| \leq t\}$. L'optimisation sur le cône du second ordre consiste à contraindre les variables du problème à appartenir à un cône de Lorentz pour une certaine dimension. Pour

2. Traduction littérale du groupe de mots anglais *Second Ordrer Cone Programming*.

plus de détails sur la relaxation de problèmes quadratiques par des problèmes du second ordre, le lecteur peut consulter [KK00, KK03].

6.2 Configurations quadratiques en interprétation abstraite

Dans cette section, nous nous intéressons à des configurations quadratiques $\mathbb{P}$ finies. Les problèmes d'optimisation que nous rencontrerons au cours de cette section seront des problèmes d'optimisation avec un nombre *fini* de contraintes. Nous utiliserons donc la dualité lagrangienne classique où le crochet de dualité est donné par le produit scalaire usuel des espaces vectoriels finis. La sous-section 6.2.1 porte sur la fonction sémantique relâchée dans le cas particulier des fonctions de base quadratiques, la sous-section 6.2.2 décrit l'itération sur les politiques dans ce cadre.

6.2.1 Fonction sémantique relâchée dans les configurations quadratiques

Dans toute cette section, nous supposons que les affectations sont des applications quadratiques ou affines et les tests sont des fonctions quadratiques ou affines. Une affectation T possède des coordonnées T_i de la forme :

$$x \mapsto x^T A_i x + b_i^T x + c_i$$

où A_i sont des matrices de taille $d \times d$, b_i sont des vecteurs de $\mathbb{R}^d$ et c_i sont des scalaires Nous autorisons les tests quadratiques, c'est-à-dire de la forme $x \in \{y \in \mathbb{R}^d \mid y^T A'_i y + y^T b'_i + c_i \leq 0\}$.

Définition 6.1 (*Configurations quadratiques*)

L'ensemble des fonctions de base $\mathbb{P}$ est une configuration quadratique si $\mathbb{P}$ est un ensemble **fini** de formes quadratiques.

Le but de cette section est d'adapter la relaxation de Shor pour des calculs d'invariants numériques non polyèdriques. On s'intéresse à des configurations quadratiques compatibles avec l'arithmétique du programme analysé. Plus particulièrement, pour des programmes écrits en arithmétique quadratique, nous utilisons une configuration linéaire et pour des programmes écrits en arithmétique affine, nous utilisons une configuration quadratique.

Pour $i \in \mathbb{A} \cup \mathbb{I}$, on note l'affectation T_i du programme (interne à une boucle ou un branchement conditionnel si $i \in \mathbb{I}$) ou simple (si $i \in \mathbb{A}$). On admet l'hypothèse suivante :

$$\forall\, i \in \mathbb{A} \cup \mathbb{I},\ p \circ T_i \text{ est une fonction quadratique pour tout } p \in \mathbb{P}\ . \qquad \text{(HQ)}$$

Comme pour les gabarits linéaires de Sankaranarayanan et al [SSM05], trouver automatiquement la configuration quadratique à considérer pour analyser un programme le plus précisément possible est une question ouverte.

Définition 6.2 (*Fonction de Lyapunov pour les systèmes lineaires discrets*)

Soit un système linéaire en temps discret $u(t+1) \leftarrow A(u(t))$. Une fonction de la forme $x \mapsto x^T Q x$ où Q est une matrice définie positive est une fonction de Lyapunov du système ssi, il existe un réel $\gamma \in]0, 1]$ tel que :

$$A^T Q A - \gamma Q$$

est une matrice définie négative

Nous savons que si un système linéaire en temps discret est convergeant alors il existe une fonction de Lyapunov quadratique positive. Il existe des travaux et des algorithmes pour calculer numériquement une fonction de Lyapunov pour des systèmes linéaires discrets [Bar77, For91, BEGFB94, Sim96]. Nous proposons d'utiliser une fonction de Lyapunov ainsi que des contraintes de types boîtes du types $x_i \mapsto x_i^2$ (quadratiques homogènes) ou $x_i \mapsto \pm x_i$ (intervalles classiques).

Exemple 6.1 (Rotation en dimension *d*)
On va vérifier grâce aux configurations quadratiques que la sphère unité est invariante par rotation. Posons $S = \{x \in \mathbb{R}^d \mid \|x\| = 1\}$ et soit A une matrice orthogonale ($A^T A = AA^T =$ Id) de taille $d \times d$. Considérons le programme de la figure 6.1 :

```
assume x in S;[1]
       y=Ax;[2]
```

Figure 6.1 – Programme simulant une rotation

On pose $\mathbb{P} = \{x \mapsto -\|x\|^2, x \mapsto \|x\|^2\}$. La configuration $\mathbb{P}$ est une configuration quadratique.

Exemple 6.2 (Oscillateur harmonique)
On s'intéresse à un oscillateur harmonique de la forme $m\ddot{x} + c\dot{x} + kx = 0$[3]. En notant v la vitesse, la discrétisation du système en temps par un schéma d'Euler conduit au système discret :

$$\begin{cases} v_h &= v + h(-(k/m)x - (c/m)v) \\ x_h &= x + hv \end{cases}$$

où h est le pas de discrétisation.

Pour simplifier les calculs, nous prenons à titre d'exemple $m = c = k = 1$. Pour évaluer la trajectoire du système dans l'espace des phases, nous proposons une implémentation du schéma discrétisé à la figure 6.2.

```
assume x in [0,1];
assume v in [0,1];
h = 0.01;
while (true) { [2]
  w = v;
  v = v*(1-h)-h*x;
  x = x+h*w; [3] }
```

Figure 6.2 – Implémentation de la discrétisation de l'oscillateur harmonique de l'exemple 6.2

3. Pour rappel, $m \neq 0$ est la masse, c le coefficient de frottement et k la constante de raideur. L'accélération est représentée par $\ddot{x}$, la vitesse par $\dot{x}$ et la position par x

En utilisant des techniques classiques comme l'interprétation abstraite sur le domaine sur les polyèdres, nous ne pouvons pas montrer que les valeurs des variables du programme sont bornées. Par exemple, en utilisant Interproc [4], nous trouvons au point de contrôle [2] que $h = 0.01$ et donc aucune information sur les variables x et v. On sait que le système discrétisé admet une fonction de Lyapunov, par exemple la fonction $(x, v) :\mapsto L(x, v) = 3v^2 + x^2 + 2xv$. Ceci assure la stabilité du schéma et donc la finitude des valeurs de x et v.

Nous allons utiliser les techniques développées dans le chapitre 5 pour prouver que les valeurs des variables de l'implémentation sont bornées.

Nous prenons comme fonction de base $p_x : (x, v) \mapsto x^2$, $p_v : (x, v) \mapsto v^2$ et $p_L : (x, v) \mapsto 3v^2 + 2x^2 + 2xv$, la configuration $\mathbb{P} = \{p_x, p_v, p_L\}$ forme ainsi une configuration quadratique.

Exemple 6.3 (Un simple programme simulant un système non convergeant)
On s'intéresse à un programme contenant une boucle **while** et test non trivial. Ce programme est décrit par la figure 6.3.

```
i=0;
j=0;[1]
  while [2](i<=42){
      i=i+1;
      j=j+i;[3]
      }
```

Figure 6.3 – Un programme simple avec une boucle et un test

On cherche à prouver, en utilisant l'itération sur les politiques pour les configurations quadratiques, que $j \leq \dfrac{i(i+1)}{2}$. Par conséquent, nous introduisons la configuration quadratique $\mathbb{P} = \{(i, j) \mapsto -\dfrac{i(i+1)}{2} + j\}$.

Dans le chapitre 5, nous avons décrit la fonction sémantique relâchée à partir d'un crochet de dualité. Dans ce chapitre, nous nous intéressons aux configurations quadratiques donc par définition, le nombre de fonctions de base est fini. Nous utilisons, par conséquent, la dualité dans le cas fini. L'espace des vecteurs duaux s'identifie aux vecteurs de $\mathbb{R}^{\mathbb{P}}$ et le cône dual s'identifie aux vecteurs positifs de $\mathbb{R}^{\mathbb{P}}_+$. Concernant les problèmes de $\mathrm{dom}(v_{\ell(a)})$ et $\mathrm{dom}(v_{\ell(\iota)})$ évoqués dans le chapitre 5, on peut s'en affranchir en posant pour $\lambda \in \mathbb{R}^{\mathbb{P}}_+$ $\lambda(p) = 0$ pour tout $p \notin \mathrm{dom}(v_{\ell(a)})$ (ou $\mathrm{dom}(v_{\ell(\iota)})$). De plus, nous pouvons conserver la même dualité tout au long de l'algorithme. On notera simplement $F^{\mathcal{R}}$ pour la fonction sémantique relâchée.

La fonction sémantique relâchée des constantes et des unions n'est pas modifiée par rapport au chapitre 5. La fonction sémantique relâchée d'une constante correspond à C^σ où C est un sous-ensemble de $\mathbb{R}^d$. La fonction sémantique relâchée d'une union correspond au supremum des deux fonctions coordonnées apparaissant dans l'union. Nous redéfinissons la fonction sémantique relâchée pour les affectations et les intersections. Celles-ci sont identiques aux fonctions sémantiques relâchées du chapitre 5 excepté que l'on considère la dualité dans le cas fini, c'est-à-dire que nous considérons $\mathbb{R}^{\mathbb{P}}$ que nous dualisons avec lui-même.

4. http://pop-art.inrialpes.fr/~bjeannet/bjeannet-forge/interproc/index.html

Définition 6.3 (*Fonction sémantique relâchée dans une configuration quadratique*)

Soit $a \in \mathbb{A}$. La coordonnée de la fonction sémantique relâchée $F_a^{\mathcal{R}}$ vérifie :

$$F_a^{\mathcal{R}}(v) = \inf_{\lambda \in \mathbb{R}_+^{\mathbb{P}}} \sup_{x \in \mathbb{R}^d} \mathbf{e}_{Tx} + \sum_{q \in \mathbb{P}} \lambda(q)(v_{\ell(a)}(q) - q(x))$$

Soit $i \in \mathbb{I}$. La coordonnée de la fonction sémantique relâchée $F_i^{\mathcal{R}}$ vérifie :

$$F_i^{\mathcal{R}}(v) = \inf_{\substack{\lambda \in \mathbb{R}_+^{\mathbb{P}} \\ \mu \in \mathbb{R}_+}} \sup_{x \in \mathbb{R}^d} \mathbf{e}_{Tx} + \sum_{q \in \mathbb{P}} \lambda(q)(v_{\ell(i)}(q) - q(x)) - \mu r(x)$$

Exemple 6.4 (Suite de l'exemple 6.1)
Nous rappelons que la matrice A est orthogonale et que la configuration utilisée est la suivante $\mathbb{P} = \{p_1 : x \mapsto -\|x\|^2,\ p_2 : x \mapsto \|x\|^2\}$. La fonction sémantique abstraite du programme de l'exemple 6.1 est la fonction :

$$\begin{array}{rcl} F_1^\sharp(v) & = & (-1, 1) \\ F_2^\sharp(v) & = & \sup\limits_{x \in v_1^{\mathbf{s}}} \mathbf{e}_{Ax} \end{array}$$

La fonction sémantique relâchée est la fonction :

$$\begin{array}{rcl} F_1^{\mathcal{R}}(v) & = & (-1, 1) \\ F_2^{\mathcal{R}}(v) & = & \inf\limits_{\lambda(p_1), \lambda(p_2), \mu \geq 0} \lambda(p_1)v_1(p_1) + \lambda(p)v_1(p_2) + \sup\limits_{x \in \mathbb{R}^d} \mathbf{e}_{Ax} + \lambda(p_1)x \cdot x - \lambda(p_2)x \cdot x \end{array}$$

A la première coordonnée, le vecteur $(-1, 1)$ désigne la fonction qui vaut -1 pour $p = x \mapsto -\|x\|^2$ et 1 pour $p = x \mapsto -\|x\|^2$.

Exemple 6.5 (Suite de l'exemple 6.2)
Soit A la matrice :

$$\begin{pmatrix} 1 & 0.01 \\ -0.01 & 0.99 \end{pmatrix}$$

Nous rappelons que les fonctions de base sont $p_x : (x, v) \mapsto x^2$, $p_v : (x, v) \mapsto v^2$ et $p_L : (x, v) \mapsto 3v^2 + 2x^2 + 2xv$, et donc la configuration est $\mathbb{P} = \{p_x, p_v, p_L\}$. La fonction sémantique abstraite du programme de l'exemple 6.2 est la fonction :

$$\begin{array}{rcl} F_1^\sharp(v) & = & (1, 1, 7) \\ F_2^\sharp(v) & = & (v_1^{\mathbf{s}} \cup v_3^{\mathbf{s}})^\sigma \\ F_3^\sharp(v) & = & \sup\limits_{x \in v_2^{\mathbf{s}}} \mathbf{e}_{Ax} \end{array}$$

La fonction sémantique relâchée est la fonction :

$$\begin{array}{rcl} F_1^{\mathcal{R}}(v) & = & (1, 1, 7) \\ F_2^{\mathcal{R}}(v) & = & \sup(v_1, v_3) \\ F_3^{\mathcal{R}}(v) & = & \inf\limits_{\lambda(p_x), \lambda(p_v), \lambda(p_L) \geq 0} \sup\limits_{x \in \mathbb{R}^d} \sum\limits_{p \in \{p_x, p_v, p_L\}} \lambda(p)v_2(p) - p(x) + \mathbf{e}_{Ax} \end{array}$$

Le vecteur $(1,1,7)$ désigne la fonction qui vaut 1 si $p = p_x$, 1 si p_v et 7 si $p = p_L$.

Les matrices :

$$\begin{pmatrix}1&0\\0&0\end{pmatrix}, \begin{pmatrix}0&0\\0&1\end{pmatrix} \text{ et } \begin{pmatrix}2&1\\1&3\end{pmatrix}$$

ont des coefficients hors diagonaux sont positifs ou nuls de plus, un simple calcul montre que les matrices :

$$\begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}1&0.01\\-0.01&0.99\end{pmatrix}\begin{pmatrix}1&0\\0&0\end{pmatrix}, \begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}1&0.01\\-0.01&0.99\end{pmatrix}\begin{pmatrix}0&0\\0&1\end{pmatrix} \text{ et}$$

$$\begin{pmatrix}2&1\\1&3\end{pmatrix}\begin{pmatrix}1&0.01\\-0.01&0.99\end{pmatrix}\begin{pmatrix}2&1\\1&3\end{pmatrix},$$

ont également des coefficients hors diagonaux positifs ou nuls.

On conclut en utilisant les résultats de Kim et Kojima [KK03], que $F_3^\sharp(v)$ coïncide avec $F_3^{\mathcal{R}}(v)$ dès qu'il existe une matrice X définie positive telle que : $X_{11} = 1$ et

$$\langle \begin{pmatrix}-v_2(p_x)&0&0\\0&1&0\\0&0&0\end{pmatrix}, X\rangle_F < 0$$
$$\langle \begin{pmatrix}-v_2(p_v)&0&0\\0&0&0\\0&0&1\end{pmatrix}, X\rangle_F < 0$$
$$\langle \begin{pmatrix}-v_2(p_L)&0&0\\0&2&1\\0&1&3\end{pmatrix}, X\rangle_F < 0$$

Cette condition permet d'assurer que Val($ShorD$) et $F_3^{\mathcal{R}}(v)$ sont égales et les résultats de Kim et Kojima assurent que Val($ShorD$) et $F_3^\sharp$ sont égales. Par exemple, si $v_2(p_x) > 0$, $v_2(p_v) > 0$ et $v_2(p_L) > 3$, la fonction sémantique abstraite coïncide avec la fonction sémantique relâchée puisqu'en posant $\alpha = \min\{v_2(p_x), v_2(p_v), \frac{v_2(p_L)-3}{2}\} > 0$, la matrice diagonale X dont la diagonale est $(1,\alpha,\alpha)$ satisfait les conditions.

Exemple 6.6 (Suite de l'exemple 6.3)
Pour l'exemple 6.3, nous introduisons la matrice suivante : $T = \begin{pmatrix}1&0\\1&1\end{pmatrix}$ et nous posons $b^T = (1,1)$. Dans cet exemple, nous avons considéré la configuration munie d'une unique fonction : $\mathbb{P} = \{(i,j) \mapsto -\frac{i(i+1)}{2} + j\}$. La fonction sémantique abstraite $F^\sharp$ du programme 6.3 est la fonction suivante :

$$\begin{array}{lcl} F_1^\sharp(v) & = & 0 \\ F_2^\sharp(v) & = & (v_1^{\mathbf{s}} \cup v_3^{\mathbf{s}})^\sigma \\ F_3^\sharp(v) & = & \sup\limits_{\substack{(i,j)\in v_2^{\mathbf{s}} \\ i\leq 42}} \mathbf{e}_{T(i,j)^T+b} \end{array}$$

La fonction sémantique relâchée est la fonction :

$$\begin{array}{lcl} F_1^{\mathcal{R}}(v) & = & 0 \\ F_2^{\mathcal{R}}(v) & = & \sup(v_1, v_3) \\ F_3^{\mathcal{R}}(v) & = & \inf\limits_{\lambda,\mu\geq 0} \lambda v_2 \sup\limits_{(i,j)\in\mathbb{R}^2} \mathbf{e}_{T(i,j)^T+b} + 42\mu - \mu i + \lambda p(i,j) \end{array}$$

La $\mathbb{P}$-enveloppe convexe s'évalue par dualité et nous notons pour $w \in \overline{\mathbb{R}}^{\mathbb{P}}$, $\mathcal{R}\text{-}\,\mathrm{vex}_{\mathbb{P}}(w)$ la relâchée de $\mathbb{P}$-enveloppe convexe.

Définition 6.4 (*P-enveloppe convexe dans une configuration quadratique*)

Soit $w \in \overline{\mathbb{R}}^{\mathbb{P}}$. La $\mathbb{P}$-enveloppe convexe relâchée de w, $\mathcal{R}\text{-}\,\mathrm{vex}_{\mathbb{P}}(w)$, est définie par :

$$\mathcal{R}\text{-}\,\mathrm{vex}_{\mathbb{P}}(w) = \inf_{\lambda \in \mathbb{R}_+^{\mathbb{P}}} \sup_{x \in \mathbb{R}^d} \mathbf{e}_x + \sum_{q \in \mathbb{P}} \lambda(q)(w(q) - q(x))$$

Nous commençons par remarquer que les fonctions $\mathbb{V}_a$ et $\mathbb{V}_i$ de la propriété 5.1 sont simples à calculer dans le cas des configurations quadratiques puisqu'il s'agit d'un problème quadratique non contraint. Pour le calcul de cette fonction il suffit d'étudier la matrice associée à une certaine forme quadratique que nous introduirons plus loin. La solution du problème quadratique induit par les fonctions $\mathbb{V}_a$ et $\mathbb{V}_i$ est décrit par la proposition 6.1.

Nous associons, à toute fonction de base $p \in P$, sa matrice symétrique A_p et son vecteur b_p. Soit une affectation T de matrice A et de vecteur b. Si l'affectation n'est pas une opération interne de boucle ou de branchement conditionnel alors, pour $\lambda \in \mathbb{R}_+^{\mathbb{P}}$ et une fonction de base $p \in \mathbb{P}$ on pose :

$$\begin{aligned} \mathcal{A}_p(\lambda) &= A_{p\circ T} - \sum_{q \in P} \lambda(q) A_q, \\ \mathcal{B}_p(\lambda) &= b_{p\circ T} - \sum_{q \in P} \lambda(q) b_q^T, \quad . \\ \mathcal{C}_p(\lambda, \mu) &= c_{p\circ T} \end{aligned}$$

Supposons maintenant que l'affectation provienne d'une boucle ou d'un branchement conditionnel et supposons que le test est une fonction $x \mapsto x^T A x + b^T x + c$ alors pour $\lambda \in \mathbb{R}_+^{\mathbb{P}}$, $\mu \in \mathbb{R}_+$ une fonction de base $p \in \mathbb{P}$ on pose :

$$\begin{aligned} \mathcal{A}_p(\lambda, \mu) &= A_{p\circ T} - \sum_{q \in P} \lambda(q) A_q - \mu A, \\ \mathcal{B}_p(\lambda, \mu) &= b_{p\circ T} - \sum_{q \in P} \lambda(q) b_q^T - \mu_i b, \\ \mathcal{C}_p(\lambda, \mu) &= c_{p\circ T} - \mu c \ . \end{aligned}$$

Dans le cadre des configurations quadratiques, la fonction $\mathbb{V}$ est définie par :

$$\begin{aligned} [\mathbb{V}_a(\lambda)](p) &= \mapsto \sup_{x \in \mathbb{R}^d} x^T \mathcal{A}_p(\lambda) x + \mathcal{B}_p(\lambda)^T x + \mathcal{C}_p(\lambda) \\ [\mathbb{V}_i(\lambda, \mu)](p) &= \mapsto \sup_{x \in \mathbb{R}^d} x^T \mathcal{A}_p(\lambda, \mu) x + \mathcal{B}_p(\lambda, \mu)^T x + \mathcal{C}_p(\lambda, \mu) \end{aligned}$$

Pour rappel, la décomposition en valeurs singulières d'une matrice non nulle A est la matrice $U \Sigma V^T$ où Σ est la matrice diagonale des racines carrées des valeurs propres de la matrice symétrique positive AA^T ordonnées de manière décroissante et U, V sont respectivement la matrice orthogonale des vecteurs propres de AA^T et la matrice orthogonale des vecteurs propres de $A^T A$. La décomposition en valeurs singulières permet de définir la pseudo-inverse (ou inverse de Moore-Penrose) d'une matrice A. La pseudo-inverse d'une matrice non nulle A est notée $A^\dagger$ et est définie comme la matrice :

$$V \begin{pmatrix} \Sigma_0^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T$$

où Σ_0^{-1} est l'inverses de la matrice diagonale valeurs propres non nulles de AA^T ordonnées de manière décroissante. La pseudo-inverse de la matrice nulle est la matrice nulle.

Proposition 6.1 (*Supremum d'une forme quadratique*)

Le calcul des fonctions $\mathbb{V}_a$ et $\mathbb{V}_\imath$ est facile puisque :

1. Si $\lambda \in \mathbb{R}_+^{\mathbb{P}}$ vérifie $\mathcal{A}_p(\lambda) \preceq 0$ et $\mathcal{B}_p(\lambda) \in \mathrm{Im}(\mathcal{A}_p(\lambda))$ alors :

$$[\mathbb{V}_a(\lambda)](p) = -\frac{1}{4}\mathcal{B}_p(\lambda)^T \mathcal{A}_p(\lambda)^\dagger \mathcal{B}_p(\lambda) + \mathcal{C}_p(\lambda).$$

2. Si $(\lambda, \mu) \in \mathbb{R}_+^{\mathbb{P}} \times \mathbb{R}_+$ vérifie $\mathcal{A}_p(\lambda, \mu) \preceq 0$ et $\mathcal{B}_p(\lambda, \mu) \in \mathrm{Im}(\mathcal{A}_p(\lambda, \mu))$ alors :

$$[\mathbb{V}_\imath(\lambda, \mu)](p) = -\frac{1}{4}\mathcal{B}_p(\lambda, \mu)^T \mathcal{A}_p(\lambda, \mu)^\dagger \mathcal{B}_p(\lambda, \mu) + \mathcal{C}_p(\lambda, \mu).$$

3. Pour tout $\lambda \in \mathbb{R}_+^{\mathbb{P}}$ tel que $\mathcal{A}_p(\lambda) \notin S_n^-$ ou $\mathcal{B}_p(\lambda) \notin \mathrm{Im}(\mathcal{A}(\lambda))$ et pour tout $(\lambda, \mu) \in \mathbb{R}_+^{\mathbb{P}} \times \mathbb{R}_+$ tel que $\mathcal{A}_p(\lambda, \mu) \notin S_n^-$ ou $\mathcal{B}_p(\lambda, \mu) \notin \mathrm{Im}(\mathcal{A}_p(\lambda, \mu))$ alors :

$$[\mathbb{V}_a(\lambda)](p) = [\mathbb{V}_i(\lambda, \mu)](p) = +\infty$$

Démonstration

Une démonstration de ce résultat peut être trouvée dans [BV04, Section 4.2.3]. ∎

De plus, par la propriété 5.1, on peut également écrire la fonction sémantique relâchée comme l'infimum de fonctions affines sur $\mathbb{R}^{\mathbb{P}}$.

Lemme 6.1 (*Fonction sémantique relâchée*)

Pour tout $a \in \mathbb{A}$, pour tout $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$, on a :

$$F_a^{\mathcal{R}}(v) = \inf_{\lambda \in \mathbb{R}_+^{\mathbb{P}}} \sum_{p \in \mathbb{P}} \lambda(p) v_{\ell(a)}(p) + \mathbb{V}_a(\lambda)$$

Pour tout $\imath \in \mathbb{A}$, pour tout $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$, on a :

$$F_\imath^{\mathcal{R}}(v) = \inf_{(\lambda,\mu) \in \mathbb{R}_+^{\mathbb{P}} \times \mathbb{R}_+} \sum_{p \in \mathbb{P}} \lambda(p) v_{\ell(\imath)}(p) + \mathbb{V}_\imath(\lambda, \mu)$$

Soit une fonction de base $p \in \mathbb{P}$, on rappelle que p est de la forme $x^T A_p x + b_p^T x$. On rappelle qu'à une forme quadratique f, on associe $M(f)$ la matrice définie par :

$$\begin{pmatrix} c_f & (1/2)b_f^T \\ (1/2)b_f & A_f \end{pmatrix}$$

On introduit également la matrice, pour y réel, la matrice $N(y)$ définie par $N(y)$ vaut y sur l'entrée $(1, 1)$ et 0 ailleurs.

Nous avons supposé que les fonctions $p \circ T$ sont quadratiques quelque soit la fonction de base $p \in \mathbb{P}$ (hypothèse (HQ)). Ainsi, en utilisant la relaxation de Shor rappelée à la section 6.1.2, on peut évaluer la fonction sémantique relâchée en résolvant un problème SDP.

Proposition 6.2 (*Calcul de la fonction sémantique relâchée pour les affectations*)

Soient $a \in \mathbb{A}$ et $p \in \mathbb{P}$. Soit $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(a)} > -\infty$ et $\mathrm{dom}(v_{\ell(a)}) \neq \emptyset$. La fonction sémantique relâchée $[F_a^{\mathcal{R}}(v)](p)$ pour la coordonnée a, au vecteur v et pour la fonction de base p s'évalue en résolvant le problème d'optimisation :

$$\begin{array}{ll} \text{Min} & \eta \\ \text{s. c} & M(p \circ T) + N(-\eta) + \sum\limits_{q \in \mathrm{dom}(v_{\ell(a)})} \lambda(q)[N(v_{\ell(a)}(q)) - M(q)] \preceq 0 \\ & \lambda(q) \geq 0 \ \forall q \in \mathrm{dom}(v_{\ell(a)}), \lambda(q') = 0 \ \forall q' \in \mathrm{dom}(v_{\ell(a)}) \eta \in \mathbb{R} \end{array}$$

Puisque le calcul de la $\mathbb{P}$-enveloppe convexe est un cas particulier d'affectation, on peut évaluer celle-ci par la relaxation de Shor et on obtient le résultat suivant :

Corollaire 6.1 (*Calcul de la $\mathbb{P}$-enveloppe convexe relâchée*)

Soit $w \in \overline{\mathbb{R}}^{\mathbb{P}}$ tel que $w(p) > -\infty$ pour tout $p \in \mathbb{P}$ et $\mathrm{dom}(w) \neq \emptyset$. Soit $p \in \mathbb{P}$, on peut calculer la P-enveloppe convexe relâchée que nous notons ici $\mathcal{R}\text{-}\,\mathrm{vex}_{\mathbb{P}}(w)$ par le programme SDP suivant :

$$\begin{array}{ll} \text{Min} & \eta \\ \text{s. c} & M(p) + N(-\eta) + \sum\limits_{q \in \mathrm{dom}(v_{\ell(a)})} \lambda(q)[N(v_{\ell(a)}(q)) - M(q)] \preceq 0 \\ & \lambda(q) \geq 0 \ \forall q \in \mathrm{dom}(w), \lambda(q') = 0 \ \forall q' \in \mathrm{dom}(w) \eta \in \mathbb{R} \end{array}$$

La fonction sémantique relâchée pour intersection s'évalue également grâce à la relaxation de Shor. On ajoute uniquement une matrice $M(r)$ ou r est la fonction quadratique définissant le test ainsi qu'un multiplicateur μ associé à la contrainte $r(x) \leq 0$.

Proposition 6.3 (*Calcul de la fonction sémantique relâchée pour les intersections*)

Soient $\imath \in \mathbb{I}$ et $p \in \mathbb{P}$. Soit $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ tel que $v_{\ell(\imath)} > -\infty$ et $\mathrm{dom}(v_{\ell(\imath)}) \neq \emptyset$. La fonction sémantique relâchée $[F_i^{\mathcal{R}}(v)](p)$ pour la coordonnée i, au vecteur v et pour la fonction de base p s'évalue en résolvant le problème d'optimisation :

$$\begin{array}{ll} \text{Min} & \eta \\ \text{s. c} & M(p \circ T) + N(-\eta) + \sum\limits_{q \in \mathrm{dom}(v_{\ell(\imath)})} \lambda(q)[N(v_{\ell(\imath)}(q)) - M(q)] - \mu M(r) \preceq 0 \\ & \lambda(q) \geq 0 \ \forall q \in \mathrm{dom}(v_{\ell(\imath)}), \lambda(q') = 0 \ q' \notin \mathrm{dom}(v_{\ell(\imath)}) \ \mu \in \mathbb{R}_+, \ \eta \in \mathbb{R} \end{array}$$

On peut néanmoins soulever le problème de réalisabilité du problème SDP de la proposition 6.2 lorsque l'affectation T est quadratique et les fonctions de base $p \in \mathbb{P}$ sont linéaires. Les matrices $N(v_{\ell(a)}(q)) - M(q)$ n'influence pas la partie purement quadratique de la somme

de matrices :

$$M(p \circ T) + N(-\eta) + \sum_{q \in \mathrm{dom}(v_{\ell(a)})} \lambda(q)[N(v_{\ell(a)}(q)) - M(q)]$$

qui uniquement contenue dans la matrice $M(p \circ T)$, il faut donc trouver une configuration $\mathbb{P}$ telle que tout $p \in \mathbb{P}$, $M(p \circ T)$ soit une matrice semi-définie négative.

Exemple 6.7 (Affectation non-linéaire)
Nous considérons le programme écrit en arithmétique quadratique de la figure 6.4.

```
x  = [0,10];
y  =      1;  [1]
xn = -3*x*x-y*y;
yn  = -y*y+x*x;
x  = xn[2]  }
```

Figure 6.4 – Un simple programme écrit en arithmétique polynomiale

Nous nous intéressons à la configuration suivante $P = \{p_1,\ p_2\}$, avec $p_1 : (x, y) \mapsto x + y$ et $p_2 : (x, y) \mapsto x - y$. Nous définissons la fonction T par :

$$(x, y) \mapsto T(x, y) = \begin{pmatrix} -3x^2 - y^2 \\ -y^2 + x^2 \end{pmatrix}$$

ce qui est l'affectation non-linéaire du programme 6.4. On voit facilement au point de contrôle [1] que $x + y \leq 11$ et que $x - y \leq 9$. La fonction sémantique abstraite est, pour $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^2$ et p_1 :

$$[F_1^\sharp(v)](p_1) = 11\}$$
$$[F_2^\sharp(v)](p_1) = \sup_{(x,y)\in(v_1)^\star} (p_1 \circ T)(x, y)$$

et pour p_2 :

$$[F_1^\sharp(v)](p_2) = 9\}$$
$$[F_2^\sharp(v)](p_2) = \sup_{(x,y)\in(v_1)^\star} (p_2 \circ T)(x, y)$$

Par ailleurs, par un calcul simple, on déduit qu'au point de contrôle [2], $xn + yn = -2x^2 - 2y^2$. On obtient ainsi la fonction sémantique au point de contrôle 2, toujours pour $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^2$ et pour la fonction de base p_1 prend la valeur suivante :

$$\inf_{\substack{\lambda(q)\geq 0 \\ \forall\, q\in\mathbb{P}_{\geq 0}}} \sup_{(x,y)\in\mathbb{R}^2} \sum_{q\in\mathbb{P}} \lambda(q) w_1(q) + (x, y) \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix} (x, y)^T - \sum_{q\in\mathbb{P}} \lambda(q) q(x, y) \tag{6.3}$$

En introduisant les matrices suivantes :

$$M(p_1) = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 \\ 0.5 & 0 & 0 \end{pmatrix},\ M(p_2) = \begin{pmatrix} 0 & 0.5 & -0.5 \\ 0.5 & 0 & 0 \\ -0.5 & 0 & 0 \end{pmatrix},$$

$$\text{et } M(p_1 \circ T) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{pmatrix}$$

on peut réécrire (6.3) comme le problème SDP de l'équation (6.4) :

$$[F_2^{\mathcal{R}}(w)](p_1) = \underset{\substack{\lambda(p)\geq 0 \\ \forall p \in P \\ \eta \in \mathbb{R}}}{\text{Min}} \ \eta \ \text{ s.t. } \ M(p_1 \circ T) + \eta N(-1) + \sum_{q \in P} \lambda(q)[N(w_1(q)) - M(q)] \preceq 0 \quad (6.4)$$

Finalement en utilisant une implémentation `Matlab`[5], `Yalmip` [Lö04] and `SeDuMi` [Stu99], nous trouvons au point de contrôle 2, $[F_2^{\mathcal{R}}(w)](p_1) = -3.2018e-09 \simeq 0$ ce qui signifie, qu'au point de contrôle [2], $x \leq -y$ et de $[F_2^{\mathcal{R}}(w)](p_2) = -1.0398e-09 \simeq 0$, on déduit qu'au point de contrôle [2], $x \leq y$. L'invariant trouvé $\{(x,y) \in \mathbb{R}^2 \mid x \leq -y, x \leq y\}$ est un ensemble non borné. On peut cependant retrouver un ensemble borné en calculant l'invariant grâce à l'arithmétique par intervalles puis de prendre les intersections des deux invariants.

Dans le cas des intersections, la contrainte induite par le test peut assurer la réalisabilité du problème du problème SDP de la proposition 6.3, c'est-à-dire, l'existence d'un réel positif μ et d'une fonction λ telle que la somme de matrices :

$$M(p \circ T) + N(-\eta) + \sum_{q \in \text{dom}(v_{\ell(\iota)})} \lambda(q)[N(v_{\ell(\iota)}(q)) - M(q)] - \mu M(r)$$

soit semi-définie négative.

Exemple 6.8 (Affectation et test quadratiques)
Considérons le programme de la figure 6.5 écrit en arithmétique polynomiale.

```
x = [0,10];
y =      1;
u =      0; [1]
if (y*y+x*x-2<=0){
    u=x;
    x = 3-y*y;
    y = u-1;[2]
    }
```

Figure 6.5 – Un progamme simple écrit en arithmétique polynomiale

Nous introduisons la fonction quadratique $r : (x,y) \mapsto y^2 + x^2 - 2$ qui représente. Nous définissons la fonction T par :

$$(x,y) \mapsto T(x,y) = \begin{pmatrix} 3 - y^2 \\ x - 1 \end{pmatrix} .$$

5. Matlab est une marque déposée de the MathWorks,Inc.

Cette fonction représente l'affectation du programme de la figure 6.5. Nous considérons le domaine des intervalles : $\mathbb{P} = \{\underline{x}, -\underline{x}, \underline{y}, -\underline{y}\}$ où $\underline{x} : (x, y) \mapsto x$, $\underline{y} : (x, y) \mapsto y$. LA fonction sémantique abstraite, pour une fonction $w \in (\overline{\mathbb{R}}^{\mathbb{P}})^2$ et $p \in \mathbb{P}$ est définie par :

$$[F_1^\sharp(w)](p) = \{\underline{x}(x,y) \leq 10,\ -\underline{x}(x,y) \leq 0, \underline{y}(x,y) \leq 1,\ -\underline{y}(x,y) \leq -1\}^\sigma(p)$$
$$[F_2^\sharp(w)](p) = \sup_{\substack{(x,y)\in w_1^\star \\ r(x,y)\leq 0}} p(T(x,y))$$

La fonction sémantique relâchée au point de contrôle [2], pour $w \in (\overline{\mathbb{R}}^{\mathbb{P}})^2$ et pour la fonction de base $\underline{x}$ est définie par :

$$\inf_{\substack{\lambda(p)\geq 0 \\ \forall\, p\in\mathbb{P} \\ \mu\in\mathbb{R}_+}} \sup_{(x,y)\in\mathbb{R}^2} \sum_{q\in\mathbb{P}} \lambda(q) w_1(q) + (x,y) \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix} (x,y)^T - \sum_{q\in\mathbb{P}} \lambda(q) q(x,y) + 3 - \mu r(x,y) \quad (6.5)$$

En prenant les matrices $M(\underline{x})$, $M(-\underline{x})$, $M(\underline{y})$ et $M(-\underline{y})$ definies par :

$$M(\underline{x}) = \begin{pmatrix} 0 & 0.5 & 0 \\ 0.5 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ M(\underline{y}) = \begin{pmatrix} 0 & 0 & 0.5 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0 \end{pmatrix},$$
$$M(-\underline{x}) = -M(\underline{x}) \text{ et} M(-\underline{y}) = -M(\underline{y})\ .$$

On considère également les matrices suivantes :

$$M(r) = \begin{pmatrix} -2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ \text{ et } M(\underline{x}\circ T) = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

on peut réécrire (6.5) comme le problème SDP de l'équation (6.6) :

$$[F_2^{\mathcal{R}}(w)](\underline{x}) = \min_{\substack{\lambda(p)\geq 0 \\ \forall\, p\in\mathbb{P} \\ \mu\in\mathbb{R}_+ \\ \eta\in\mathbb{R}}} \eta \text{ s.t. } M(\underline{x}\circ T) + \eta N(-1) - \mu M(r) + \sum_{q\in\mathbb{P}} \lambda(q)[N(w_1(q)) - M(q)] \preceq 0 \quad (6.6)$$

On trouve utilisant à nouveau, `Yalmip`, `SeduMi` et `Matlab`, $[F_2^{\mathcal{R}}(w)](\underline{x}) = 2$. Pour la borne inférieure de "$3 - y^2$", nous trouvons $[F_2^{\mathcal{R}}(w)](-\underline{x}) = -1$. Pour la variable y au point de contrôle [2], nous trouvons que $[F_2^{\mathcal{R}}(w)](\underline{y}) = -3.4698e - 09 \simeq 0$ et que $[F_2^{\mathcal{R}}(w)](-\underline{y}) = 1$. Nous remarquons que $[F_2^{\mathcal{R}}(w)](\underline{y}) \simeq 0$ ce qui signifie qu'au point de contrôle [2], les valeurs de de "$x - 1$" sont plus petites ou égales à 0. En effet, la relaxation de Shor détecte que le test est vérifié ssi $1 + x^2 - 2 \leq 0$ ce qui est équivalent à $x^2 \leq 1$ et par conséquent "$x - 1$" est majoré par 0.

On conclut cette sous-section par la théorème suivant :

Théorème 6.1 (*Evaluation de la fonction sémantique relâchée*)

On peut évaluer la fonction sémantique relâchée pour les affectations et les intersections ainsi que la $\mathbb{P}$-enveloppe convexe en résolvant un problème d'optimisation semi-definie grâce à la relaxation de Shor.

6.2.2 Itérations sur les politiques dans les configurations quadratiques

En utilisant le lemme 6.1, on peut montrer que la fonction sémantique relâchée est croissante dans le treillis complet $(\overline{\mathbb{R}}^{\mathbb{P}})^n$. En effet, lorsque l'on fixe des multiplicateurs de Lagrange, la fonction $v \mapsto F(\pi, v)$ est croissante dans $(\overline{\mathbb{R}}^{\mathbb{P}})^n$ et donc la fonction sémantique relâchée est l'infimum de fonctions croissantes, elle est donc croissante. Ces deux fonctions possèdent donc chacune un plus petit point fixe. Par le théorème de sur-approximation sûre, le plus petit point fixe de $F^{\mathcal{R}}$ majore le plus petit point fixe de $F^{\sharp}$. Quand on restreint les fonctions $v \mapsto F(\pi, v)$ sur $(\mathbb{R}^{\mathbb{P}})^n$ ces fonctions sont affines et le plus petit point fixe est la solution du problème de minimisation linéaire :

$$\text{Min}\{\sum_{i=1}^{d}\sum_{p\in\mathbb{P}} v_i(p) \mid F(\pi, v) \leq v\}$$

Il reste à bien choisir la politique initiale. Pour la déterminer, deux conditions doivent être satisfaite :

1. Pour tout $p \in P$, pour tout $a \in \mathbb{A}$, pour tout $\imath \in \mathbb{I}$:

$$[\mathbb{V}_a(\pi_a^0(p))](p) < +\infty \text{ et } [\mathbb{V}_i(\pi_i^0(p))](p) < +\infty$$

2. L'ensemble :

$$\{v \in (\mathbb{R}^{\mathbb{P}})^n \mid F(\pi^0, v) \leq v\} \neq \emptyset$$

Grâce à la proposition 6.1, le premier point est vérifié si $\mathcal{A}_p(\pi_a^0(p)) \preceq 0$ et $\mathcal{B}_p(\pi_a^0(p)) \in \text{Im}(\mathcal{A}_p(\pi_a^0(p)))$ et $\mathcal{A}_p(\pi_i^0(p)) \preceq 0$ et $\mathcal{B}_p(\pi_i^0(p)) \in \text{Im}(\mathcal{A}_p(\pi_a^0(p)))$. La condition d'appartenance à l'image des matrices peut être affranchie si $\mathcal{A}_p(\pi_a^0(p))$ et $\mathcal{A}_p(\pi_i^0(p)) \preceq 0$ sont définie négative (et donc inversibles). Le deuxième point est plus difficile à vérifier numériquement et extraire les politiques π^0 compatibles.

On peut définir une itération de Kleene puisque la fonction sémantique relâchée est croissante sur $(\overline{\mathbb{R}}^{\mathbb{P}})^n$. La méthode décrite ici, correspond à un équivalent de la version théorique classique de la méthode de Kleene donnée par l'algorithme 1.

Algorithme 6: Itération de Kleene dans le domaine des sous-niveaux

$v^0 = \bot$;
tant que $F^{\mathcal{R}}(v^k) \neq v^k$ **faire**
  $v^{k+1} = F^{\mathcal{R}}(v^k)$;
  $k = k + 1$;

On peut utiliser l'itération de Kleene pour définir la politique initiale. En théorie, l'itération de Kleene 6 puisque $v \mapsto F^{\mathcal{R}}(v)$ est croissante sur $(\overline{\mathbb{R}}^{\mathbb{P}})^n$, la suite $(v^k)_{k\geq 0}$ converge vers le plus petit point fixe de $F^{\mathcal{R}}$ lorsque $F^{\mathcal{R}}$ est Scott-continue (voir théorème de 2.2). Cependant la convergence peut être très lente. On peut également définir une technique d'accélération simple pour sur-approximer le plus petit point fixe de $F^{\mathcal{R}}$ comme par exemple, après un certain nombre d'iterations et pendant quelques itérations, on peut arrondir par excès avec une précision décroissante (voir l'élargissement [GPBG08]). On peut réduire la perte de précision de l'invariant trouvé par l'itération de Kleene accéléré en prenant la P-enveloppe convexe ou au moins la relaxation de celle-ci juste après la dernière itération. Pour trouver une politique initiale à l'itération sur les politiques, on peut itérer l'algorithme de Kleene combiné à une

technique d'accélération jusqu'à l'atteinte d'un post point fixe $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$. La configuration $\mathbb{P}$ étant finie, on peut affaiblir la propriété de sélection en la remplaçant par la contrainte de qualification de Slater 5.8 qui nous permet de donner une condition simple d'existence d'une politique π telle que $F^{\mathcal{R}}(v) = F(\pi, v)$. Nous notons $\mathcal{FS}$, l'ensemble des éléments v de $(\overline{\mathbb{R}}^{\mathbb{P}})^n$ tels qu'il existe $x \in \mathbb{R}^d$, $v_{\ell(a)} - \mathbf{e}_x > 0$ et il existe $y \in \mathbb{R}^d$ tel que $v_{\ell(\imath)} - \mathbf{e}_y > 0$ et $r_\imath(y) < 0$ pour tout $a \in \mathbb{A}$ et $\imath \in \mathbb{I}$. Ainsi si l'itération de Kleene accélérée fournit un élément $v \in \mathcal{FS}$ alors, il existe une politique π^0 qui détermine une politique initiale.

Puisqu'on peut considérer la même dualité au cours de l'algorithme, on peut simplifier l'algorithme sur les politiques énoncé au chapitre 5. En considérant uniquement l'algorithme 7.

Algorithme 7: Itérations sur les politiques dans les configurations finies

Choisir $\pi^0 \in \Pi$ telle que pour tout $j \in \mathbb{A} \cup \mathbb{I}$, $\mathbb{V}_j^{\pi_j^0} < +\infty$;
Calculer pour tout $j \in \mathbb{A} \cup \mathbb{I}$, $\mathbb{V}_j(\pi_j^0) = \{[\mathbb{V}_j(\pi_j^0)](q)\}_{q \in \mathbb{P}}$;
Calculer le plus petit point fixe v^0 dans $(\overline{\mathbb{R}}^{\mathbb{P}})^n$ de $F(\pi^0, \cdot)$;
Calculer $w^0 = \mathcal{R}\text{-}\operatorname{vex}_{\mathbb{P}}(v^0)$;
Evaluer $F^{\mathcal{R}}(w^0)$;
$k = 0$;
tant que $F^{\mathcal{R}}(w^k) \neq w^k$ **faire**
  si $w^k \notin \mathcal{FS}$ **alors**
    retourner w^k
  sinon
    Prendre π^{k+1} tel que $F^{\mathcal{R}}(w^k) = F(\pi^{k+1}, w^k)$;
    Calculer $j \in \mathbb{A} \cup \mathbb{I}$, $\mathbb{V}_j(\pi_j^{k+1}) = \{[\mathbb{V}_j(\pi_j^{k+1})](q)\}_{q \in \mathbb{P}}$;
    Calculer le plus petit point fixe v^{k+1} dans $(\overline{\mathbb{R}}^{\mathbb{P}})^n$ de $F(\pi^{k+1}, \cdot)$;
    Calculer $w^{k+1} = \mathcal{R}\text{-}\operatorname{vex}_{\mathbb{P}}(v^{k+1})$;

Cet algorithme construit une suite décroissante de fonctions v^k :

Théorème 6.2

Les assertions suivantes sont vraies :

1. $F^{\mathcal{R}}(v^l) \neq v^l \implies F^{\mathcal{R}}(v^l) < v^l$.
2. La suite v^l calculée par Algorithm 7 est strictlement décroissante.

Démonstration

1. Soit $l \in \mathbb{N}$. Nous supposons qu'il existe $l > 0$ tel que $F^{\mathcal{R}}(v^l) \neq v^l$, il existe π^l telle que, $F^{\pi^l}(v^l) = v^l$ et puisque $F^{\mathcal{R}} = \inf F^{\pi}$, nous avons $F^{\mathcal{R}}(v^l) \leq F^{\pi^l}(v^0) = v^l$ et de $F^{\mathcal{R}}(v^l) \neq v^l$, nous concluons que $F^{\mathcal{R}}(v^l) < v^l$.

2. Nous prouvons le deuxième point par récurrence $l \in \mathbb{N}$. Nous supposons que $l = 0$ et que $F^{\mathcal{R}}(v^0) \neq v^0$ alors $v^0 \in \mathcal{FS}$ sinon l'algorithme s'arrête. Il existe π^1 telle que $F^{\mathcal{R}}(v^0) = F^{\pi^1}(v^0) < v^0$ par la première assertion. Par ailleurs, v^1 est le plus petit élément de $\{v \in \overline{\mathbb{R}}^{\mathbb{P}} \mid F^{\pi^1}(v) \leq v\}$ d'où $v^1 \leq v^0$ et puisque $F^{\pi^1}(v^0) < v^0$ nous concluons que $v^1 < v^0$. Le même raisonnement s'applique à $v^l < v^{l-1}$ et $v^l \in \mathcal{FS}$. ∎

Retour aux exemples

Nous illustrons l'itération sur les politiques 7 sur les trois exemples exposés plus haut. Commençons tout d'abord par montrer en analysant la simple affectation que la rotation préserve le cercle unité invariant.

Exemple 6.9 (Fin de l'exemple 6.1)
Nous voulons donc prouver que la sphère unité est invariante par rotation c'est-à-dire $A(S^1) = S^1$ où A est une matrice orthogonale. Les inclusions $S^1 \subseteq A(S^1)$ et $A(S^1) \subseteq S^1$ sont de même nature puisque A est matrice orthogonale : $((x,y) \in S^1 \implies (x,y) \in A(S^1)) \iff (A^T(x,y) \in S^1 \text{ for } (x,y) \in S^1)$, donc, montrer que $S^1 \subseteq T(S^1)$ se ramène $A^T(S^1) \subseteq S^1$. Nous ne prouvons que $A(S^1) \subseteq S^1$ et nous utilisons la relaxation de Shor pour le prouver.

Nous reprenons à nouveau l'ensemble des fonctions quadratiques $P = \{p_1(x,y) \mapsto x \cdot x,\ p_2(x,y) \mapsto -(x \cdot x)\}$. Au deuxième point de contrôle, nous avions :

$$[F_2^\sharp(v)](p_1) = A(v_1^\star)^\dagger(p_1) = \sup\{p_1(A(x,y)) \mid p_1(x,y) \leq v_1(p_1),\ p_2(x,y) \leq v_1(p_2)\}$$
$$[F_2^\sharp(v)](p_2) = A(v_1^\star)^\dagger(p_2) = \sup\{p_2(A(x,y)) \mid p_1(x,y) \leq v_1(p_1),\ p_2(x,y) \leq v_1(p_2)\}$$

et

$$\begin{array}{rcl} F_1^{\mathcal{R}}(v) & = & (-1,1) \\ F_2^{\mathcal{R}}(v) & = & \inf\limits_{\lambda(p_1),\lambda(p_2),\mu \geq 0} \lambda(p_1)v_1(p_1) + \lambda(p_2)v_1(p_2) + \sup\limits_{x \in \mathbb{R}^d} \mathbf{e}_{Ax} + \lambda(p_1)(x \cdot x) - \lambda(p_2)(x \cdot x) \end{array}$$

Comme il n'y a pas de boucle, il faut chosir une politique initiale π^0 telle que $[\mathbb{V}_2(\pi^0)](p) < +\infty$, pour tout $p = p_1, p_2$. Ceci signifie que l'on doit choisir λ associée à p_1 telle que :

$$\sup_{x \in \mathbb{R}^d} -Ax \cdot Ax + \lambda(p_1)(x \cdot x) - \lambda(p_2)(x \cdot x) < +\infty \ .$$

et λ' associé à p_2 telle que :

$$\sup_{x \in \mathbb{R}^d} Ax \cdot Ax + \lambda'(p_1)(x \cdot x) - \lambda'(p_2)(x \cdot x) < +\infty \ .$$

Comme A est orthogonale ceci est équivalent à :

$$\sup_{x \in \mathbb{R}^d} -x \cdot x + \lambda(p_1)(x \cdot x) - \lambda(p_2)(x \cdot x) < +\infty \ .$$

et

$$\sup_{x \in \mathbb{R}^d} x \cdot x + \lambda'(p_1)(x \cdot x) - \lambda'(p_2)(x \cdot x) < +\infty \ .$$

Nous choisissons donc $\lambda(p_1) = 1$ et $\lambda(p_2) = 0$ et $\lambda'(p_1) = 0$ et $\lambda'(p_2) = 1$. On trouve grâce à cette politique initiale, $[\mathbb{V}_2(\pi^0)](p) = 0$ pour $p = p_1, p_2$. Nous devons maintenant résoudre le programme linéaire suivant :

$$\text{Min}\{v_1(p_1)+v_1(p_2)+v_2(p_1)+v_2(p_2) \mid -1 \leq v_1(p_1),\ 1 \leq v_1(p_2),\ v_1(p_1) \leq v_2(p_1),\ v_1(p_2) \leq v_2(p_2)\}$$

Ainsi on trouve $v_1(p_1) = v_2(p_1) = -1$ et $v_1(p_2) = v_2(p_2) = 1$.

Nous introduisons les matrices suivantes :

$$M(p_1 \circ A) = \begin{pmatrix} 0 & 0 \\ 0 & -\mathrm{Id} \end{pmatrix}, \; M(p_2 \circ A) = -M(p_1 \circ A),$$

$$M(p_1) = \begin{pmatrix} 0 & 0 \\ 0 & -\mathrm{Id} \end{pmatrix} \text{ et } M(p_2) = -M(p_1) \; .$$

Par la relaxation de Shor, le problème (ShorD) nous donne les égalités suivantes :

$$[F_2^{\mathcal{R}}(v)](p_1) = \underset{\substack{\lambda(p_1)\geq 0 \\ \lambda(p_2)\geq 0 \\ \eta\in\mathbb{R}}}{\mathrm{Min}} \eta \text{ s.t. } M(p_1 \circ A) + \eta N(-1) + \lambda(p_1)(N(-1) - M(p_1)) + \lambda(p_2)(N(1) - M(p_2)) \preceq 0$$

et

$$[F_2^{\mathcal{R}}(v)](p_2) = \underset{\substack{\lambda(p_1)\geq 0 \\ \lambda(p_2)\geq 0 \\ \eta\in\mathbb{R}}}{\mathrm{Min}} \eta \text{ s.t. } M(p_2 \circ A) + \eta N(-1) + \lambda(p_1)(N(-1) - M(p_1)) + \lambda(p_2)(N(1) - M(p_2)) \preceq 0$$

On peut réécrire les deux programmes SDP comme suit :

$$[F_2^{\mathcal{R}}(v)](p_1) = \underset{\substack{\lambda(p_1)\geq 0 \\ \lambda(p_2)\geq 0 \\ \eta\in\mathbb{R}}}{\mathrm{Min}} \eta \text{ s.t. } \begin{pmatrix} -\eta - \lambda(p_1) + \lambda(p_2) & 0 \\ 0 & -\mathrm{Id} + \lambda(p_1)\,\mathrm{Id} - \lambda(p_2)\,\mathrm{Id} \end{pmatrix} \preceq 0$$

and

$$[F_2^{\mathcal{R}}(v)](p_2) = \underset{\substack{\lambda(p_1)\geq 0 \\ \lambda(p_2)\geq 0 \\ \eta\in\mathbb{R}}}{\mathrm{Min}} \eta \text{ s.t. } \begin{pmatrix} -\eta - \lambda(p_1) + \lambda(p_2) & 0 \\ 0 & \mathrm{Id} + \lambda(p_1) - \lambda(p_2) \end{pmatrix} \preceq 0$$

Pour résoudre ces problèmes, on pourrait appeler un solveur SDP, mais dans ce cas, il suffit de résoudre un système d'inégalités : tous les éléments diagonaux elements doivent être négatifs, par exemple, pour le premier problème, cela implique $\eta \geq -1$ et puisque que l'on minimise η on obtient $\eta = -1$. Finalement, on trouve $v_2(p_1) = -1$ and $v_2(p_2) = 1$ et nous avons prouvé que $A(v_1^\star)^\dagger = v_1$ et on conclut que la sphère unité $S^1 = \{x \in \mathbb{R}^d \mid \|x\| = 1\}$ est invariante par rotation.

Exemple 6.10 (Fin de l'exemple 6.2)
Nous rappelons que les fonctions de base sont $p_x : (x, v) \mapsto x^2$, $p_v : (x, v) \mapsto v^2$ et $p_L : (x, v) \mapsto 3v^2 + 2x^2 + 2xv$, et donc la configuration est $\mathbb{P} = \{p_x, p_v, p_L\}$. On s'intéresse à la troisième coordonnée de la fonction sémantique relâchée $[F^{\mathcal{R}}(v)](p)$, considérons, par exemple, $p = p_x$, on obtient : $[F_3^{\mathcal{R}}(v)](p_x) =$

$$\inf_{\lambda\in\mathbb{R}_+^{\mathbb{P}}} \sup_{(x,v)\in\mathbb{R}^2} \sum_{q\in P} \lambda(q) w_2(q) + (x, v) \left(\begin{pmatrix} 1 - \lambda(p_x) & h/2 \\ h/2 & h^2 - \lambda(p_v) \end{pmatrix} - \lambda(p_L) L \right) (x, v)^T \tag{6.7}$$

En introduisant les matrices suivantes, nous pouvons réécrire (6.7) comme (6.8) :

$$M(p_x) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; M(p_v) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } M(p_x \circ T) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & h/2 \\ 0 & h/2 & h^2 \end{pmatrix}$$

$$[F_3^{\mathcal{R}}(w)](p_x) = \underset{\substack{\lambda\in\mathbb{R}_+^{\mathbb{P}}\\ \eta\in\mathbb{R}}}{\text{Min}}\ \eta \ \text{ s.t. } \ M(p_x\circ T)+\eta N(-1)+\sum_{q=p_x,p_v,p_L}\lambda(q)[N(w_2(q))-M(q)] \preceq 0 \quad (6.8)$$

Pour initialiser l'itération sur les politiques 7, nous devons choisir la politique π^0. Pour la troisième coordonnée de $F^{\mathcal{R}}$, nous choisissons la politique :

$$\pi_3^0(p_x) = (0,0,1),\ \pi_3^0(p_v) = (0,0,1),\ \pi_3^0(p_L) = (0,0,1)\ .$$

Cela consiste, pour $p = p_x$, à prendre $\lambda(p_x) = \lambda(p_v) = 0$ et $\lambda(p_L) = 1$ dans (6.7). Par la proposition 6.1 nous trouvons :

$$V_3^{\pi_3^0}(p_x) = \sup_{(x,v)\in\mathbb{R}^2}(x,v)\begin{pmatrix}-1 & h/2-1\\ h/2-1 & h^2-3\end{pmatrix}(x,v)^T = 0$$

$$V_3^{\pi_3^0}(p_v) = \sup_{(x,v)\in\mathbb{R}^2}(x,v)\begin{pmatrix}h^2-2 & h(1-h)-1\\ h(1-h)-1 & (1-h)^2-3\end{pmatrix}(x,v)^T = 0$$

$$V_3^{\pi_3^0}(p_L) = \sup_{(x,v)\in\mathbb{R}^2}(x,v)(T^TLT-L)(x,v)^T = 0$$

La solution du problème de maximisation est nulle puisque les trois matrices sont définie négatives. La troisième matrice $A^TLA - L$ est définie négative car L satisfait la condition de Lyapunov pour le système linéaire discret $(x,v) = A(x,v)$. Pour calculer le plus petit point fixe de $F(\pi^0,\cdot)$, nous résolvons le programme linéaire suivant :

$$\min\sum_{i=1}^{3}\sum_{p\in P}\beta_i(p)$$
$$\beta_2(p_L)\leq\beta_3(p_x),\ \beta_2(p_L)\leq\beta_3(p_v),\ \beta_2(p_L)\leq\beta_3(p_L)$$
$$\beta_3(p_x)\leq\beta_2(p_x),\ \beta_3(p_v)\leq\beta_2(p_v),\ \beta_3(p_L)\leq\beta_2(p_L)$$
$$1\leq\beta_2(p_x),\ 1\leq\beta_2(p_v),\ 7\leq\beta_2(p_L)$$
$$1\leq\beta_1(p_x),\ 1\leq\beta_1(p_v),\ 7\leq\beta_1(p_L)$$

Nous utiisons le solveur linéaire `Linprog`, et nous trouvons :

$$\begin{array}{lll} u_1^0(p_x) = 1.0000 & u_2^0(p_x) = 7.0000 & u_3^0(p_x) = 7.0000\\ u_1^0(p_v) = 1.0000 & u_2^0(p_v) = 7.0000 & u_3^0(p_v) = 7.0000\\ u_1^0(p_L) = 7.0000 & u_2^0(p_L) = 7.0000 & u_3^0(p_L) = 7.0000\end{array}$$

L'approximation de la clôture de u^0, $\mathcal{R}$-$\text{vex}_{\mathbb{P}}(u^0)$ est donnée par une implémentation `Matlab`, utilisant `Yalmip` et `SeDuMi` retourne le vecteur w_1^0 :

$$\begin{array}{lll} w_1^0(p_x) = 1.0000 & w_2^0(p_x) = 4.2000 & w_3^0(p_x) = 4.2000\\ w_1^0(p_v) = 1.0000 & w_2^0(p_v) = 2.8000 & w_3^0(p_v) = 2.8000\\ w_1^0(p_L) = 7.0000 & w_2^0(p_L) = 7.0000 & w_3^0(p_L) = 7.0000\end{array}$$

En utilisant à nouveau `Yalmip` et le solveur `SeDuMi`, nous trouvons que w n'est pas un point fixe de $F^{\mathcal{R}}$. Comme $w_2^0(p_x) > 0$, $w_2^0(p_v) > 0$ et $w_2^0(p_L) > 3$, $F_3^{\mathcal{R}}$ coïncide avec $F_3{}^\sharp$. Nous trouvons une nouvelle politique et nous trouvons :

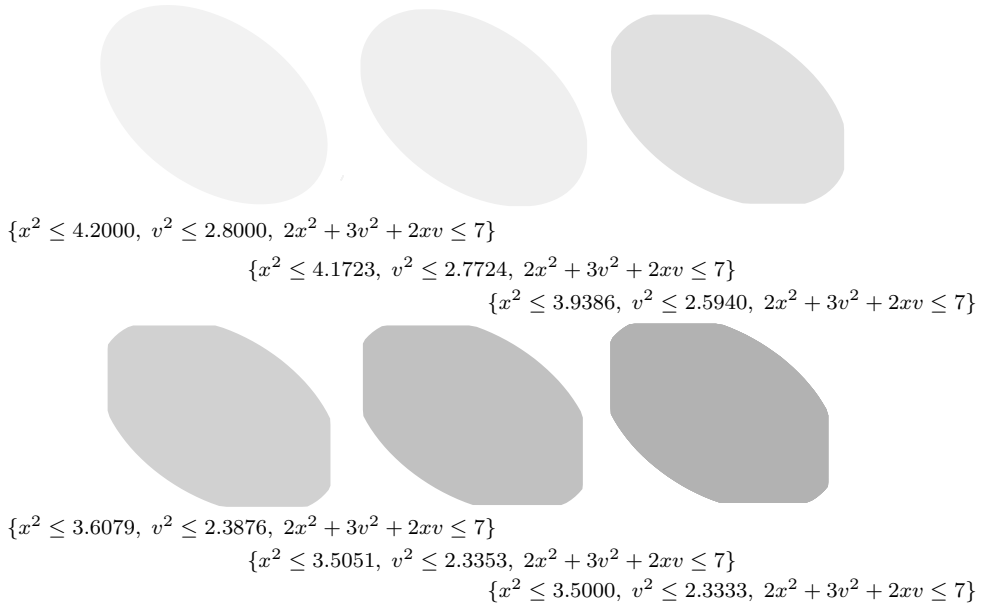


Figure 6.6 – Itérations successives de l'itération sur les politiques au point de contrôle 2.

$$\pi_3^1(p_x) = (0, 0, 0.596),\ \pi_3^1(p_v) = (0, 0, 0.3961),\ \pi_3^1(p_L) = (0, 0, 0.9946)\ .$$

Finalement, après 5 itérations nous trouvons l'invariant de boucle $w_2^\star$ au point de contrôle 2 l'ensemble :

$$\{x^2 \leq 3.5000,\ v^2 \leq 2.3333,\ 2x^2 + 3v^2 + 2xv \leq 7\}\ .$$

Les dessins des ensembles successifs $w_2^\star$ à chaque itérations de l'itération sur les politiques sont dérits à la figure 6.6.

Remarquons que puisque w^∞ final est un point fixe de $F^{\mathcal{R}}$ et puisque $w_2^\infty(p_x) > 0$, $w_2^\infty(p_v) > 0$ et $w_2^\infty(p_L) > 3$ alors $F_3^{\mathcal{R}} = {F_3}^\sharp$ et donc w_3^∞ est une fonction $\mathbb{P}$-convexe et comme w_1^∞ est une fonction $\mathbb{P}$-convexe alors w_2^∞ est une fonction $\mathbb{P}$-convexe en tant que supremum de fonctions $\mathbb{P}$-convexes et finalement w^∞ est un point fixe de $F^\sharp$.

Exemple 6.11 (Fin de l'exemple 6.3)
Nous rappelons que prenons la configuration munie d'une unique fonction : $\mathbb{P} = \{(i, j) \mapsto -\frac{i(i+1)}{2} + j\}$. La fonction sémantique abstraite $F^\sharp$ du programme 6.3 est la fonction suivante :

$$\begin{array}{rcl} {F_1}^\sharp(v) & = & 0 \\ {F_2}^\sharp(v) & = & (v_1^{\mathbf{s}} \cup v_3^{\mathbf{s}})^\sigma \\ {F_3}^\sharp(v) & = & \sup\limits_{\substack{(i,j) \in v_2^{\mathbf{s}} \\ i \leq 42}} \mathbf{e}_{T(i,j)^T + b} \end{array}$$

La fonction sémantique relâchée est la fonction :

$$\begin{array}{lcl} F_1^{\mathcal{R}}(v) & = & 0 \\ F_2^{\mathcal{R}}(v) & = & \sup(v_1, v_3) \\ F_3^{\mathcal{R}}(v) & = & \inf\limits_{\lambda,\mu\geq 0} \lambda v_2 \sup\limits_{(i,j)\in\mathbb{R}^2} \mathbf{e}_{T(i,j)^T+b} + 42\mu - \mu i + \lambda p(i,j) \end{array}$$

où $T = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ et $b^T = (1,1)$. On cherche une politique initiale telle que :

$$\sup_{(i,j)\in\mathbb{R}^2} \frac{1}{2}(-(i+1)(i+2) + \lambda i(i+1)) + 1 + i + j - \lambda j + \mu(42-i) < +\infty \ .$$

On voit que la fonction objectif est linéaire en j, pour que le sup soit fini, il faut que $\lambda = 1$ et dans ce cas où cherche maintenant $\mu \geq 0$ tel que :

$$\sup_{(i,j)\in\mathbb{R}^2} \frac{1}{2}(-i^2 - 3i - 2 + i^2 + i) + 1 + i + 42\mu - \mu i < +\infty \ .$$

ce qui équivaut à :

$$\sup_{(i,j)\in\mathbb{R}^2} -i - 1 + 1 + i + 42\mu - \mu i < +\infty \ .$$

On conclut que $\mu = 0$. En prenant comme politique initiale $\lambda = 1$ et $\mu = 0$, on obtient $\mathbb{V}_3(\pi^0) = 0$ et donc nous devons résoudre :

$$\begin{array}{c} v_1 \geq 0 \\ v_2 \geq v_1 \\ v_2 \geq v_3 \\ v_3 \geq v_2 \end{array}$$

et on trouve finalement $v_1 = v_2 = v_3 = 0$. Nous pourrions calculer la fonction sémantique relâchée à l'aide de Shor mais l'algorithme s'arrête avec ce vecteur là puisque $\lambda = 1$ et $\mu = 0$ fournit l'unique politique telle que $\mathbb{V}_3(\pi^0) < +\infty$.

6.2.3 Itération max-stratégie dans le domaine des configurations quadratiques

Gawlitza et Seidl [GS10] ont développé leur technique de max-stratégie ou d'itération sur les politiques ascendantes. Les fonctions que considèrent les auteurs sont des maxima d'un nombre fini de fonctions concaves. Pour les affectations et les tests, les auteurs définissent une fonction sémantique relâchée qui est le dual de la relaxation de Shor. Les auteurs initialisent leur algorithme avec la fonction identiquement égale à $-\infty$. A chaque itération, pour chaque coordonnée, ils sélectionnent la fonction qui atteint le maximum et calculent le plus grand point fixe de la fonction associée à cette sélection. A la manière de l'itération de Kleene, la suite générée par l'algorithme est une suite croissante dont tous les termes minorent le plus petit point fixe de $F^{\mathcal{R}}$, la limite de la suite est le plus petit point fixe de $F^{\mathcal{R}}$. La limite est atteinte en temps fini puisque le nombre de politiques est fini. Cependant aucun terme de la suite générée par l'algorithme ne fournit un invariant valide.

L'article soumis [GSA+] récapitule les deux techniques d'itération sur les politiques dans les configurations quadratiques.

CHAPITRE 7

CONCLUSION SUR LA PARTIE DOMAINES DES SOUS-NIVEAUX

Nous terminons la partie domaine numériques des sous-niveaux par un récapitulatif des résultats obtenus et par des extensions possibles du travail effectué. Cette partie a été l'objet d'une publication [AGG10] ainsi que d'une soumission [AGG]. Nous découpons ce chapitre de conclusion en deux sections, la section 7.1 qui rappelle les résultats présentés dans cette partie puis les perspectives dans la section 7.2.

7.1 Résumé

7.1.1 Treillis des ensembles convexes abstraits et des fonctions convexes abstraites

Nous avons commencé ce chapitre par la construction d'un domaine numérique abstrait général dont les éléments sont des intersections de sous-nveaux de fonctions prélablement fixées. Nous avons montré que ces ensembles étaient uniquement caractérisés par les fonctions qui définissent le niveaux. Nous avons montré que la paire d'applications entre l'ensemble des applications sur les fonctions de base et les parties de $\mathbb{R}^d$ forme une correspondance de Galois. Nous avons donc par la suite considéré les treillis des parties de $\mathbb{R}^d$ closes et ainsi que les applications sur les fonctions de base closes. Les éléments clos décrits sont en réalité des convexes abstraits. Nous avons proposé une généralisation des travaux de Sankaranarayanan et al [SSM05] qui généralisaient déjà les travaux de Miné [Min04].

7.1.2 Construction de la fonction sémantique relâchée par dualité

Dans ce chapitre, nous avons appliqué des techniques classiques d'optimisation sous contraintes, pour construire une fonction sémantique relâchée. En effet, chaque fonction $v \in (\overline{\mathbb{R}}^{\mathbb{P}})^n$ sur les fonctions de base $p \in \mathbb{P}$ définissent une famille de problème de maximisation. Puisque le nombre de fonction de base n'est, a priori, pas fixé et que la configuration $\mathbb{P}$ ne dispose a priori, pas de bonnes structures, la dualité peut varier au cours de l'analyse suivant le domaine des fonctions v sur la configuration $\mathbb{P}$. De plus, le problème dual introduit possède lui de bonnes structures et nous utilisons cette bonne structure pour déterminer une borne valide des valeurs prises par la fonction sémantique abstraite. Nous avons également développé

une méthode d'itération sur les politiques dynamique. La géométrie de l'invariant cherché se dessine pas à pas en choisissant de plus en plus de fonctions de base.

7.1.3 Configurations quadratiques en interprétation abstraite

Dans un deuxième temps, nous sommes intéressés aux configurations quadratiques finies. Dans ce cadre, nous avons utilisé la relaxation de Shor pour calculer la fonction sémantique relâchée. Le point important est que la fonction sémantique dans ce cas se calcule en temps polynomial en résolvant un problème d'optimisation semi-definie positive. Dans les exemples, nous avons proposé d'utiliser l'information d'une fonction de Lyapunov pour déterminer un invariant numérique sous la forme d'intersections d'ellipsoïdes.

7.2 Perspectives

7.2.1 Problème de la politique initiale

Le problème de trouver une heuristique pour trouver facilement des politiques initiales est un problème important. Nous avons vu que trouver une politique initiale se ramenait à trouver une politique dont le problème d'optimisation linéaire qui permet de trouver le plus petit point est réalisable. Dans le cas où cette politique n'existe pas, il faudrait trouver une condition simple pour le caractériser. Lorsque les programmes contiennent simplement des boucles, on peut initialiser l'algorithme d'itérations sur les politiques en considérant uniquement l'information donnée par une fonction de Lyapunov. En effet, lorsqu'on considère les configurations quadratiques de la forme :

$$\{x \mapsto \pm x_i,\ L_a : x \mapsto x^T L_a x,\ L_\imath : x \mapsto x^T L_\imath x\}\ .$$

où $x \mapsto x^T L_a x$ et $x \mapsto x^T L_\imath x$ sont des fonctions de Lyapunov pour les affectations aux points de contrôle $a \in \mathbb{A}$ et $\imath \in \mathbb{I}$ la politique définie par :

$$[\pi^0_a(p)](q) = \begin{cases} 0 & \text{si } q \neq L_a \\ 1 & \text{si } q = L_a \end{cases} \quad \text{et } [\pi^0_\imath(p)](q) = \begin{cases} 0 & \text{si } q \neq L_\imath \\ 1 & \text{si } q = L_\imath \end{cases}$$

semble fournir une politique initiale valide. Il faudrait pouvoir prouver ce résultat et définir un heuristique pour déterminer des politiques initiales en fonction des types des configurations utilisées.

7.2.2 Configuration et arithmétique polynomiale

Nous n'avons pas traité les programmes à arithmétiques polynomiales dont les affectations et tests sont des polynômes multivariés de degré supérieurs à 2. Une technique classique d'optimisation polynomiale est d'utiliser la programmation en sommes de carrés [1]. Cependant, la décomposition en sommes des carrés est coûteuse. Nous tenterons donc de développer une approche par la programmation géométrique où l'on considère des fonctions qui sont de la somme d'éléments de la forme :

$$c_k x_1^{a_i} x_2^{a_2} \dots x_d^{a_d}$$

1. traduction directe du terme "sum of squares" (SOS)

où $c_k > 0$ et a_i sont des nombres réels. L'ensemble des fonctions de cette forme contient l'ensemble des polynômes à coefficients positifs. Une arithmétique formée de ce type de fonctions permet de considérer les divisions de variables.

7.2.3 Fonctions de Lyapunov non quadratique et systèmes hybrides

Dans les exemples du chapitre 6, nous avons utilisé des fonctions de Lyapunov quadratiques. En effet, nous nous sommes intéressés à la condition de Lyapunov pour les systèmes discrets et linéaires. Cependant, des travaux de Blanchini [Bla95] sur l'utilisation de fonctions de Lyapunov non quadratiques pour les problèmes de contrôle robuste, pourraient généraliser notre approche. Par ailleurs, Johansson et Rantzer [JR98] ont étudié la construction et le calcul de fonctions de Lyapunov quadratiques par morceaux pour les systèmes hybrides. Cette approche pourrait nous permettre d'analyser des systèmes hybrides.

7.2.4 Problème de calculs garantis et flottants

Il existe des solveurs de programmes semi-definis qui peuvent calculer des solutions garanties comme par exemple VSDP [JCK07]. De tels solveurs existent également pour l'optimisation linéaire comme [Kei05]. Cependant, notre algorithme et tous les calculs sont a priori en arithémétique exacte. Pour la programmation lineaire, on peut utiliser le fait qu'un programme lineaire admet une solution dans le corps algébrique des données.

7.2.5 Convergence de l'itération sur les politiques

Dans l'exemple 6.3, l'itération sur les politiques s'arrête puisque les multiplicateurs de Lagrange sont uniques. Une perspective de travail peut se diriger vers des conditions d'arrêt caractérisées par l'unicité des politiques. De plus, nous pouvons utiliser les résultats de "dualité forte" en optimisation pour assurer la convergence de l'algorithme d'itérations sur les politiques.

Troisième partie

A la recherche du plus petit point fixe

CHAPITRE 8

POINTS FIXES MINIMAUX D'APPLICATIONS SEMIDIFFÉRENTIABLES

L'algorithme d'itération sur les politiques s'arrête sur un point fixe qui n'est pas nécessairement minimal. Dans ce chapitre, nous proposons de caractériser analytiquement le plus petit point fixe lorqu'il est fini. Dans [CGG+05, Theorem 3], les auteurs ont montré que, dans le cadre contractant au sens large, l'itération sur les politiques s'arrête sur le plus petit point fixe si celui-ci n'est atteint que par une unique politique. Une question se pose immédiatement : que se passe-t-il quand le point fixe est commun à plusieurs politiques ? Pour y répondre, nous interprétons ce problème comme un problème de minimisation. Un minimiseur d'une fonction convexe se caractérise par un point où la dérivée s'annule dans le cas différentiable, un point pour lequel le sous-différentiel contient la forme linéaire nulle dans le cas convexe. Les fonctions apparaissant en analyse statique ne sont ni dérivables ni sous-différentiables en général, par conséquent, nous utilisons une notion de différentiablité plus faible, la semidifférentiabilité. Le problème comporte une deuxième contrainte : nous cherchons un point fixe. Notre méthode consiste à déplacer notre problème de point fixe global à des problèmes locaux sur l'ensemble des points fixes de la semidifférentielle au point. Dans le cas linéaire, un problème de point fixe devient un problème de valeurs et vecteurs propres. Grâce aux propriétés de la semidifférentielle, nous pourrons obtenir un analogue non linéaire du rayon spectral. Une telle approche s'inspire de l'approche d'Akian, Gaubert et Nussbaum pour des problèmes d'unicité de point fixe en dimension infinie [AGN]. Nous commençons à la section 8.1 par définir les outils d'analyse non-linéaire qui nous permettront de caractériser le plus petit point fixe. Dans la section 8.2, nous allons énoncer les résultats de caractérisation du plus petit de point fixe. Enfin, à la section 8.3, nous terminons ce chapitre en exposant les problèmes calculatoires de la caractérisation du plus petit point fixe. Une partie des résultats a fait l'objet d'une publication [AGG08a].

8.1 Préliminaires

Dans cette section, nous débutons par un exemple simple de programme pour lequel l'initialisation de l' itérations sur les politiques par l'heuristique classique sur le domaine des intervalles ne conduit pas au plus petit point fixe. Puis nous énonçons un résultat d'existence

de plus petit point fixe fini à la sous-section 8.1.1. Dans la sous-section 8.1.2, nous définissons le rayon spectral non-linéaire et dans la sous-section 8.1.3, nous définissons les applications semidifférentielles.

Dans l'étude de la minimalité, on s'intéressera ici aux points fixes à coordonnées finies, on cherchera donc des points fixes minimaux par rapport à l'ordre produit usuel de $\mathbb{R}^d$.

Même dans les cas simples, le calcul de point fixe par itération sur les politiques ne permet de garantir la minimalité du point fixe d'une fonction sémantique abstraite d'analyse statique par interprétation abstraite. Pour illustrer ces propos, nous introduisons un programme simple à l'exemple 8.1.

Exemple 8.1
Nous utilions pour cet exemple, l'algorithme d'itération sur les politiques dans le cadre des intervalles. La politique initiale est choisie grâce à l'heuristique classique.

```
x:=0;  [1]
while  (x<=99){  [2]
  x=1-x;  [3]
  }  [4]
```

$$F_1^\sharp(X) = [0,0]$$
$$F_2^\sharp(X) = [-\sup(0, x_2^+), \inf(99, \sup(0, x_2^- + 1))]$$
$$F_3^\sharp(X) = [-(x_2^+ - 1), 1 + x_2^-]$$
$$F_4^\sharp(X) = [-\inf(-100, \sup(0, x_2^+ - 1)), \sup(0, x_2^- + 1)]$$

Figure 8.1 – Un simple programme à gauche et à droite, la fonction sémantique abstraite associée

$$\begin{array}{rcl} F_1^\sharp(X) & = & \{0\} \\ F_2^\sharp(X) & = & [-98, 99] \\ F_3^\sharp(X) & = & [-98, 99] \\ F_4^\sharp(X) & = & \emptyset \end{array}$$

Figure 8.2 – Les ensembles invariants à chaque point de contrôle du programme de la figure 8.1

Le programme de la figure 8.1 simule une boucle infinie et la valeur de la variable x est soit 0 soit 1. Cependant, l'itération sur les politiques trouve au point de contrôle 2, l'intervalle $[-98, 99]$.

Soit un treillis complet $\mathcal{L}$ et soit une fonction f satisfaisant :

$$f = \inf\{f^\pi \mid \pi \in \Pi\} \text{ et } \forall x \in \mathcal{L},\ \exists \pi \in \Pi,\ f(x) = f^\pi(x)\ .$$

Deux théorèmes sur la minimalité d'un point fixe avaient été énoncés dans l'article [CGG+05]. Le premier résultat [CGG+05, Theorem 1] permet de caractériser $\mathrm{lfp}(f)$ comme le minimum des $\mathrm{lfp}(f^\pi)$:

$$\mathrm{lfp}(f) = \min\{\mathrm{lfp}(f^\pi) \mid \pi \in \Pi\}\ . \tag{8.1}$$

Ce résultat indique que l'on pourrait calculer le plus petit point fixe de f en calculant les plus petits points fixes des fonctions f^π. Bien évidemment, cette méthode de calcul n'est pas robuste et peut exploser lorsque le nombre de politiques augmente. Le deuxième résultat [CGG+05, Theorem 3] affirme que, dans le cas des applications contractantes au sens large

(nous reviendrons sur cette définition plus tard), si l'itération sur les politiques termine sur un point fixe fini x et s'il existe une unique fonction f^π telle que $f(x) = f^\pi(x)$ alors x est le plus petit point fixe de f.

8.1.1 Existence de solutions finies

Dans tout ce chapitre, nous munissons, sauf mention du contraire, $\mathbb{R}^d$ de la topologie usuelle ainsi que de l'ordre usuel sur $\mathbb{R}^d$. Dans cette sous-section, nous nous intéressons aux fonctions croissantes $f : \overline{\mathbb{R}}^d \mapsto \overline{\mathbb{R}}^d$ qui ont un plus petit point fixe fini, c'est-à-dire aux fonctions croissantes f telles que $\inf\{v \in \overline{\mathbb{R}}^d \mid f(v) \leq v\} \in \mathbb{R}^d$. On va établir, à titre d'exemple, un critère d'existence d'un plus petit point fixe fini.

Proposition 8.1 (*Existence de plus petit point fixe*)

Supposons $f : \overline{\mathbb{R}}^d \mapsto \overline{\mathbb{R}}^d$ croissante. Notons, $\perp$ le plus petit élément de $\overline{\mathbb{R}}^d$. Supposons de plus :

1. $\exists\, \bar{k} \in \mathbb{N},\ f^{\bar{k}}(\perp) \in \mathbb{R}^d$;
2. $\exists\, w \in \mathbb{R}^d$ tel que $(f^k(w))_{k\geq 0}$ est bornée dans $\mathbb{R}^d$;
3. f est continue sur $\mathbb{R}^d$.

Alors $\mathrm{lfp}(f) \in \mathbb{R}^d$.

Démonstration
Soit $w \in \mathbb{R}^d$ tel que $(f^k(w))_{k\geq 0}$ soit bornée et posons $z = \liminf_{k\to+\infty} f^k(w) = \lim_{k\to+\infty} \inf_{m\geq k} f^m(w)$. Le vecteur $z \in \mathbb{R}^d$ et par continuité de f, on a $f(z) = \lim_{k\to+\infty} f(\inf_{m\geq k} f^m(w))$ puis par croissance de f, on a $f(z) \leq \lim_{k\to+\infty} \inf_{m\geq k+1} f^m(w) = z$. Comme $\perp \leq z$ et f est croissante $f^{\bar{k}}(\perp) \leq z$. On déduit du fait que $\perp \leq f(\perp)$ et $f(z) \leq z$, l'ensemble :

$$L = \{y \in \mathbb{R}^d \mid f^{\bar{k}}(\perp) \leq y \leq z\}$$

est un treillis complet invariant par f et donc f possède un plus petit point fixe u dans L. On conclut que u est le plus petit point fixe de f tel que $f^{\bar{k}}(\perp) \leq u$ et comme tout point fixe v de f dans $\overline{\mathbb{R}}^d$ satisfait $f^{\bar{k}}(\perp) \leq v$ alors u est le plus petit point fixe de f dans $\overline{\mathbb{R}}^d$ et donc $\mathrm{lfp}(f) \in \mathbb{R}^d$. ∎

Le théorème de Tarski 2.1 donne une idée du calcul de point fixe : on sait qu'il faut considérer l'ensemble des $\{x \in \overline{\mathbb{R}}^d \mid f(x) \leq x\}$ dès que f est croissante. Il suffit donc de trouver une fonction g telle que $g(x_0) \leq g(y)$ pour tout $y \in \{x \in \mathbb{R}^d \mid f(x) \leq x\}$ implique que x_0 est le plus petit point fixe de f. On peut donc considérer une fonction g strictement croissante sur $\mathbb{R}^d$. Par exemple, la fonction qui à un vecteur associe la somme de ses coordonnées est une fonction qui vérifie cette propriété.

Théorème 8.1 (*Plus petit point fixe fini et optimisation*)

Soit $f : \overline{\mathbb{R}}^d \mapsto \overline{\mathbb{R}}^d$ une fonction croissante telle que $\mathrm{lfp}(f) \in \mathbb{R}^d$. Soit g une fonction strictement croissante sur $\mathbb{R}^d$. Alors $\mathrm{lfp}(f)$ est l'unique solution du problème d'optimisation :

$$\text{Min} \quad g(x) \quad \text{s.c} \quad f(x) \leq x, x \in \mathbb{R}^d$$

Démonstration
On pose $\bar{x} = \text{lfp}(f)$. Le vecteur $\bar{x}$ est clairement une solution réalisable. Soit $x \in \mathbb{R}^d$ réalisable. Par le théorème de Tarski, $\bar{x} \leq x$ puis par croissance de g, $g(\bar{x}) \leq g(x)$. La stricte croissance de g implique l'unicité de la solution. ∎

Le théorème 8.1, donne un moyen théorique de calculer un plus petit point fixe lorsque celui-ci est fini. Ce moyen est efficace quand chaque coordonnée de f est une fonction convexe. En général les fonctions à étudier en analyse statique ne sont pas convexes, mais ce théorème demeure utile pour les fonctions associées à une politique qui sont en général affines.

8.1.2 Rayon spectral non linéaire

Après avoir montré, avec des hypothèses simples, l'existence d'un plus petit point fixe fini, on introduit les premières définitions à la base du travail effectué. Nous rappelons ce qu'est un cône.

Définition 8.1 (*Cônes*)
On dit qu'un ensemble $C \subseteq \mathbb{R}^d$ est un cône si pour tout λ réel positif ou nul et pour tout $x \in C$, $\lambda x \in C$.

Exemple 8.2
Les ensembles $\mathbb{R}^d_+$ et $\mathbb{R}^d_-$ sont des cônes.

Exemple 8.3
Soit l'ensemble $C = \{(x, y) \in \mathbb{R}^2_+ \mid x \leq 2y,\ y \leq 2x\}$. L'ensemble C est un cône.

Exemple 8.4
Considérons l'ensemble $A = \bigcup_{a=1,\ldots,N} \{(x, y) \in \mathbb{R}^d_+ \times \mathbb{R}_+ \mid y = a \sum_i x_i\}$ où N désigne un entier quelconque. L'ensemble A est un cône.

Dans la théorie des applications linéaires, il est utile de calculer le rayon spectral pour déterminer comment une application linéaire dilate la boule unité. Le rayon spectral peut être utilisé pour étudier localement des applications différentiables. Dans notre cas, nous voulons étudier localement des applications qui n'ont pas de "différentielles linéaires", nous introduisons un rayon spectral pour des applications non linéaires.

Définition 8.2 (*Rayon spectral non linéaire*)
Soit C un cône de $\mathbb{R}^d$, pour une fonction g de C dans C, continue, on définit le rayon spectral $\rho_C(g)$ de g, relativement à C de la manière suivante :

$$\rho_C(g) = \sup\{\lambda \geq 0 \mid \exists\ x \in C \backslash \{0\}, g(x) = \lambda x\}$$

Exemple 8.5 (Algèbre linéaire)
Prenons une matrice :

$$A = \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}$$

L'application linéaire associée à la matrice A préserve le cône positif. En calculant les valeurs propres de la matrice, nous trouvons $1+\sqrt{2}$ et $1-\sqrt{2}$. L'espace propre associé à la valeur propre $1+\sqrt{2}$ est la droite engendrée par le vecteur $(1, \sqrt{2}/2)$.

On a par conséquent $\rho_{\mathbb{R}^2_+}(A) = 1+\sqrt{2}$.

Exemple 8.6 (Application non-linéaire)
On définit la fonction f sur $\mathbb{R}^2_+$ par :

$$f(x_1, x_2) = \begin{pmatrix} \sup(x_1, x_2) \\ \inf(x_1, x_2) \end{pmatrix}$$

La fonction f est stable sur $\mathbb{R}^2_+$ et $\lambda \geq 0$ est valeur propre de f si, pour un certain couple $(x_1, x_2) \neq (0,0)$ si :

$$\begin{aligned} \sup(x_1, x_2) &= \lambda x_1 \\ \inf(x_1, x_2) &= \lambda x_2 \end{aligned}$$

Si $x_1 = 0$ alors $\sup(x_1, x_2) = x_2 = \lambda x_1$ et donc $x_2 = 0$, on conclut qu'un couple solution (x_1, x_2) vérifie $x_1 > 0$. Comme $\lambda x_1 \geq x_1$ et $\lambda x_2 \leq x_2$, on obtient $\lambda = 1$ d'où $\rho_{\mathbb{R}^2_+}(f) = 1$.

Exemple 8.7 (Application non-linéaire 2)
Soit $f : \mathbb{R} \mapsto \mathbb{R}$ telle que :

$$f(x) = \begin{cases} \sqrt{x} & \text{si } x \in [0,4] \\ (1/2)x & \text{si } x \in [4,6] \\ \min((1/12)x^2, 5) & \text{sinon} \end{cases}$$

Le graphe de la fonction est donné par la figure 8.3.

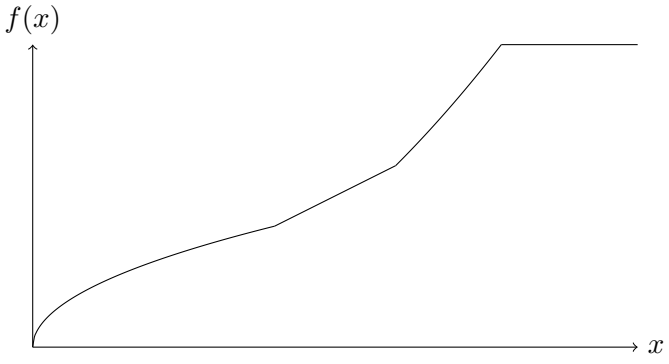


Figure 8.3 – Représentation graphique de la fonction f

Pour trouver les valeurs propres de f, il faut résoudre l'équation $f(x) = \lambda x$ sur les intervalles $[0,4]$, $[4,6]$, $[6, 2\sqrt{15}]$ et $[2\sqrt{15}, +\infty[$. Sur l'intervalle $[4,6]$, il n'existe pas de valeurs propres, la seule valeur propre possible serait $1/2$ mais cette valeur propre aurait pour vecteur propre $x = 0$ qui n'appartient pas à $[4,6]$. Sur les autres intervalles, la valeur propre associée à un vecteur propre est unique mais une valeur propre peut avoir bien évidemment plusieurs vecteurs propres. Le graphe des valeurs propres en fonction des vecteurs propres est donné par la figure 8.4.

On conclut que $\rho_{\mathbb{R}_+}(f) = +\infty$.

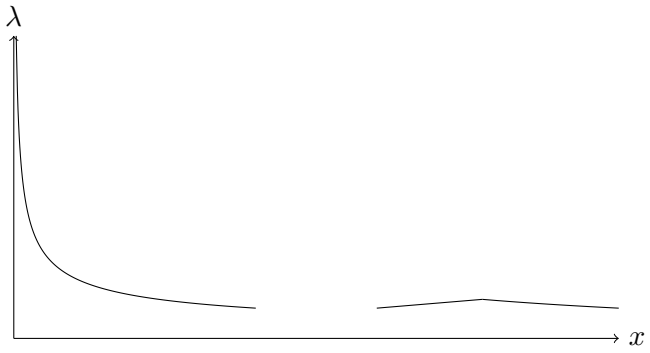


Figure 8.4 – Représentation graphique des valeurs propres en fonction de leur vecteur propre

Exemple 8.8 (Cône généré par deux demi-droites et fonction non-linéaire)
Considérons le cône $C = \{(x, y) \in \mathbb{R}^2_+ \mid x \leq 2y,\ y \leq 2x\}$ et une fonction f sur $\mathbb{R}^2$ définie par :

$$f(x,y) = \begin{pmatrix} f_1(x,y) \\ f_2(x,y) \end{pmatrix} = \begin{pmatrix} \min(2x,y) + x - (1/3)y \\ (2/3)\left(\min(y,(1/2)x) + y\right) \end{pmatrix}$$

La fonction f préserve le cône C, en effet, lorsque $(x, y) \in C$ on obtient $2f_1(x,y) - f_2(x,y) = 2y+2x-(2/3)y-(1/3)x-(2/3)y = (5/3)x+(2/3)y \geq 0$ et $f_2(x,y)-(1/2)f_1(x,y) = (1/3)x + (2/3)y - (1/2)y - (1/2)x + (1/6)y = -(1/6)x + (2/6)y = (1/6)(2y - x) \geq 0$.

Maintenant, soit $(x, y) \in C$ différent de $(0, 0)$, pour trouver les valeurs propres, on résout le système en fonction de λ :

$$\begin{aligned} x + (2/3)y &= \lambda x \\ (1/3)x + (2/3)y &= \lambda y \end{aligned} \tag{8.2}$$

Pour le système (8.2), x et y sont non-nuls, en effet, si x ou y est nul alors $(x, y) = (0, 0)$. On trouve que $\lambda = 4/3$ ou $\lambda = 1/3$ mais si $\lambda = 1/3$ le système (8.2) n'a pas de solution dans C et donc la seule valeur propre est $\lambda = 4/3$ qui admet pour vecteur propre $(2, 1) \in C$, on conclut que $\rho_C(f) = 4/3$.

Contre-exemple 8.1 (Fonction ne possédant aucune valeur propre)
Prenons tout $\mathbb{R}^d$ et soit l'application continue f qui à x associe $-x$. Supposons qu'il existe $\lambda \geq 0$ et $x \neq 0$ tel que $f(x) = \lambda x$ alors on a $-x = \lambda x$ ce qui équivaut à $-(1 + \lambda)x = 0$ d'où $\lambda = -1$ ce qui contredit la positivité de λ. La fonction f n'admet pas de valeur propre et donc $\rho_{\mathbb{R}^d}(f) = -\infty$.

Pour rester cohérent avec la théorie spectrale des matrices, on va montrer que le rayon spectral ainsi défini est toujours positif ou nul pour une classe bien précise de cônes. En effet, il n'est pas évident que $\{\lambda \geq 0 \mid \exists\, x \in C\backslash\{0\}, g(x) = \lambda x\}$ soit non vide. On va donc montrer que pour tout cône convexe fermé pointé C, pour toute fonction continue préservant C, il existe toujours un "vecteur propre" non nul et une "valeur propre" positive ou nulle. Le résultat est classique, on pourrait consulter [Nus88], nous en donnons une preuve pour la commodité du lecteur.

Définition 8.3 (*Cône convexe fermé pointé*)
Soit C un cône convexe fermé, on dit que C est pointé si $C \cap -C = \{0\}$.

Exemple 8.9
Les ensembles $\mathbb{R}^d_+$ et $\mathbb{R}^d_-$ sont des cônes convexes fermés pointés.

Exemple 8.10 (Le cône de Lorentz)
Le cône de Lorentz, c'est-à-dire, le cône défini par : $\{(x,t) \in \mathbb{R}^{d-1} \times \mathbb{R}_+ \mid \|x\| \leq t\}$ est un cône convexe fermé pointé.

Exemple 8.11 (Cône engendré par deux vecteurs non-colinéaires)
Soient $x, y \in \mathbb{R}^d$ deux vecteurs non colinéaires et le cône $C = \{c \in \mathbb{R}^d \mid c = \lambda x + \mu y, \lambda, \mu \geq 0\}$. Supposons qu'il existe $c \in C$ non nul tel que $-c \in C$, on a alors $c = \lambda x + \mu y$ et $-c = \lambda' x + \mu' y$ d'où $(\lambda + \lambda')x + (\mu + \mu')y = 0$ et comme c est non nul, $\lambda + \lambda' \neq 0$ et $\mu + \mu' \neq 0$ et finalement $x = -\dfrac{\lambda + \lambda'}{\mu + \mu'} y$ mais x et y ne sont pas colinéaires et on conclut que C est pointé.

Un cône pointé engendré par deux vecteurs non colinéaires est représenté par la figure 8.5.

Figure 8.5 – Un cône pointé engendré par deux vecteurs non colinéaires

Proposition 8.2 (*Existence de valeurs propres non linéaires*)
Si C est un cône convexe fermé pointé de $\mathbb{R}^d$ non réduit à $\{0\}$ alors pour tout $g : C \mapsto C$ continue, $\rho_C(g)$ est positif ou nul, c'est-à-dire que g admet une valeur propre positive et un vecteur propre non nul.

Pour montrer l'existence de valeur propre non-linéaire, nous commençons par démontrer le lemme suivant 8.1 :

Lemme 8.1
Si C est un cône convexe fermé pointé de $\mathbb{R}^d$ non réduit à $\{0\}$ alors il existe une forme linéaire non nulle ψ telle que : $\psi(c) > 0$ pour tout $c \in C \setminus \{0\}$.

Démonstration (Lemme 8.1)
Pour commencer, montrons que $C\setminus\{0\}$ est convexe 0. Ceci revient à montrer que 0 est un point extrême de C. Supposons qu'il existe, $x, y \in C$ tels que $x + y = 0$, on en déduit que $x = -y$ d'où, $x \in -C$, mais $x \in C$ d'où, $x = 0$ et on conclut que $y = 0$ donc 0 est un point êxtreme de C d'où $C\setminus\{0\}$ est convexe. On pose $D = \text{co}(C \cap S(0,1))$, où co désigne l'enveloppe convexe. Par définition de l'enveloppe convexe, on a $D \subseteq C\setminus\{0\}$. De plus, comme C est fermé non-vide alors D est compact non-vide. On peut donc séparer fortement $\{0\}$ et D et il existe une forme linéaire ψ non nulle et un réel α, tels que $\psi(x) > \alpha > 0$ pour tout $x \in D$. Maintenant soit $c \in C\setminus\{0\}$, alors $c(\|c\|)^{-1} \in D$ et on conclut que $\psi(c) > \alpha\|c\| > 0$ ce qu'il fallait démontrer. ∎

Démonstration (Proposition 8.2)
S'il existe x dans $C \setminus\{0\}$ tel que $g(x) = 0$, alors en prenant $\lambda = 0$, il existe $x \in C \setminus\{0\}$ tel que $g(x) = \lambda x$ d'où $\rho_C(g) \geq 0$.

Sinon, $\forall\, x \in C \setminus\{0\},\ g(x) \in C \setminus\{0\}$. En prenant la forme linéaire du lemme 8.1, on pose :

$$\Sigma = \{x \in C \mid \psi(x) = 1\} = C \cap \{x \in \mathbb{R}^d \mid \psi(x) = 1\}\ .$$

L'ensemble Σ est convexe fermé en tant qu'intersection d'ensembles convexes fermés, montrons que Σ est borné. Nous avons montré dans la preuve du lemme 8.1 que pour tout $c \in C \setminus\{0\}$ $\psi(c) > \alpha\|c\|$ pour un certain réel strictement positif α. On conclut que pour tout $c \in \Sigma$, $1 > \alpha\|c\|$ d'où $\|c\| < \alpha^{-1}$ et donc Σ est borné. Par conséquent, Σ est convexe compact (dimension finie) non vide et ne contient pas 0.

Pour $x \in \Sigma$, on pose $\hat{g}(x) = g(x)(\psi(g(x)))^{-1}$, puisque $g(x) \in C \setminus\{0\}$ pour tout $x \in C \setminus\{0\}$, $\psi(g(x)) \neq 0$ pour tout $x \in C \setminus\{0\}$. De plus, $\hat{g}$ est continue de Σ dans Σ, par le théorème de Brouwer, $\hat{g}$ possède un point fixe $\bar{x} \in \Sigma$ donc non nul.

On obtient par conséquent, $g(\bar{x}) = \psi(g(\bar{x}))\bar{x}$, donc il existe $\lambda > 0$ et $x \in C \setminus\{0\}$ tel que $g(x) = \lambda x$. En conclusion, $\rho_C(g)$ est positif ou nul pour toute fonction $g : C \mapsto C$ continue. ■

8.1.3 Semidifférentielle

Nous voulons étudier localement des fonctions non différentiables qui ne sont ni convexes ni concaves. Cependant, la structure d'opérateur dynamique de nos fonctions nous autorise à utiliser une notion non linéaire de la différentiabilité. Nous commençons par rappeler ce qu'est une application positivement homogène.

Définition 8.4 (*Fonctions positivement homogènes de degré 1*)

Soit C un cône, et soit g est application de C dans $\mathbb{R}^d$. La fonction g est dite (positivement) homogène (de degré un), si pour tout réel $\lambda > 0$, pour tout $x \neq 0$, $g(\lambda x) = \lambda g(x)$.

Exemple 8.12 (Applications linéaires)
Par définition, les applications linéaires sont homogènes.

Exemple 8.13 (Min Max d'applications linéaires)
Soit la fonction f qui associe à $x \in \mathbb{R}^d$, $f(x) = \min_{i \in I} \max_{j \in J} a^{ij} \cdot x$ où I, J dont des ensembles finis est une fonction homogène.

On va maintenant introduire une notion plus faible de différentiabilité qui reste compatible avec le rayon spectral.

Définition 8.5 (*Fonction semidifférentiable*)

Soit $f : \mathbb{R}^d \mapsto \mathbb{R}$. Soit $u \in \mathbb{R}^d$, f est dite semidifférentiable au point u s'il existe un voisinage $\mathcal{V}$ de 0 et une fonction g homogène continue tels que pour tout $h \in \mathcal{V}$:

$$f(u + h) = f(u) + g(h) + o(\|h\|)$$

L'application g est unique. Elle est appelée semidifférentielle de f au point u et est notée f'_u.

Remarque 8.1 Si g est homogène et continue sur un cône fermé C non vide non réduit à $\{0\}$ alors $g(0) = 0$. En effet, si on prend un point x de C et une suite $\lambda_n > 0$, $\lambda_n \to 0^+$ on a d'une part, $g(\lambda_n x) = \lambda_n g(x) \to 0$, par homogénéité et $g(x) < \infty$ et d'autre part, $g(\lambda_n x) \to g(0)$, par continuité.

Le contexte de la dimension finie nous permet d'étendre simplement la semidifférentiabilté aux fonctions à valeurs dans $\mathbb{R}^n$ et on dira qu'une application de $\mathbb{R}^d$ dans $\mathbb{R}^n$ est semidifférentiable si toutes ses fonctions coordonnées sont semidifférentiables.

Exemple 8.14 (Fonctions différentiables)
Les applications différentiables sont semidifférentiables et la semidifférentielle est la différentielle classique (voir [RW98, Corollary 7.22]).

Exemple 8.15 (Opérations linéaires sur les fonctions semidifférentiables)
Soient f et g deux applications semidifférentiables et $\mu \in \mathbb{R}$. La fonction $\mu f + g$ est semidifférentiable et la semidifférentielle en un point x est la fonction $h \mapsto \mu f'_x(h) + g'_x(h)$.

Soit $\varepsilon > 0$, il existe $\alpha_0, \alpha_1 > 0$ tels que : $\|h\| \leq \alpha_0$ implique $\|\mu f(x+h) - \mu f(x) - \mu f'_x(h)\| \leq \frac{\varepsilon}{2}\|h\|$ et $\|h\| \leq \alpha_1$ implique $\|g(x+h) - g(x) - g'_x(h)\| \leq \frac{\varepsilon}{2}\|h\|$. On en déduit en prenant $\alpha = \min(\alpha_0, \alpha_1)$ que $\|h\| \leq \alpha$ implique $\|(\mu f + g)(x+h) - (\mu f + g)(x) - (\mu f'_x + g'_x)(h)\| \leq \|\mu f(x+h) - \mu f(x) - \mu f'_x(h)\| + \|g(x+h) - g(x) - g'_x(h)\| \leq 2\frac{\varepsilon}{2}\|h\| = \varepsilon\|h\|$ d'où le résultat.

Exemple 8.16 (Min max finis d'applications affines)
Les applications min max finies d'applications affines sont semidifférentiables. Nous reviendrons sur cette classe d'applications un peu plus tard.

Considérons la fonction f sur $\mathbb{R}$ définie par, pour $x \in \mathbb{R}$, $f(x) = \min(\max(x+3, 0), 4)$,. Aux points $x = -3$ et $x = 1$, la fonction f n'est pas dérivable.

Mais :

$$f(-3+t) = f(-3) + \max(t, 0) + 0 \text{ et } f(1+t) = f(1) + \min(t, 0) + 0$$

Les fonctions $t \mapsto \max(t, 0)$ et $t \mapsto \min(t, 0)$ sont homogènes et continues, la fonction f est donc semidifférentiable en -3 et 1 et la semidifférentielle de f en -3 est $t \mapsto \max(t, 0)$ et $t \mapsto \min(t, 0)$ en 1. Le chemininement est décrit par la figure 8.6.

Exemple 8.17 (Fonctions convexes et fonctions concaves)
Les fonctions convexes à valeurs réélles sont semidifférentiables. Ce résultat est démontré par Rockafellar et Wets (voir [RW98, Theorem 7.26,Example 7.27]). De plus, la semidifférentielle d'une fonction convexe f en un point x est la fonction $y \mapsto \sup_{p \in \partial f(x)} p \cdot y$ où $\partial f(x)$ est le sous-différentiel de f en x, c'est-à-dire, l'ensemble $\{p \in \mathbb{R}^d \mid f(x) + p \cdot (y - x) \leq f(y),\ \forall\, y \in \mathbb{R}^d\}$.

Les fonctions concaves f sont également semidifférentiables puisque $-f$ l'est en tant que fonction convexe. De plus, la semidifférentielle d'une fonction concave f au point x est la fonction $y \mapsto \inf_{p \in \hat{\partial} f(x)} p \cdot y$ où $\hat{\partial} f = -\partial(-f)$.

Proposition 8.3 (*Continuité des applications semidifférentiables*)
Si f est semidifférentiable au point u alors f est continue en u.

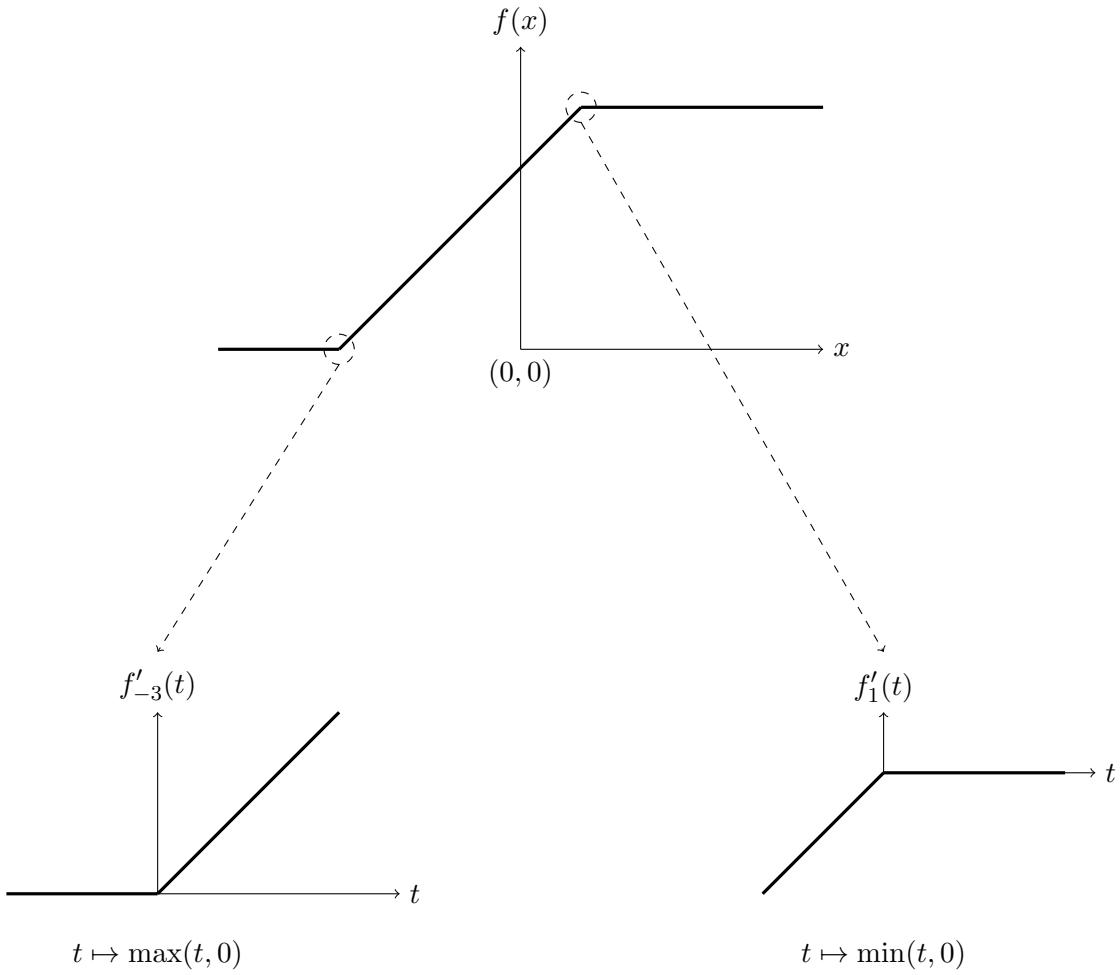


Figure 8.6 – Calcul de la semidifférentielle de la fonction f de l'exemple 8.16

Démonstration
Soit $u \in \mathbb{R}^d$, on suppose donc la fonction f est semidifférentiable au point u. Soit $\varepsilon > 0$, il existe $\alpha > 0$ tel que : $\|x-u\| \leq \alpha$ implique que $\|f(x) - f(u) - f'_u(x-u)\| \leq \varepsilon\|x-u\|$. Prenons une suite $(x_n)_{n\in\mathbb{N}}$ qui converge vers u. Soit $\varepsilon > 0$, il existe $N \in \mathbb{N}$, tel que $n \geq N$, implique $\|x_n - u\| \leq \alpha$. On a par conséquent, $\|f(x_n) - f(u)\| \leq \varepsilon\|x_n - u\| + \|f'_u(x_n - u)\|$ et comme f'_u est continue en 0 et que, par la remarque 8.1, $f'_u(0) = 0$, on a donc $(f(x_n))_n$ converge vers $f(u)$ et donc f est continue en u. ∎

On en déduit immédiatement la famille de contre-exemples suivants.

Remarque 8.2 (Applications non continues) Les applications non continues ne sont pas semidifférentiables aux points de discontinuités. En effet, si une fonction est semidifférentiable au point u, elle y est continue.

Remarque 8.3 Si f est semidifférentiable au point u, on a pour tout $t > 0$ et pour tout h dans un certain voisinage de 0 :

$$f(u+th) = f(u) + tf'_u(h) + o(t\|h\|)$$

il s'ensuit que

$$f'_u(h) = \lim_{t \to 0^+} \frac{f(u+th) - f(u)}{t}$$

elle coïncide avec la dérivée directionnelle unilatérale de f au point u dans la direction h.

Proposition 8.4 (*Semidifférentielle et limite uniforme*)

Soit $f : \mathbb{R}^d \mapsto \mathbb{R}$, semidifférentiable en $u \in \mathbb{R}^d$ de semidifférentielle f'_u si et seulement si

$$\lim_{\substack{t \to 0^+ \\ h' \to h}} \frac{f(u+th') - f(u)}{t} = f'_u(h).$$

Une démonstration de la proposition est donnée par Rockafellar et Wets dans [RW98, Theorem 7.21].

Remarque 8.4 Si pour une fonction f, la limite existe, on dit que la fonction admet une dérivée directionnelle au point u au sens de Hadamard (voir par exemple, [Roc84]).

8.2 Caractérisation du plus petit point fixe

Dans tout problème d'optimisation, on essaye d'introduire une condition du premier ordre (annulation de la dérivée, sous-différentiel contenant 0...). Ici, la condition du premier ordre sera donnée par la semidifférentielle. Dans la sous-section 8.2.1, nous énonçons les premiers résultats, le premier résultat est une condition suffisante de non-minimalité et le second fournit des conditions suffisantes de minimalité locale. Dans la sous-section 8.2.2, nous établissons de nouveaux résultats sur les fonctions affines par morceaux. Puis, à la sous-section 8.2.3, nous présentons les conditions nécessaires et suffisantes de minimalité dans le cas des fonctions croissantes et contractantes au sens large pour la norme infinie. Nous terminons cette section par la sous-section 8.2.4, par un raffinement de l'itération sur les politiques actuelle pour toujours calculer le plus petit point fixe fini d'une fonction croissante affine par morceaux et contractantes au sens large.

8.2.1 Premiers résultats

Dans tout ce chapitre, nous considérons une fonction f continue et dont l'ensemble des points fixes est non-vide.

Nous établissons un premier théorème qui, grâce au rayon spectral, donne un critère de non-minimalité d'un point fixe.

Théorème 8.2 (*Non-minimalité d'un point fixe*)

Soient $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ une application croissante et $u \in \mathrm{fix}(f)$. Nous supposons f semidifférentiable en u. De plus, nous supposons que $\{x \in \mathbb{R}^d \mid f(x) \leq x\}$ est inférieurement borné. Si $\rho_{\mathbb{R}^d_-}(f'_u) > 1$ alors u n'est pas le plus petit point fixe de f.

Démonstration

Par définition du rayon spectral, $\rho_{\mathbb{R}^d_-}(f'_u) > 1$ implique l'existence de $\mu > 1$ et de $h \in \mathbb{R}^d_- \setminus \{0\}$ tels que $f'_u(h) = \mu h$. On pose $\varepsilon := \min_{h_i < 0} -\|h\|^{-1}(\mu - 1)h_i$. Par définition de la semidifférentielle,

il existe $t_0 > 0$, tel que, pour tout $t \in]0, t_0[$, $(\|f(u+th) - f(u) - tf'_u(h)\|)(t\|h\|)^{-1} \leq \varepsilon$ et de $u \in \text{fix}(f)$ et $f'_u(h) = \mu h$, on en déduit qu'il existe $t_0 > 0$, tel que, pour tout $t \in]0, t_0[$, $f_i(u+th) \leq u_i + t\mu h_i + t\varepsilon\|h\|$, pour tout i tel que $h_i < 0$. Par définition de ε, on obtient, pour tout i tel que $h_i < 0$, $f_i(u+th) \leq u_i + t\mu h_i - t\|h\|^{-1}(\mu - 1)\|h\| = u_i + th_i < u_i$. De plus, par croissance de f on a $f(u+th) \leq u$ pour tout $t \geq 0$ d'où, pour tout i tel que $h_i = 0$, $f_i(u+th) \leq u_i + th_i$.

On conclut que $f(u+th) < u + th$ puis que $(f^k(u+th))_{k\in\mathbb{N}}$ est une suite décroissante et puisque l'ensmeble $\{x \in \mathbb{R}^d \mid f(x) \leq x\}$ est inférieurement borné, la suite est minorée donc convergente. Par continuité de f, la limite de $(f^k(u+th))_{k\in\mathbb{N}}$ est point fixe de f qui est strictement plus petit que u. ■

Définition 8.6 (*Point fixe localement minimal*)

Un point fixe u est dit localement minimal s'il existe un voisinage $\mathcal{V}$ de u tel que :

$$\forall\ v \in \mathcal{V},\ f(v) = v \text{ et } v \leq u \implies v = u.$$

En d'autres termes, un point fixe est localement minimal s'il existe un voisinage pour lequel tout point fixe *comparable* est plus grand.

Exemple 8.18

Posons pour $x \in \mathbb{R}$, $f(x) = \sup(-4, x, 2x-3)$. La fonction est représentée par la figure 8.7. La fonction est en gras sur la figure et la partie hachurée représente l'intersection du graphe de f et celui de l'identité. On en déduit que l'ensemble des points fixes de f, fix(f), est $[-4, 3]$. Le point fixe est -4 est localement minimal.

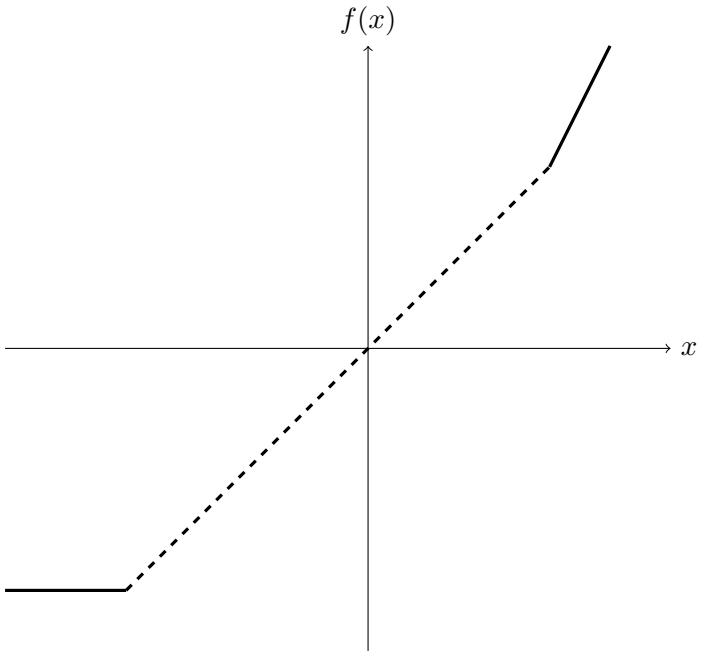


Figure 8.7 – La fonction f de l'exemple 8.18

Exemple 8.19
Soit la fonction f de $\mathbb{R}^2$ dans $\mathbb{R}^2$ définie par :

$$f(x,y) = \begin{pmatrix} f_1(x,y) \\ f_2(x,y) \end{pmatrix} = \begin{pmatrix} \max(\min(x,y),0) + 2\max(\min(x+1,0),-2) \\ \max(\max(x,y),0) + 2\max(\min(y+1,0),-2) \end{pmatrix}$$

On peut étudier la fonction f en découpant $\mathbb{R}^2$ sur différents ensembles. La décomposition de la fonction f est donnée par le tableau 8.1.

		$x > y$	$x \leq y$
$x \leq -3$	$y \leq -3$	$f_1(x,y) = -4$ $f_2(x,y) = -4$	$f_1(x,y) = -4$ $f_2(x,y) = -4$
	$y \in]-3,-1[$	Non définie	$f_1(x,y) = -4$ $f_2(x,y) = 2y+2$
	$y \geq -1$	Non définie	$f_1(x,y) = -4$ $f_2(x,y) = \max(y,0)$
$x \in]-3,-1[$	$y \leq -3$	$f_1(x,y) = 2x+2$ $f_2(x,y) = -4$	Non définie
	$y \in]-3,-1[$	$f_1(x,y) = 2x+2$ $f_2(x,y) = 2y+2$	$f_1(x,y) = 2x+2$ $f_2(x,y) = 2y+2$
	$y \geq -1$	Non définie	$f_1(x,y) = 2x+2$ $f_2(x,y) = \max(y,0)$
$x \geq -1$	$y \leq -3$	$f_1(x,y) = 0$ $f_2(x,y) = \max(x,0) - 4$	Non définie
	$y \in]-3,-1[$	$f_1(x,y) = 0$ $f_2(x,y) = \max(x,0) + 2y + 2$	Non définie
	$y \geq -1$	$f_1(x,y) = \max(y,0)$ $f_2(x,y) = \max(x,0)$	$f_1(x,y) = \max(x,0)$ $f_2(x,y) = \max(y,0)$

TABLE 8.1 – Décomposition de la fonction f de l'exemple 8.19

Grâce au tableau 8.1, nous pouvons calculer les points fixes de la fonction f. Les points fixes de f sont donnés sous la forme d'un tableau 8.2

En conclusion, l'ensemble des points fixes de f est $\{(-4,-4)\} \cup \{(-4,-2)\} \cup \{-4\} \times \mathbb{R}_+ \cup \{(-2,-4)\} \cup \{(-2,-2)\} \cup \{-2\} \times \mathbb{R}_+ \cup \{0,-4\} \cup \{(0,-2)\} \cup \{(x,y) \in \mathbb{R}^2 \mid y \leq x \leq 0\}$. L'ensemble des points fixes de f est représenté par la figure 8.8. Les points fixes $(-4,-4)$, $(-4,-2)$, $(-2,-4)$, $(-2,-2)$, $(0,-4)$ et $(0,-2)$ sont localement minimaux car ce sont des points fixes isolés i.e pour ces points fixes, il existe un voisinage sur lequel il n'existe pas d'autre point fixe. Les points fixes $(-4,0)$, $(-2,0)$, $(0,0)$ sont également localement minimaux.

Remarque 8.5 (Globalement minimal et plus petit point fixe) Par extension, on pourrait définir une notion de points fixes globalement minimaux par :

$$\forall\, v \in \mathbb{R}^d,\ f(v) = v \text{ et } v \leq u \implies v = u.$$

Cependant si le plus petit point fixe $\mathrm{lfp}(f)$ est fini alors $\mathrm{lfp}(f)$ est l'unique point fixe globalement minimal. En effet, il est clair que si $\mathrm{lfp}(f) \in \mathbb{R}^d$ alors il est globalement minimal. Maintenant prenons un point fixe u, s'il était globalement minimal alors, comme $\mathrm{lfp}(f) \leq u$, on

		$x > y$	$x \leq y$
$x \leq -3$	$y \leq -3$	Aucun point fixe	$x = -4$ $y = -4$
	$y \in]-3, -1[$	Non définie	$x = -4$ $y = -2$
	$y \geq -1$	Non définie	$x = -4$ $y \geq 0$
$x \in]-3, -1[$	$y \leq -3$	$x = -2$ $y = -4$	Non définie
	$y \in]-3, -1[$	Aucun point fixe	$x = -2$ $y = -2$
	$y \geq -1$	Non définie	$x = -2$ $y \geq 0$
$x \geq -1$	$y \leq -3$	$x = 0$ $y = -4$	Non définie
	$y \in]-3, -1[$	$x = 0$ $y = -2$	Non définie
	$y \geq -1$	Aucun point fixe	$x \geq 0$ $y \geq 0$

TABLE 8.2 – Points fixes de la fonction f de l'exemple 8.19

aurait $u = \text{lfp}(f)$. Nous remarquons donc que la globale minimalité dans notre cas correspond à la minimalité du point fixe.

Théorème 8.3 (*Conditions suffisantes de locale minimalité*)

Soit $f : \mathbb{R}^d \mapsto \mathbb{R}^d$. Soit $u \in \text{fix}(f)$. On suppose, de plus, que f est semidifférentiable en u.

1. u est localement minimal ;
2. f'_u n'admet pas de point fixe non nul dans $\mathbb{R}^d_-$;
3. $\rho_{\mathbb{R}^d_-}(f'_u) < 1$.

On a $3 \implies 2 \implies 1$.

Démonstration

$3 \implies 2$ Par définition du rayon spectral.

$2 \implies 1$ On va montrer en réalité, $\neg 1 \implies \neg 2$. Supposons que u ne soit pas localement minimal : $\forall\, r > 0,\ \exists\, h \in B(0,r) \cap \mathbb{R}^d_-$, $h \neq 0$, tel que $u + h \in \text{fix}(f)$. Prenons une suite $\{r_n\}_{n\geq 0}$ tendant vers 0. Pour tout n, il existe $h_n \in B(0, r_n) \cap \mathbb{R}^d_-$ tel que $u + h_n \in \text{fix}(f)$ avec $h_n \neq 0$. Posons, $y_n = h_n \|h_n\|^{-1} \in S(0,1) \cap \mathbb{R}^d_-$ qui est compact (dimension finie), et quitte à extraire, il existe $\bar{y}$ tel que $\|\bar{y}\| = 1$ et $\bar{y} \in \mathbb{R}^d_-$ et $y_n \to \bar{y}$ quand $n \to +\infty$. On obtient par la proposition 3.1.3

$$\lim_{n\to+\infty} \left\| \frac{f(u + \|h_n\| y_n) - f(u)}{\|h_n\|} - f'_u(\bar{y}) \right\| = 0$$

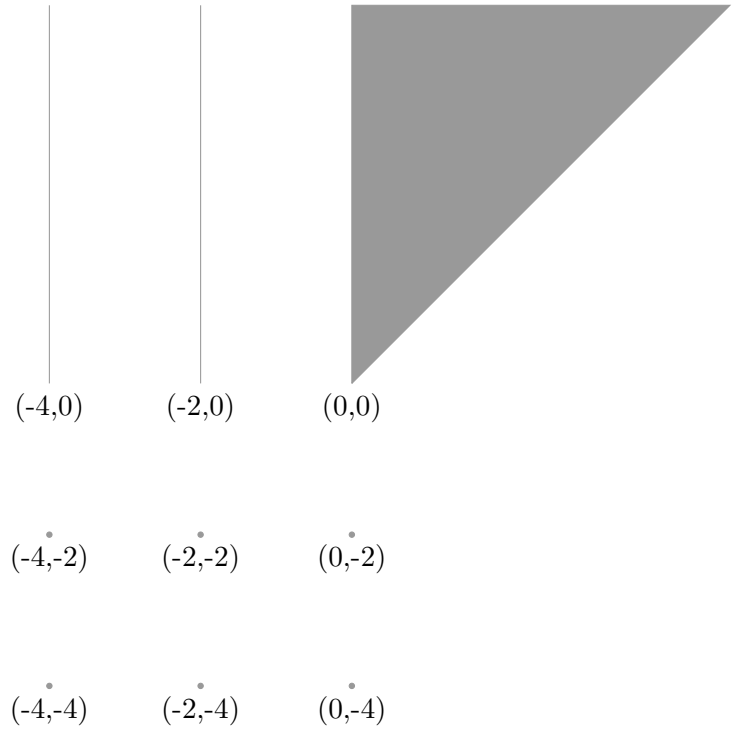


Figure 8.8 – Ensemble de points fixes de l'exemple 8.19

d'où, comme u et $u + \|h_n\|y_n$ sont dans $\mathrm{fix}(f)$,

$$\lim_{n\to+\infty} \left\| \frac{u + \|h_n\|y_n - u}{\|h_n\|} - f'_u(\bar{y}) \right\| = 0 \text{ puis, } \lim_{n\to+\infty} \left\| y_n - f'_u(\bar{y}) \right\| = 0$$

ce qui entraîne,

$$\|\bar{y} - f'_u(\bar{y})\| = 0.$$

Finalement, on a montré qu'il existe $\bar{y}$ non nul à coordonnées négatives qui est un point fixe de f'_u. ■

Exemple 8.20 (Application du théorème 8.3 pour l'exemple 8.18)
On reprend la fonction f de l'exemple 8.18.

Au point -4, la semidifférentielle est $t \mapsto \sup(0, t)$, sur $\mathbb{R}_-$, cette fonction est la fonction identiquement nulle, elle n'admet donc aucun point fixe négatif non nul et le rayon spectral est égal à 0.

Soit $x \in]-4, 3[$, la fonction f est l'identité, elle est donc (semi)différentiable et la (semi)différentielle est la fonction identité et donc elle admet un point fixe négatif non nul, de plus, le rayon spectral est égal à 1.

En 3, la semidifférentielle vaut $t \mapsto \sup(2t, t)$, si $t \leq 0$ alors la semidifférentielle restreinte à $\mathbb{R}_-$ est l'identité, elle admet, donc, un point fixe négatif et le rayon spectral vaut 1.

Exemple 8.21 (Application du théorème 8.3 pour l'exemple 8.19)
On reprend la fonction f de l'exemple 8.19. En tout point fixe, la fonction f est semidifférentiable car au voisinage des points fixes, la fonction f est une somme de fonctions semidifférentiables. Nous nous intéressons à trois points fixes : $(-4,-4)$, $(-2,0)$ et $(0,0)$.

En $u_1 = (-4,-4)$ et au voisinage de ce point, la fonction f est égale à la fonction constante $(x,y) \mapsto (-4,-4)$, la semidifférentielle f'_{u_1} est la fonction nulle. Le rayon spectral de f'_{u_1} est 0 et la semidifférentielle n'admet pas de point fixe négatif non-nul, le point fixe $u_1 = (-4,-4)$ est localement minimal.

En $u_2 = (-2,0)$ et au voisinage de ce point, la fonction f est égale à la fonction $(x,y) \mapsto (2x+2, \max(y,0))$, elle est donc semidifférentiable et la semidifférentielle f'_{u_1} est la fonction $(h_1,h_2) \mapsto (2h_1, \max(h_2,0)$. Le rayon spectral sur $\mathbb{R}^2_-$ de f'_{u_2} est 2, on conclut d'après le théorème 8.2, ce point fixe n'est pas le plus petit point fixe de f. Mais, comme f'_{u_2} n'admet pas de point fixe négatif non nul, le point fixe $u_2 = (-2,0)$ est donc localement minimal.

En $u_3 = (0,0)$ et au voisinage de ce point, la fonction f est égale à la fonction $(x,y) \mapsto (\max(\min(x,y),0), \max(\max(x,y),0))$, elle est donc semidifférentiable et la semidifférentielle f'_{u_3} est la fonction $(h_1,h_2) \mapsto (\max(\min(h_1,h_2),0), \max(\max(h_1,h_2),0))$. Le rayon spectral de f'_{u_3} est 0 et la semidifférentielle n'admet pas de point fixe négatif non-nul, le point fixe $u_3 = (0,0)$ est donc localement minimal.

8.2.2 Applications affines par morceaux

Le théorème de Danskin [Dan67] permet de différentier des maxima et minima d'applications différentiables. Nous utilisons ce théorème pour prouver que le calcul de la semidifférentielle est facile pour ce type d'applications et obtenons un résultat plus fort en montrant que les applications affines par morceaux ont un développement limité exact.

Définition 8.7 (*Applications affines par morceaux version géométrique*)
On dit que f est affine par morceaux si elle est continue et s'il existe une famille finie de convexes fermés $\{C_i\}_{i\in I}$, telle que

$$\mathbb{R}^d \subseteq \bigcup_{i\in I} C_i,\ \forall\, l,k \in I,\ l \neq k,\ \operatorname{int}(C_l) \cap \operatorname{int}(C_k) = \emptyset. \tag{8.3}$$

et $f_{|C_i}$ est affine pour tout $i \in I$ i.e pour tout $j \in \{1,\dots,d\}$, $f_{j|C_i}$ est affine.

Il serait plus intuitif de supposer que les C_i soient des polyèdres mais S.Ovchinnikov montre qu'on obtient les mêmes classes de fonctions avec des convexes fermés d'intérieur non vides [Ovc02, AT07].

Après cette définition géométrique d'affine par morceaux, on va donner une définition équivalente, plus analytique, qui nous rapprochera des fonctions étudiées par la suite.

Lemme 8.2 (*Applications affines par morceaux version analytique*)
Une application f est affine par morceaux ssi pour tout $j \in \{1,\dots,d\}$, il existe un ensemble fini A_j, une famille d'ensembles finis $\{B_a\}_{a\in A_j}$ et une famille de fonctions affines $\{g_{a,b}\}_{a\in A_j, b\in B_a}$ vérifiant :

$$f_j = \min_{a\in A_j} \max_{b\in B_a} g_{a,b}$$

Une démonstration de ce lemme peut se trouver dans [Ovc02, AT07]. Dans [AT07], les auteurs donnent un algorithme explicite pour passer de la forme géométrique à la forme

analytique et vice-versa.

Proposition 8.5 (*Semidifférentielle d'applications affines par morceaux*)

Si f est affine par morceaux alors, f est semidifférentiable pour tout $u \in \mathbb{R}^d$. De plus, on a pour tout $h \in \mathbb{R}^d$, pour tout $j \in \{1, \dots, d\}$:

$$(f'_u)_j(h) = \min_{a \in \bar{A}} \max_{b \in \bar{B}_a} m_{a,b} \cdot h$$

où $\bar{A} = \{a \in A \mid f_j(u) = \max_{b \in B_a} g_{a,b}(u)\}$ et $\bar{B}_a = \{b \in B_a \mid f_j(u) = \min_{a \in A} g_{a,b}(u)\}$.

Pour la démonstration, on aurait pu utiliser le théorème de semidifférentiation sous le signe sup énoncé par T.Rockafellar et J-B.Wets dans [RW98, Exercice 10.27], dans le cas où les inf et sup portent sur un nombre fini de fonctions affines. Pour un résultat plus général, le lecteur est invité à consulter [AGN].

Ici, on va montrer la proposition 8.5 d'une autre manière : on va montrer que, pour la semidifférentiabilité, les applications affines par morceaux jouent le même rôle que les applications affines pour la différentiabilité classique. En effet, le "développement limité" à l'ordre 1 au sens semidifférentiel du terme d'une application affine par morceaux est exact.

Lemme 8.3 (*Développement limité exact des applications affines par morceaux*)

Soit $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ une fonction affine par morceaux. Soit $u \in \mathbb{R}^d$. Il existe un voisinage $\mathcal{V}$ de u tel que pour tout $u + h \in \mathcal{V}$,

$$f(u + h) = f(u) + f'_u(h)$$

Démonstration

Puisque f est affine par morceaux pour tout $j \in \{1, \dots, d\}$, il existe un ensemble fini A, des ensembles finis $\{B_a\}_{a \in A}$ et une famille finie de fonctions affines $g_{a,b}$ tels que :

$$f_j = \min_{a \in A} \max_{b \in B_a} g_{a,b}$$

Soient $\overline{A} = \{a \in A \mid f_j(u) = \max_{b \in B_a} g_{a,b}(u)\}$ et pour $a \in \overline{A}$, on pose $\overline{B_a} = \{b \in B_a \mid f_j(u) = g_{a,b}(u)\}$.

Il existe $\mathcal{V}_j$ tel que $f_{j|\mathcal{V}_j} = \min_{a \in \overline{A}} \max_{b \in \overline{B_a}} g_{a,b}$. Soit $h \in \mathbb{R}^d$ tel que $u + h \in \mathcal{V}_j$. On a, par conséquent :

$$f_j(u + h) = \min_{a \in \bar{A}} \max_{b \in \overline{B_a}} g_{a,b}(u + h) \text{ qui équivaut à } f_j(u + h) = \min_{a \in \overline{A}} \max_{b \in \overline{B_a}} (g_{a,b}(u) + m_{a,b} \cdot h)$$

Comme $\min_{a \in \overline{A}} \max_{b \in \overline{B_a}} g_{a,b}(u) = f_j(u)$, on conclut que :

$$f_j(u + h) = f_j(u) + \min_{a \in \overline{A}} \max_{b \in \overline{B_a}} (m_{a,b} \cdot h) \ .$$

Finalement, on a $(f_j)'_u(h) = \min_{a \in \overline{A}} \max_{b \in \overline{B_a}} (m_{a,b} \cdot h)$ avec $a \in \overline{A}$ et $b \in \overline{B_a}$, $f_j(u + h) = f_j(u) + (f_j)'_u(h)$. Puis, en prenant $\mathcal{V} = \cap_j \mathcal{V}_j$, on conclut que $f(u + h) = f(u) + f'_u(h)$. ∎

Corollaire 8.1 (*Direction de descente*)

Soit $f : \mathbb{R}^d \mapsto \mathbb{R}^d$, et soit $u \in \text{fix}(f)$, si, de plus, f est affine par morceaux, alors u est localement minimal si et seulement si f'_u n'admet pas de point fixe non nul dans $\mathbb{R}^d_-$.

Démonstration

On a déjà montré au théorème 8.3 que le fait que f'_u n'admette pas de point fixe négatif non-nul impliquait que u était localement minimal.

Maintenant, supposons que f'_u admette un point fixe h_0 négatif non-nul et soit un voisinage $\mathcal{V}$ de u. D'après le lemme 8.3, il existe un voisinage $\mathcal{V}_0$ de u tel que $f(u+h) = u + f'_u(h)$ pour tout h tel que $u + h \in \mathcal{V}_0$. Il existe $\alpha > 0$ tel que $u + \alpha h_0 \in \mathcal{V}_0 \cap \mathcal{V}$ et d'où, par homogénéité de f'_u, $f(u + \alpha h_0) = u + \alpha h_0 < u$ et donc u n'est pas localement minimal. ∎

Exemple 8.22 (Suite de l'exemple 8.18)

On reprend la fonction f de (8.18). Cette fonction est clairement affine par morceaux d'après le lemme 8.2. En -4, la semidifférentielle vaut $t \mapsto \sup(0, t)$, elle n'admet pas de point fixe négatif. En tout point fixe $x \in]-4, -3]$, la semidifférentielle f'_x admet un point fixe négatif non-nul.

Exemple 8.23 (Suite de l'exemple 8.19)

Nous reprenons la fonction de l'exemple 8.19. Le tableau 8.1 donne une famille de convexes fermés $(C_i)_{i\in I}$ dont les intérieurs sont disjoints (où il faut considérer $x \geq y$ à la place de $x > y$, découper $x \geq -1$ en $x \geq 0$ et $x \in [-1, 0]$, remplacer $y \in]-3, -1[$ par $y \in [-3, -1]$...) sur lequels, la fonction f est affine et donc f est affine par morceaux par définition.

On a déjà vu dans l'exemple 8.21 qu'aux points fixes $(-4, -4)$, $(-2, 0)$ et $(0, 0)$, les semidifférentielles n'avaient pas de points fixes négatifs non nuls. On peut montrer que pour tout point fixe u localement minimal de f'_u n'admet pas de point fixe négatif non-nul.

Soit un point fixe (u_1, v_1) tel que $u_1 = -4$ et $v_1 > 0$, la fonction f au voisinage de ce point fixe est la fonction $(x, y) \mapsto (-4, y)$ et donc la semidifférentielle au point fixe (u_1, v_1) est la fonction $(h_1, h_2) \mapsto (0, h_2)$, cette fonction admet, par exemple, $(0, -1)$ comme point fixe négatif non-nul. On trouve, de même, $(0, -1)$ comme point fixe négatif non-nul pour la semidifférentielle aux points fixes (u_2, v_2) tels que $u_2 = -2$ et $v_2 > 0$.

Maintenant, considérons, (u_3, v_3) tels que $v_3 \geq u_3 > 0$. On peut trouver un voisinage de (u_3, v_3) telle que f soit égale à $(x, y) \mapsto (\min(x, y), \max(x, y))$ la semidifférentielle au point (u_3, v_3) est $(h_1, h_2) \mapsto (\min(h_1, h_2), \max(h_1, h_2))$, cette fonction admet donc, par exemple, $(-1, -1)$ comme point fixe négatif non-nul.

Maintenant, considérons, (u_4, v_4) tels que $v_4 > 0$ et $u_4 = 0$. On peut trouver un voisinage de (u_3, v_3) telle que f soit égale à $(x, y) \mapsto (\max(x, 0), y)$ la semidifférentielle au point (u_4, v_4) est $(h_1, h_2) \mapsto (\max(h_1, 0), h_2)$, cette fonction admet donc, par exemple, $(0, -1)$ comme point fixe négatif non-nul.

8.2.3 Applications contractantes au sens large

Dans cette partie, on trouvera souvent des additions de vecteurs et d'une constante, en réalité, il s'agira de l'addition d'un vecteur et du vecteur dont toutes les coordonnées sont égales à cette constante. De même pour les inégalités entre des vecteurs et des constantes, on écrira $z \leq \mu$ avec μ dans $\mathbb{R}$ pour dire que z est plus petit que le vecteur dont toutes les coordonnées sont égales à μ, le contexte précisera quelles sont les additions et inégalités classiques et les additions et inégalités constantes-vecteurs.

Définition 8.8 (*Fonction contractante au sens large*)
Soit $f : \mathbb{R}^d \mapsto \mathbb{R}^d$, f est dite contractante au sens large si pour tout $x, y \in \mathbb{R}^d$, $\|f(x) - f(y)\| \leq \|x - y\|$.

Par la suite, nous dirons simplement contractante à la place de contractante au sens large.

Remarque 8.6 Une fonction f est contractante au sens large ssi toutes les coordonnées f_i sont contractantes au sens où $|f_i(x) - f_i(y)| \leq \|x - y\|$.

Exemple 8.24 (Applications croissantes sous-additivement homogène)
On dit qu'une fonction $g : \mathbb{R}^d \mapsto \mathbb{R}^d$ est sous-additivement homogène si pour tout $t \in \mathbb{R}_+$, pour tout $u \in \mathbb{R}^d$, $g(u + t) \leq g(u) + t$, la somme $u + t$ est égale au vecteur qui a pour coordonnée $u_i + t$. Automatiquement, on a $g(u) - t \leq g(u - t)$ pour tout $u \in \mathbb{R}^d$, pour tout $t \in \mathbb{R}^+$. Pour cela, il suffit de voir que $g(u - t + t) \leq g(u - t) + t$.

Soit g une fonction monotone et sous-additivement homogène. Soit $x, y \in \mathbb{R}^d$, on a donc par définition de la norme infinie $-\|x - y\| \leq x - y \leq \|x - y\|$ (au sens où toutes les composantes de $x - y$ sont encadrées par $-\|x - y\|$ et $\|x - y\|$). Par croissance de f, on obtient $f(y - \|x - y\|) \leq f(x) \leq f(y + \|x - y\|)$ et finalement par sous-additivité homogène, on conclut que $f(y) - \|x - y\| \leq f(x) \leq f(y) + \|x - y\|$ d'où $\|f(x) - f(y)\| \leq \|x - y\|$.

Exemple 8.25 (Limite simple de fonctions contractantes)
Soit $(g_n)_{n\geq 0}$ une suite de fonctions contractantes convergeant simplement vers une fonction g. La fonction g est contractante.

En effet, soit $n \in \mathbb{N}$ et $x, y \in \mathbb{R}^d$, $\|g_n(x) - g_n(y)\| \leq \|x - y\|$ d'où $g_n(x) \leq \|x - y\| + g_n(y)$ et $g_n(y) + \leq \|x - y\| + g_n(x)$ puis par passage à la limite $g(x) \leq \|x - y\| + g(y)$ et $g(y) \leq \|x - y\| + g(x)$ et finalement $\|g(x) - g(y)\| \leq \|x - y\|$.

Exemple 8.26 (Inf sup d'applications contractantes)
Soient S, T deux ensembles non vides et $(f_{s,t})_{\substack{s\in S \\ t\in T}}$ une famille d'applications contractantes. La fonction $f = \inf_{s\in S} \sup_{t\in T}$ telle que $f(x) \in \mathbb{R}$ est contractante.

En effet, soient $s, t \in S \times T$, $x, y \in \mathbb{R}^d$, $\|f_{s,t}(x) - f_{s,t}(y)\| \leq \|x - y\|$ d'où $f_{s,t}(x) \leq \|x - y\| + f_{s,t}(y)$ et $f_{s,t}(y) + \leq \|x - y\| + f_{s,t}(x)$ puis en prenant successivement le supremum sur T puis l'infimum sur S on obtient $f(x) \leq \|x - y\| + f(y)$ et $f(y) \leq \|x - y\| + f(x)$ et finalement $\|f(x) - f(y)\| \leq \|x - y\|$.

Exemple 8.27 (Applications affines à vecteur directeur sous stochastique)
Soient $i \in \{1, \ldots, d\}$ et $a^i \in \mathbb{R}^d$ tel que $\mathbf{a}^i_j \geq 0$ et $\sum_{j=1} a^i_j \leq 1$ et $b \in \mathbb{R}^d$. La fonction f définie, composante par composante $x \mapsto f_i(x) = a^i \cdot x + b_i$ est contractante. Soient $x, y \in \mathbb{R}^d$,
$\|f(x) - f(y)\| = \sup_{i\in\{1,\ldots,d\}} \sum_{j=1}^{d} a^i_j (x_i - y_i) \leq \sup_{i\in\{1,\ldots,d\}} \sum_{j=1}^{d} a^i_j \|x - y\| \leq \|x - y\|$.

Contre-exemple 8.2 (Somme de variables)
Soit la fonction f qui à $x \in \mathbb{R}^d$ associe $\sum_{i=1} x_i$ sur la première composante et 0 sinon. En prenant $x \equiv 1$, le vecteur dont toutes les coordonnées valent 1 et y le vecteur nul, on a
$\|f(x) - f(y)\| = |\sum_{i}^{d} x_i| = d > \|x - y\| = \|x\| = \sup_{i=1,\ldots,d} |x_i| = 1$.

Rappelons que la norme $\|\cdot\|$ définit la norme infinie, la contraction s'entend donc au sens de cette norme.

Proposition 8.6 (*Semidifférentielle d'une contraction*)

Soit $g : \mathbb{R}^d \to \mathbb{R}^d$ une fonction contractante et soit $x \in \mathbb{R}^d$. Si g est semidifférentiable en x, alors g'_x est contractante.

Démonstration

Par définition de la semdifférentielle, g'_x est la limite simple de $t_n^{-1}(g(x+t_nh)-g(x))$ en prenant $(t_n)_{n\in\mathbb{N}}$ une suite qui tend vers 0. D'après l'exemple 8.25, la fonction g'_x est contractante au sens large. ∎

La contraction d'une fonction assure que le rayon spectral de la semidifférentielle ne dépasse pas 1. Ce résultat qui va être utilisé pour en déduire une équivalence entre le rayon spectral et l'ensemble des points fixe non nul de la semidifférentielle.

Proposition 8.7

Soit $f : \mathbb{R}^d \to \mathbb{R}^d$ une fonction contractante au sens large et soit $u \in \mathbb{R}^d$. Si, de plus, f est semidifférentiable en u alors f'_u n'a pas de point fixe dans $\mathbb{R}^d_- \backslash \{0\}$ ssi $\rho_{\mathbb{R}^d_-}(f'_u) < 1$.

Démonstration

Il suffit de montrer que $\rho_{\mathbb{R}^d_-}(f'_u) \leq 1$. Tout d'abord, remarquons que comme la fonction f est une contraction au sens large, alors, d'après la proposition 8.6, f'_u est aussi une contraction au sens large. Supposons $\rho_{\mathbb{R}^d_-}(f'_u) > 1$. Il existe $\mu > 1$ et $x \in \mathbb{R}^d_- \backslash \{0\}$, tels que $f'_u(x) = \mu x$. D'où $\|f'_u(x) - f'_u(0)\| = \|f'_u(x)\| = \mu\|x\| > \|x\|$, ce qui contredit le fait que f'_u soit contractante. ∎

La contraction est une hypothèse cruciale pour passer de la minimalité locale à la minimalité globale c'est-à-dire, dans notre cas, nous allons montrer que le seul point fixe localement minimal est le plus petit point fixe.

Théorème 8.4 (*Unicité du point fixe localement minimal*)

Soit $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ et soit $u \in \mathrm{fix}(f)$, supposons f contractante au sens large alors u est localement minimal si et seulement si u est le plus petit point fixe de f.

Pour prouver ce théorème, nous prouvons l'existence d'un retract non-linéaire. Cette approche a déjà été utilisée dans [CGG+05] pour prouver qu'un point fixe était le plus petit point fixe dès qu'il était atteint par une unique politique.

Lemme 8.4 (*Retract non-linéaire*)

Il existe une application $P : \mathbb{R}^d \to \mathrm{fix}(f)$ croissante contractante qui vérifie $P^2 = P$ et $\mathrm{fix}(P) = \mathrm{fix}(f)$.

Démonstration

On suit l'idée de Gaubert et Gunawardena dans [GG04], nous proposons de construire la fonction P comme suit : $P(x) = \lim_{k\to+\infty} f^k(y)$ où $y = \limsup_{l\to+\infty} f^l(x)$. Comme f est contractante au sens large et que $u \in \mathrm{fix}(f)$ alors pour tout $k \in \mathbb{N}$ et pour tout $x \in \mathbb{R}^d$, $\|f^k(x) - f^k(u)\| \leq \|x - u\|$, donc $\|f^k(x) - u\| \leq \|x - u\|$, par conséquent $\{f^k(x)\}_{k\in\mathbb{N}}$ est bornée pour tout $x \in \mathbb{R}^d$.

On pose pour tout $x \in \mathbb{R}^d$: $Q(x) = \inf_{k\geq 1} \sup_{n\geq k} f^n(x) = \lim_{k\to+\infty} \sup_{m\geq k} f^m(x)$. Comme $\{f^k(x)\}_{k\in\mathbb{N}}$ est bornée, $Q(x)$ existe et est unique.

Soit $x \in \mathbb{R}^d$ et $k \in \mathbb{N}$ $\sup_{n\geq k} f^n(x) \geq f^m(x)$ pour tout $m \geq k$ puis comme f est croissante, $f(\sup_{n\geq k} f^n(x)) \geq f(f^m(x))$ pour tout $m \geq k$ et en prenant le sup sur $m \geq k$ à droite on obtient $f(\sup_{n\geq k} f^n(x)) \geq \sup_{m\geq k} f(f^m(x))$ en faisant tendre k vers $+\infty$, et en utilisant la continuité de f, on obtient $f(Q(x)) \geq Q(x)$.

Finalement, par croissance de f, on déduit que $\{f^l(Q(x))\}_{l\in\mathbb{N}}$ est une suite croissante majorée car bornée, elle converge et on peut poser :

$$P(x) = \lim_{l\to+\infty} f^l(Q(x))$$

La fonction f est contractante, donc Q est contractante en tant que inf sup d'applications contractantes (exemple 8.26) et P est contractante en tant que limite simple d'applications contractantes (exemple 8.25). De plus, par construction P est croissante car f.

Enfin $f(P(x)) = f(\lim f^l(x)) = \lim f^{l+1}(x) = P(x)$, ce qui entraîne que pour tout $x \in \mathbb{R}^d$, $P(x) \in \mathrm{fix}(f)$. Par ailleurs,

$$P(P(x)) = \lim_{l\to+\infty} f^l[\limsup_{k\to+\infty} f^k(P(x))]$$

d'où $P(P(x)) = P(x)$ puisque $P(x) \in \mathrm{fix}(f)$.

On a montré que $\mathrm{fix}(P) \subseteq \mathrm{fix}(f)$, l'inclusion inverse est évidente. ∎

Démonstration (Théorème 8.4)
Prenons une application P comme dans le lemme 8.4. On suppose que u est localement minimal mais pas le plus petit point fixe. Il existe $v \in \mathrm{fix}(f)$, tel que $\inf(v, u) \neq u$. On pose $\omega_t = \inf(v + t, u)$ avec $t \geq 0$. Soit $t \geq 0$, par croissance de P, $P(\omega_t) \leq P(u) = u$ et $P(\omega_t) \leq P(v + t)$. De plus, comme P est contractante $\|P(v + t) - P(v)\| \leq \|v + t - v\| = t$, et par définition de la norme infinie, $P(v+t) \leq P(v)+t = v+t$ et on conclut que $P(\omega_t) \leq \omega_t$.

Soit maintenant, $t_0 = \inf\{t > 0 \mid \omega_t = u\}$. Pour tout $0 < t < t_0$, $P(\omega_t) \leq w_t < w_{t_0} = u$ et par le lemme 8.4, $P(\omega_t) \in \mathrm{fix}(f)$. La fonction P étant continue et ω_t dépendant continuement de t, $P(\omega_t)$ tend vers u quand t tend vers t_0 ce qui contredit la locale minimalité de u.

Finalement, u localement minimal implique u est le plus petit point fixe. La réciproque est immédiate. ∎

Exemple 8.28
Soit $\alpha \geq 1/4$. Considérons la fonction f définie par :

$$f(x, y) = \begin{pmatrix} \sqrt{|y| + \alpha} \\ \max(x, y) \end{pmatrix}$$

La fonction f est contractante puisque $\|\sqrt{|y| + \alpha} - \sqrt{|y'| + \alpha}\| \leq (2\sqrt{\alpha})^{-1}\|(x, y) - (x', y')\|$. L'ensemble des points fixes est $\{(x, y) \in \mathbb{R}^2 \mid x \geq (1 + \sqrt{1 + 4\alpha})/2,\ y = x^2 - \alpha\}$.

Soit $(x, y) \in \mathrm{fix}(f)$, tel que $x = y = (1 + \sqrt{1 + 4\alpha})/2 = \psi(\alpha)$. On pose, de plus, que $\gamma(\alpha) = 2\sqrt{\psi(\alpha) + \alpha}$ on en déduit que la semidifférentielle de f en $(\psi(\alpha), \psi(\alpha))$ est la fonction $h \mapsto (h_2(\gamma(\alpha))^{-1}, \max(h_1, h_2))$. On note cette fonction $f'_{\psi(\alpha)}$. Pour montrer que ce point

fixe est le plus petit, il suffit de montrer qu'il est localement minimal et on peut utiliser le théorème 8.3. Soit $\lambda \geq 0$ et $h = (h_1, h_2) \neq 0$ tels que $h \in \mathbb{R}^2_-$ et $f'_{\psi(\alpha)}(h) = \lambda h$ ce qu'on peut traduire par :

$$\begin{cases} h_2 = \lambda\gamma(\alpha)h_1 \\ \lambda h_2 = \max(h_1, h_2) \end{cases} \text{ ou encore } \begin{cases} h_2 = \lambda\gamma(\alpha)h_1 \\ \lambda^2\gamma(\alpha)h_1 = \max(h_1, \lambda\gamma(\alpha)h_1) \end{cases}$$

On en déduit que $\lambda^2\gamma(\alpha)h_1 \geq h_1$ et que $\lambda^2\gamma(\alpha)h_1 \geq \lambda\gamma(\alpha)h_1$. La première inégalité implique $\lambda^2\gamma(\alpha) \leq 1$. Par ailleurs, on remarque que si $h_1 = 0$ alors h est le vecteur nul donc $h_1 < 0$. Ceci entraîne que $\lambda \leq 1$ (que l'on peut aussi déduire des propriétés de contraction de f). On conclut que $0 \leq \lambda \leq \inf(\sqrt{\gamma(\alpha)^{-1}}, 1) = \sqrt{\gamma(\alpha)^{-1}} < 1$. Le rayon spectral de $f'_{\psi(\alpha)}$ sur $\mathbb{R}^2_-$ est donc strictement plus petit que 1, en conclusion $(\psi(\alpha), \psi(\alpha))$ est bien le plus petit point fixe de f.

En conclusion, en concaténant le théorème 8.3, le corollaire 8.1 et la proposition 8.7, si f est affine par morceaux et contractante au sens large, on obtient une condition nécéssaire et suffisante de minimalité de point fixe fini :

Théorème 8.5 (*Caractérisation du plus petit point fixe*)

Soit $f : \mathbb{R}^d \to \mathbb{R}^d$, croissante, affine par morceaux et contractante. Soit $u \in \text{fix}(f)$. Les assertions suivantes sont équivalentes :

(1) u est le plus petit point fixe ;
(2) f'_u n'a pas de point fixe non nul dans $\mathbb{R}^d_-$;
(3) $\rho_{\mathbb{R}^d_-}(f'_u) < 1$.

Démonstration

Puisque f est contractante au sens large d'après le théorème 8.4, le plus petit point fixe u de f est l'unique point fixe localement minimal, l'assertion (1) est donc équivalent à la proposition "u est localement minimal". De plus, on a montré dans le théorème 8.3 que (3) $\implies$ (2) $\implies$ (1).

Il suffit donc de prouver les implications inverses. D'après le corollaire 8.1, comme f est affine par morceaux alors (1) $\implies$ (2). Enfin, d'après la proposition 8.7, (2) $\implies$ (3) ce qui achève la preuve du théorème. ∎

Exemple 8.29

On considère la fonction f définie par :

$$f(x, y) = \begin{pmatrix} \min(\max(x, 0), \max(y, 0)) \\ \max(\max(x, 0), \max(y, 0)) \end{pmatrix}$$

On a déjà vu que l'ensemble des points de f était égal à $\{(x, y) \in \mathbb{R}^2 \mid y \geq x \geq 0\}$. Soit un point fixe $(u, v) \neq (0, 0)$.

Supposons que $v > 0$ et $u = 0$. D'après la proposition 8.5, la semidifférentielle en (u, v) est la fonction $h \mapsto (\max(h_1, 0), h_2)$ et cette fonction admet un point fixe négatif non nul (par exemple $(0, -1)$) et donc (u, v) n'est pas le plus petit point fixe de f.

Maintenant, si u et v sont strictement positifs et vérifient $v \geq u$. D'après la proposition 8.5, la semidifférentielle de f en (u, v) est la fonction $h \mapsto (h_1, h_2)$ si $v > u$ et la fonction $h \mapsto (\min(h_1, h_2), \max(h_1, h_2))$ et dans les deux cas, le vecteur $(-1, -1)$ est un point fixe négatif

non-nul de la semidifférentielle et donc dans les deux cas (u, v) n'est pas le plus petit point fixe de f.

Finalement, en prenant $(u, v) = (0, 0)$, la semidifférentielle en $(0, 0)$ est la fonction $h \mapsto (\min(\max(h_1, 0), \max(h_2, 0)), \max(\max(h_1, 0), \max(h_2, 0)))$ et sur $\mathbb{R}^2_-$ la semidifférentielle en $(0, 0)$ est la fonction nulle et donc n'admet pas de point fixe négatif non nul et donc $(0, 0)$ est le plus petit point fixe de f.

Exemple 8.30
Soit la fonction f définie par :

$$f(x, y) = \begin{pmatrix} \frac{1}{2}\min(x, y) + \frac{1}{3}\max(x, y) + 5 \\ \max(\min(\frac{1}{4}x + \frac{3}{4}y, y), 0) \end{pmatrix}$$

La fonction est clairement croissante et est affine par morceaux car la première coordonnée est égale à la fonction $(x, y) \mapsto \min(\max(\frac{5}{6}x, \frac{1}{2}x + \frac{1}{3}y), \max(\frac{1}{3}x + \frac{1}{2}y, \frac{5}{6}y)) + 5$. La fonction est également contractante en tant que min-max d'applications contractantes.

L'ensemble $\text{fix}(f)$ est $\{(x, y) \in \mathbb{R}^2 \mid 4x = 3y + 30,\ 30 \geq y \geq 0\}$. Le point $(15/2, 0)$ est clairement le plus petit point fixe de f. Calculons les semidifférentielles en tout point fixe de f. Au point $(15/2, 0)$, la semidifférentielle de f est la fonction :

$$(h_1, h_2) \mapsto f'_{(15/2,0)}(h) = \begin{pmatrix} \frac{1}{3}h_1 + \frac{1}{2}h_2 \\ \max(h_2, 0) \end{pmatrix}$$

On conclut que $h = f'_{(0,0)}(h)$ implique que $h_2 = 0$ puis que $h_1 = 0$. Finalement le seul point fixe négatif de $f'_{(0,0)}$ est 0 ce qui implique que $(0, 0)$ est le plus petit point fixe.

Soit $(x, y) \in \text{fix}(f)$ tel que $30 \geq y > 0$. Au point (x, y), la semidifférentielle de f est la fonction :

$$(h_1, h_2) \mapsto f'_{(x,y)}(h) = \begin{pmatrix} \frac{1}{3}h_1 + \frac{1}{2}h_2 \\ h_2 \end{pmatrix}$$

Le point $(-1, -4/3)$ est un point fixe négatif de $f'_{(x,y)}$ ce qui montre bien que (x, y) n'est pas le plus petit point fixe de f.

Contre-exemple 8.3
On reprend la fonction f de l'exemple 8.19. On ne peut pas utiliser le théorème 8.5 pour caractériser le plus petit point fixe de f : la fonction possède plusieurs points fixes localement minimaux. On remarque aussi que l'ensemble des points fixes de f n'est pas connexe par arcs car la fonction n'est pas contractante.

De manière duale, tout le travail développé dans ce chapitre peut être appliqué pour calculer le plus grand point fixe. On pourrait donc appliquer ces travaux pour prouver qu'un point fixe est unique en montrant qu'il est à la fois le plus petit point fixe et le plus grand point fixe.

8.2.4 Raffinement de l'algorithme d'itération sur les politiques

Le théorème 8.5 permet de construire une étape supplémentaire dans l'algorithme d'itération sur les politiques dans le cas affine par morceaux contractant et croissant.

Soit une fonction f affine par morceaux, contractante et croissante, elle est donc de la forme :

$$\min_{a\in A(i)} \max_{b\in B(i,a)} P_i^{ab}\cdot x + R_i^{ab} \tag{$\diamond$}$$

où, pour tout $i \in \{1,\dots,d\}$, $A(i)$ et $B(i,a)$ sont des ensembles finis, pour tout $a \in A(i)$ et $b \in B(i,a)$, pour tout $j \in \{1,\dots,d\}$, $P_{ij}^{ab} \geq 0$ et $\sum_{j=1}^{d} P_{ij}^{ab} \leq 1$. Pour $a \in A(i)$ et $b \in B(i,a)$, les éléments R_i^{ab} sont des nombres réels.

On note :

$$\Pi = \{\pi : \{1,\dots,d\} \mapsto \bigcup_{i=1,\dots d} A(i) \mid \pi(i) = \pi_i \in A(i)\}$$

d'où la formulation :

$$f = \min_{\pi\in\Pi} f^\pi \tag{$\diamond\diamond$}$$

où l'application $f^\pi : \mathbb{R}^d \to \mathbb{R}^d$ est telle que pour tout $x \in \mathbb{R}^d$ et pour tout $i \in \{1,\dots,d\}$,

$$f_i^{\pi_i}(x) = \max_{b\in B(i,\pi_i)} P_i^{\pi_i b}\cdot x + R_i^{\pi_i b} \tag{8.4}$$

Les applications f^π sont des applications convexes car toutes leurs fonctions coordonnées sont des maxima d'applications affines. De plus, elles sont croissantes en tant que maxima de fonctions affines dont les vecteurs directeurs sont des vecteurs positifs. En outre, en vertu de (8.1), le plus petit point fixe de f est fini si et seulement si aucun plus petit point fixe d'une application f^π n'a de coordonnée $-\infty$ et au moins l'un de ces points fixes appartient à $\mathbb{R}^d$.

Remarque 8.7 Le nombre de politiques est fini, donné par $\prod_i^d \mathrm{card}(A(i))$.

Pour prouver la convergence de l'algorithme 8, nous devons faire une hypothèse sur les plus petits points fixe des applications f^π. La prémière nous permettra d'utiliser le théorème 8.1 :

$$\forall\, \pi \in \Pi \;\; \mathrm{lfp}(f^\pi) \in \mathbb{R}^d \tag{H1}$$

La seconde hypothèse permet de considérer les applications f et f^π uniquement en tant qu'applications dans $\mathbb{R}^d$. On suppose que toute les politiques f^π possède un plus petit point fixe dans $\mathbb{R}^d$ au sens où pour tout $\pi \in \Pi$, il existe un vecteur $x^\pi \in \mathbb{R}^d$ tel que $x^\pi \leq x$ pour tout point fixe $x \in \mathbb{R}^d$ de f^π. On note ainsi l'hypothèse :

$$\forall\, \pi \in \Pi,\; \exists\, x^\pi \in \mathbb{R}^d \;(f^\pi(x) = x, x \in \mathbb{R}^d) \implies x^\pi \leq x \tag{H2}$$

Algorithme 8: Calcul du plus petit point fixe par itération sur les politiques

Entrées : Soit f une fonction de la forme ($\diamond$).
Sorties : Le plus petit point fixe de f dans $\mathbb{R}^d$.
début
  Choisir une politique π^0, $k = 0$;
  Détermination de la valeur ($\mathbf{D_k}$) :
  Calculer le plus petit point fixe u^k de f^{π^k};
  Amélioration de la politique ($\mathbf{I_k}$) :
  si $f(u^k) < u^k$ **alors**
    Choisir π^{k+1} telle que $f(u^k) = f^{\pi^{k+1}}(u^k)$;
    Aller à l'étape ($\mathbf{D_{k+1}}$);
  sinon// $f(u^k) = u^k$
    Calculer $\alpha_k := \rho(f'_{u^k})$;
    si $\alpha_k < 1$ **alors**
      retourner u^k, le plus petit point fixe de f;
    sinon// $\alpha_k = 1$
      Prendre $h \in \mathbb{R}^d_- \backslash \{0\}$ tel que $f'_{u^k}(h) = h$;
      pour chaque $j \in \{1, \ldots, d\}$ **faire**
        Choisir $\pi^{k+1}(j)$, l'action a qui atteint le minimum dans $(f'_{u^k})_j(h) = \min_{a \in \overline{A_j}} (f'_a)_{u^k}(h)$ où $\overline{A_j} = \{a \in A(j) \mid f_a(u^k) = f(u^k)\}$;
      Aller à l'étape ($\mathbf{D_{k+1}}$).

Théorème 8.6 (*Convergence en temps fini de l'algorithme*)

Si (H1) ou (H2) sont satisfaites, les propriétés suivantes sont vraies :

1. La suite $(u_k)_{k \geq 0}$ générée par l'algorithme 8 est strictement décroissante.
2. L'algorithme retourne le plus petit point fixe de f en temps fini.

Le lemme suivant est un analague du théorème 8.1 dans le cas des applications sur $\mathbb{R}^d$.

Lemme 8.5

Soit $g : \mathbb{R}^d \to \mathbb{R}^d$ croissante et contractante au sens large. Supposons que g possède un plus petit point fixe u dans $\mathbb{R}^d$. Alors les assertions suivantes sont vraies :

1. u est le plus petit élément de l'ensemble $\{x \in \mathbb{R}^d \mid f(x) \leqslant x\}$;
2. u est l'unique solution du problème d'optimisation :

$$\text{Min} \left\{ \sum_i^d x_i \mid g(x) \leq x,\ x \in \mathbb{R}^d \right\} \tag{P}$$

Démonstration (Lemme 8.5)
Le vecteur u est un point réalisable de (P). Si x est un autre point réalisable, comme g est croissante et contractante au sens large et possède un point fixe, alors $(g^k(x))_k$ est une suite décroissante et bornée. Appelons y la limite de $g^k(x)$, nous obtenons $y \leq x$ et y est

un point fixe fini de g d'où $y \geq u$ ce qui montre la première assertion. On en déduit que $\sum_{1\leq i\leq d} u_i \leq \sum_{1\leq i\leq d} y_i \leq \sum_{1\leq i\leq d} x_i$. Comme x est un point réalisable arbitraire, il s'ensuit que u est une solution optimale. Si x est une solution optimale, on doit avoir $\sum_{1\leq i\leq d} u_i = \sum_{1\leq i\leq d} x_i$ et puisque $u \leqslant x$, on conclut que $u = x$ ce qui montre la seconde assertion. ∎

Démonstration (Théorème 8.6)

1. Le résultat est établi dans Gaubert, Goubault et al dans [GGTZ07, Theorem 3] dans le cadre de l'hypothèse (H1). Nous en redonnons la preuve ici. Soit $k \geq 0$. Si $u^k \in \mathbb{R}^d$ n'est pas un point fixe de f alors il existe π^{k+1} tel que $f(u^k) = f^{\pi^{k+1}}(u^k)$. Soit $E = \{x \in \mathbb{R}^d \mid f^{\pi^{k+1}}(x) \leq x\}$. Sous (H1), $f^{\pi^{k+1}}$ possède un plus petit point fixe $u^{k+1} \in \mathbb{R}^d$ qui est d'après le théorème de Tarski et la croissance de $f^{\pi^{k+1}}$, le plus petit élément de E. Sous (H2), $f^{\pi^{k+1}}$ possède un plus petit point fixe dans $\mathbb{R}^d$, u^{k+1} qui est d'après la première assertion du lemme 8.5, le plus petit élément de l'ensemble E. Or, comme $u^k \in E$ et donc $u^{k+1} \leq u^k$. De plus, par croissance de $f^{\pi^{k+1}}$, on a $f^{\pi^{k+1}}(u^{k+1}) = u^{k+1} \leq f^{\pi^{k+1}}(u^k) = f(u_k) < u_k$. Maintenant si u_k est un point fixe mais pas le plus petit alors il existe $h \in \mathbb{R}^d_-$ non nul tel que $u^{k+1} \leq f(u^k + h) = u^k + h < u^k$.
2. Deux politiques ne peuvent pas être sélectionnées deux fois puisque, d'après le point 1, la suite générée par l'algorithme est strictement décroissante et comme le nombre de politiques est fini alors l'algorithme retourne le plus petit point fixe en au plus card(Π) étapes. ∎

Nous avons déjà montré au théorème 8.1 comment calculer le plus petit point fixe d'une fonction associée à une politique lorsque l'hypothèse (H1) est vérifiée. En effet, une fonction associée à une politique est une fonction convexe en tant que supremum de fonctions convexes. La deuxième assertion du lemme 8.5 nous dit aussi que sous l'hypothèse (H2), le plus petit point fixe d'une fonction associée à une politique se calcule par optimisation convexe. nous résumons ce résultat sous la forme d'un corollaire.

Corollaire 8.2 (*Calcul du plus petit point fixe d'une politique*)

Si (H1) ou (H2) sont satisfaites alors pour tout $\pi \in \Pi$, $\mathrm{lfp}(f^\pi)$ est l'unique solution du programme linéaire :

$$\mathrm{Min}\left\{\sum_i^d x_i \mid f^\pi(x) \leq x,\ x \in \mathbb{R}^d\right\} \qquad (P_\pi)$$

Il nous reste à donner des méthodes pour calculer le rayon spectral d'une application homogène et continue.

8.3 Aspects numériques et pratiques de la minimalité

Le but de cette section est d'expliciter des méthodes pour calculer ou évaluer le rayon spectral. Dans le théorème 8.5, nous avons caractérisé les points fixes grâce au rayon spectral de la semidifférentielle aux points fixes. Il faut donc trouver une méthode pour calculer ou évaluer le rayon spectral et trouver un certificat pour prouver que le rayon spectral est strictement plus petit que 1 ou qu'il est strictement plus grand que 1. On verra que le cas problématique

est le cas où le rayon spectral vaut 1. La sous-section 8.3.1 traite de l'encradrement du rayon spectral, cet encadrement se base sur résultat de Nussbaum [Nus86] puis dans la section 8.3.2, nous présentons le cas particulier des fonctions min-max homogènes.

8.3.1 Encadrement du rayon spectral

Méthode des puissances et borne supérieure du rayon spectral

En dimension finie, pour un cône C convexe fermé pointé non réduit à $\{0\}$ et d'intérieur non vide et une fonction continue, croissante et homogène g, on va pouvoir encadrer le rayon spectral de g. En dimension finie, il existe d'autres définitions équivalentes du rayon spectral. Nous commençons pare la définition du rayon spectral relatif à un cône [1].

Définition 8.9 (*Rayon spectral relatif à un cône*)

Soit C un cône convexe fermé pointé. Soit une fonction continue g homogène de C dans C. On appelle rayon spectral relatif au cône C

$$\hat{\rho}_C(g) = \sup_{x\in C} \limsup_{k\to\infty} \|g^k(x)\|_\infty^{1/k}$$

Une autre définition équivalente est le rayon spectral de Bonsall.

Définition 8.10 (*Rayon spectral de Bonsall*)

Soit C un cône convexe fermé pointé. Soit une fonction continue g homogène de C dans C. Le rayon spectral de Bonsall est défini

$$\begin{aligned}\tilde{\rho}_C(g) &= \lim_{n\to+\infty} \sup_{\substack{\|x\|\leq 1\\ x\in C}} \|g^n(x)\| \\ &= \inf_{n\geq 1} \sup_{\substack{\|x\|\leq 1\\ x\in C}} \|g^n(x)\| \ .\end{aligned}$$

Mallet-Paret et Nussbaum ont montré dans [MPN02] qu'en dimension finie, l'égalité (♣) est satisfaite :

$$\rho_C(g) = \hat{\rho}_C(g) = \tilde{\rho}_C(g) \qquad (\clubsuit)$$

Remarque 8.8 Mallet-Paret et Nussbaum, dans [MPN02], ont montré que l'égalité (♣) demeurait vraie en dimension infinie lorsque le rayon spectral est plus grand qu'une mesure de compacité.

De plus, Nussbaum a montré dans [Nus86], qu'en prenant $C = \mathbb{R}^d_+$, on a, de plus :

$$\rho_C(g) = \inf_{x\in \mathrm{int}(C)} \max_{1\leq i\leq d} \frac{g_i(x)}{x_i} \qquad (8.5)$$

$$= \sup_{\substack{x\in\mathbb{R}^d_+\\ x\neq 0}} \min_{\substack{1\leq i\leq d\\ x_i\neq 0}} \frac{g_i(x)}{x_i} \qquad (8.6)$$

1. Nous traduisons ainsi le terme "cone spectral radius" [MPN02].

Le résultat reste valide en prenant $C = \mathbb{R}^d_-$ et $g : C \mapsto C$. Nous supposerons maintenant que $C = \mathbb{R}^d_-$. En utilisant la définition du rayon spectral de Bonsall (définition 8.10), on obtient que $\rho_C(g^p) = (\rho_C(g))^p$ pour tout entier $p \geq 1$.

Soit $y \in \text{int}(C)$. On déduit, en utilisant le résultat de Nussbaum de l'équation (8.5), pour tout $p \geq 1$ entier :

$$\rho_C(g) \leq \max_{1\leq i\leq d} \left(\frac{g_i^p(y)}{y_i} \right)^{\frac{1}{p}}$$

Pour tout $i \in \{1, \ldots, d\}$, y_i est strictement négatif d'où $\min_{j=1,\ldots,d}(y_j) \leq y_i$, ce qui équivaut à $(\min_{i=1,\ldots,d} y_j) y_j^{-1} \geq 1$.

De plus, nous déduisons de $g_i^p(y) \leq 0$ que, pour tout $i \in \{1, \ldots, d\}$:

$$(\min_{j=1,\ldots,d} y_j) y_i^{-1} g_i^p(y) \geq g_i^p(y)$$

d'où :

$$y_i^{-1} g_i^p(y) \leq -(\min_{j=1,\ldots,d} y_j)^{-1} (-g_i^p(y))$$

en prenant le max sur i, on conclut que :

$$\max_{i=1,\ldots,d} y_i^{-1} g_i^p(y) \leq (\max_{j=1,\ldots,d} -y_j)^{-1} \max_{i=1,\ldots,d} (-g_i^p(y)) \ .$$

En posant $M = \max_{j=1,\ldots,d} -y_j > 0$, on obtient que :

$$\rho_C(g) \leq (\max_{1\leq i\leq d} g_i^p(y) y_i^{-1})^{1/p} \leq M^{-1/p} \|g^p(y)\|^{1/p} \tag{8.7}$$

Or la lim sup quand $p \to \infty$ de $M^{-1/p}\|g^p(y)\|^{1/p}$ est $\limsup_{p\to+\infty} \|g^p(y)\|^{1/p}$ puis, en prenant le supremum sur $\text{int}(C)$ on obtient, pour tout $y \in \text{int}(C)$:

$$\begin{aligned} \rho_C(g) &\leq \liminf_{p\to+\infty} (\max_{1\leq i\leq d} g_i^p(y) y_i^{-1})^{1/p} \\ &\leq \limsup_{p\to+\infty} (\max_{1\leq i\leq d} g_i^p(y) y_i^{-1})^{1/p} \\ &\leq \sup_{z\in\text{int}(C)} \limsup_{p\to+\infty} \|g^p(z)\|^{1/p} \quad \leq \quad \hat{\rho}_C(g) = \rho_C(g) \ . \end{aligned}$$

En conclusion, $(\max_{1\leq i\leq d} g_i^p(y) y_i^{-1})^{1/p}$ converge vers $\rho_C(g)$ quand $p \to \infty$. Cette majoration donne une méthode pour prouver que $\rho_C(g) < 1$.

Algorithme 9: Méthode de la puissance pour le calcul du rayon spectral

Entrées : Une fonction homogène croissante contractante au sens large g et un vecteur $y \in \mathbb{R}^d_-$

Sorties : Une preuve que $\rho_{\mathbb{R}^d_-}(g) < 1$

$\rho^1 := \left(\max_{1\leq i\leq d} \frac{g_i(y)}{y_i} \right)$;

$k = 1$;

tant que $\rho^k \geq 1$ **faire**

  $\rho^k := \left(\max_{1\leq i\leq d} \frac{g_i^k(y)}{y_i} \right)$

L'algorithme 9 peut ne pas terminer si $\rho_C(g) = 1$.

Borne inférieure par la formule de Collatz-Wielandt

On peut également utiliser le résultat de Nussbaum (8.6) pour calculer la borne inférieure du rayon spectral. Prenons $x \neq 0$, et posons $\alpha(p,x) = \min\limits_{\substack{1\leq i\leq d\\ x_i\neq 0}} g_i^p(x)x_i^{-1}$, on obtient par conséquent :

$$\rho_C(g) \geq \alpha(p,x)^{1/p}$$

Cette minoration donne une borne valide du rayon spectral mais cette méthode d'approximation est difficilement implémentable, en effet, il faut choisir un « bon » rang p et un « bon » vecteur x non-nul pour avoir une borne intéressante c'est-à-dire x non-nul et $p \in \mathbb{N}^*$ tels que $\alpha(p,x)^{1/p} \geq 1$.

8.3.2 Fonctions min-max homogènes

Une application $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ est appelée application min-max homogène si toutes ses coordonnéees sont des fonctions min-max homogènes. La définition 8.11 est inspirée par Gunawardena [Gun94] et Olsder [Ols91].

Définition 8.11 (*Fonctions* min-max *homogènes*)

Une fonction $f : \mathbb{R}^d \mapsto \mathbb{R}$ est une fonction min-max homogènes des variables $h_1, \ldots, h_d$ si elle s'écrit : $X \mapsto \min(X, X), \max(X, X), h_1, \ldots, h_d, 0$.

Une application $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ sera dite min-max homogène si toutes les coordonnées f_i sont des fonctions min-max homogènes.

Exemple 8.31
La fonction $h \mapsto \min(h_1, \max(h_2, h_3, 0)$ est une fonction min-max homogène.

Exemple 8.32
L'application $f : \mathbb{R}^3 \mapsto \mathbb{R}^3$ définie par :

$$f(x,y,z) = \begin{pmatrix} \max(x, \max(z,0)) \\ \min(z,0) \\ \max(x,y,z) \end{pmatrix}$$

est min-max homogène.

Contre-exemple 8.4
La fonction qui à h associe $\min(h_1 + h_2, 0)$ n'est pas une application min-max homogène. En effet, elle n'est pas contractante et on verra dans la proposition 8.9 qu'une application min-max homogène est contractante.

Proposition 8.8 (*Stabilité des applications* min-max *homogènes sur* $\mathbb{Z}^d$)

Une application g min-max homogène vérifie $g(\mathbb{Z}^d) \subseteq \mathbb{Z}^d$.

Démonstration
Pour tout $i \in \{1, \ldots, d\}$, $h \mapsto h_i$ est à valeurs dans $\mathbb{Z}$ si h appartient à $\mathbb{Z}^d$, la fonction nulle est également à valeurs dans $\mathbb{Z}$. Soit une famille de fonctions $\{f_k\}_{k\in K}$ telle que $f_k : \mathbb{Z}^d \mapsto \mathbb{Z}$ où K est fini et non-vide vérifie $\max\limits_{k\in K} f_k : \mathbb{Z}^d \mapsto \mathbb{Z}$, ceci est vrai aussi pour un min. ∎

Proposition 8.9 (*Contraction d'une application* min-max *homogène*)

Une application min-max homogène est contractante au sens large.

Démonstration

Pour tout $i \in \{1, \dots, d\}$, les applications $h \mapsto h_i$ sont contractantes au sens larges puisque $|h_i - h'_i| \leq \sup_{i=1,\dots d} |h_i - h'_i| = \|h - h'\|$. La fonction nulle est également contractante. Maintenant, comme un min-max de fonctions contractantes est contractante alors une fonction min-max homogène est contractante. ∎

Proposition 8.10 (*Rayon spectral d'une application* min-max *homogène*)

Soit g une application min-max homogène sur $\mathbb{R}^d$. Soit e l'élément de $\mathbb{R}^d_-$ tel que $e_i = -1$, pour tout $i = 1, \dots, d$.

Les assertions suivantes sont satisfaites :

1. $\rho_{\mathbb{R}^d_-}(f'_u)(g) \in \{0, 1\}$.
2. $\rho_{\mathbb{R}^d_-}(f'_u)(g) = 0 \iff \lim_{k \to +\infty} g^k(e) = 0$.
3. La limite $\lim_{k \to +\infty} g^k(e)$ est atteinte en au plus d itérations.

Démonstration

Puisque g est une fonction min-max homogène, elle est croissante, contractante et homogène et donc $g(0) = 0$. On a par conséquent $\|g(e)\| \leq \|e\| = 1$ et comme $g(e) \leq 0$ d'où $g(e) \geq e$. En utilisant la monotonie de g, on déduit que $(g^k(e))_{k \geq 0}$ est une suite croissante et majorée par le vecteur nul, elle est donc convergente vers $b \geq 0$. Comme g préserve les entiers alors g converge vers $b \in \mathbb{Z}^d$ en au plus d pas ceci démontre le point 3.

Supposons que $b = 0$. Pour tout $y \in \mathbb{R}^d_-$, il existe $t \geq 0$ tel que $0 \geq y \geq te$ et comme g est monotone et homogène alors $0 \geq g^k(y) \geq tg^k(e) = 0$ pour tout $k \geq d$ ce qui implique que $\rho_{\mathbb{R}^d_-}(f'_u) = 0$. Si $b < 0$, nous avons $g(b) = b$ donc $\rho_{\mathbb{R}^d_-}(f'_u) \geq 0$ et g est contractante comme $\rho_{\mathbb{R}^d_-}(f'_u) \leq 1$. Ceci prouve les points 1 et 2. ∎

D'autres cas particuliers permettent de résoudre facilement les problèmes de points fixes lorsque la fonction est monotone, affine par morceaux, contractante au sens large. Si la fonction coïncide, sur le cône négatif, avec une application linéaire alors le problème devient de singularité de matrices. Le problème devient particulièrement facile lorsque la matrice associée à l'application linéaire est stochastique puisqu'elle admet dans ce cas, comme point fixe négatif le vecteur qui a pour coordonnée -1 partout.

CHAPITRE 9

CALCUL DE PLUS PETIT POINT FIXE : APPLICATIONS À L'ANALYSE STATIQUE ET AUX JEUX STOCHASTIQUES

Dans ce chapitre, nous présentons quelques applications des théorèmes de minimalité développés dans le chapitre 8. Nous nous intéressons en particulier aux jeux stochastiques dont la valeur est à la fois la limite des itérées de l'opérateur de Shapley et le plus petit point fixe de cet opérateur. De plus, pour travailler avec des applications affines par morceaux, nous considérons des opérateurs définis à partir des actions pures. Dans la première section 9.1, nous étudions des jeux stochastiques particuliers où les actions optimales sont mixtes et chargent toutes les actions, puis nous nous intéressons à des jeux stochastiques positifs particuliers : dans un premier temps, nous regardons des jeux où les joueurs n'ont que deux choix à chaque état, ils peuvent soit continuer à jouer soit arrêter le jeu ; ensuite, nous regardons des jeux stochastiques à information parfaite avec options d'arrêt et paiements positifs. Dans la deuxième section 9.2, nous présentons des exemples de programmes où la fonction sémantique abstraite est une application affine par morceaux contractante. On peut voir ces fonctions sémantiques abstraites comme des opérateurs dynamiques de jeux déterministes. Enfin, nous terminons à la section 9.3 sur quelques mots sur l'implémentation ce chapitre sur quelques mots sur le prototype écrit en C implémentant la méthode dans le cas de l'interprétation abstraite.

9.1 Applications aux jeux

La section débute par des rappels sur les jeux stochastiques à la sous-section 9.1.1, puis nous terminons cette section par dans la sous-section 9.1.2 sur le cas des jeux à paiements positifs.

9.1.1 Rappels sur les jeux stochastiques

Nous rappelons le cadre des jeux stochastiques du chapitre 3. Nous considérons des jeux stochastiques à deux joueurs à somme nulle, le joueur $J1$ est le minimiseur et le joueur $J2$ le maximiseur. Nous nous intéressons aux jeux stochastiques en temps discret. Le jeu est à espace d'états fini et nous notons $S = \{1, \dots, d\}$ l'espace d'états. Pour chaque état $s \in S$,

les joueurs $J1$ et $J2$ ont des espaces d'actions finis respectivement $A(s)$ et $B(s)$. Pour chaque état $s \in S$, un couple d'actions $(a, b) \in A(s) \times B(s)$ induit une probabilité de transition $P_s(a, b)$: un nouvel état t est tiré suivant la probabibilité $P_{st}(a, b)$, en outre, le joueur $J1$ paie instantanément $R_s(a, b)$ au joueur $J2$.

Le but des deux joueurs est d'optimiser un paiement final qui peut être l'espérance de la somme actualisée des paiements instantanées, la moyenne des paiements instantanés, la limite supérieure des paiements instantanés... Lorsque le jeu est en horizon fini le paiement final se calcule grâce à une équation récursive. En horizon infini, dans le cas escompté, la valeur du jeu est l'unqiue point fixe d'un opérateur qui représente le gain quotidien de chaque joueur. L'opérateur de Shapley représente cette optimisation journalière : l'opérateur associe à un vecteur $v \in \mathbb{R}^d$, la valeur du jeu en un coup déterminé par v. En vertu du théorème de Von-Neumann, à chaque date, la valeur du jeu en un coup existe quand on autorise les joueurs à choisir des actions mixtes et on obtient l'opérateur de Shapley :

$$x \mapsto F_s(x) = \inf_{\sigma \in \Delta(A(s))} \sup_{\tau \in \Delta(B(s))} \tilde{R}_s(\sigma, \tau) + \tilde{P}_s(\sigma, \tau) \cdot x, \quad \text{pour tout } s \in S$$

où $\Delta(A(s))$, $\Delta(B(s))$ désignent respectivement les ensembles de mesures de probabilités sur les espaces d'actions $A(s)$ et $B(s)$ des joueurs 1 et joueur 2 et $\tilde{R}_s$ et $\tilde{P}_s$ désignent les extensions mixtes.

Originellement, Shapley a considéré des modèles avec taux d'escompte ($\sum_{t \in S} P_{st}(a, b) < 1$ pour tout $s \in S$) et ainsi la limite de la suite $(F^n(x_0))_{n \geq 0}$ existe quelque soit $x_0 \in \mathbb{R}^d$ et converge vers l'unique point fixe de F. L'unicité et l'existence sont assurées par le fait que F est, pour des jeux à taux d'escompte, une fonction strictement contractante et la conclusion découle du théorème de point fixe de Picard.

Dans notre modèle, on considère des jeux sans taux d'escompte et donc la limite de la suite $(F^n(x_0))_{n \geq 0}$ n'existe pas forcément. Cependant, par croissance de F, cette limite existe et éventuellement à valeur infinie dès lors que x_0 satisfait $x_0 \leq F(x_0)$ ou $F(x_0) \leq x_0$. Par ailleurs, on sait par continuité de la fonction que la limite est un point fixe de F.

Remarque 9.1 On aurait pu considérer des jeux dont la valeur est donnée par $\limsup_{n \to +\infty} F^n(x_0)$ ou $\liminf_{n \to +\infty} F^n(x_0)$.

9.1.2 Les jeux à paiements positifs

Ce type de jeu a été étudié dans [FV97]. Ces jeux font partie de la classe des jeux à paiement total : on s'intéresse à la somme des paiements gagnés. Comme indique leur nom, les jeux à paiements positifs vérifient pour tout état s, pour tout couple d'actions (a, b), $R_s(a, b) \geq 0$. Les paiements finaux correspondent à l'espérance de la somme des paiements instantanés : en prenant un couple de stratégie (voir définition au chapitre 3) $(\sigma, \tau) = ((\pi^0, \ldots, \pi^k, \ldots), (\pi'^0, \ldots, \pi'^k, \ldots))$, on s'intéresse à la fonction :

$$w_s(\sigma, \tau) = E_s^{\sigma, \tau} \left(\sum_{k=0}^{\infty} R_{s^k}(\pi^k, \pi'^k) \right) \quad ,$$

nous rappelons que ce jeu a une valeur si $v(s) := \inf_\sigma \sup_\tau w_s(\sigma, \tau) = \sup_\tau \inf_\sigma w_s(\sigma, \tau)$.

Dans [FV97, Theorem 4.4.4], les auteurs montrent que la valeur du jeu positif existe et quand cette valeur est finie (en tout état), la valeur est la limite v de la suite $(v_n)_{n\in\mathbb{N}}$ définie récursivement, pour $s \in S$, pour $n \in \mathbb{N}$ par :

$$v_0(s) = \begin{cases} \min\limits_{a\in\Delta(A(s))} \max\limits_{b\in\Delta(B(s))} ([R_s(a,b)]_{a,b}) \\ \max\limits_{b\in\Delta(B(s))} \min\limits_{a\in\Delta(A(s))} ([R_s(a,b)]_{a,b}) \end{cases}$$

$$v_{n+1}(s) = \begin{cases} \min\limits_{a\in\Delta(A(s))} \max\limits_{b\in\Delta(B(s))} ([P_s(a,b)\cdot v_n + R_s(a,b)]_{a,b}) \\ \max\limits_{b\in\Delta(B(s))} \min\limits_{a\in\Delta(A(s))} ([P_s(a,b)\cdot v_n + R_s(a,b)]_{a,b}) \end{cases}$$

En outre, Filar et Vrieze [FV97, Theorem 4.4.3] démontrent que v est le plus petit point fixe positif (éventuellement à valeur $+\infty$) de l'opérateur de Shapley F :

$$x \mapsto F(x) = \begin{cases} \min\limits_{a\in\Delta(A(s))} \max\limits_{b\in\Delta(B(s))} ([P_s(a,b)\cdot x + R_s(a,b)]_{a,b}) \\ \max\limits_{b\in\Delta(B(s))} \min\limits_{a\in\Delta(A(s))} ([P_s(a,b)\cdot x + R_s(a,b)]_{a,b}) \end{cases}$$

Par ailleurs, les auteurs montrent que leur modèle est valable les jeux (sous-)stochastiques c'est-à-dire où le vecteur $P_s(a,b)$ est sous-stochastique. Nous supposons par conséquent désormais que pour tout $s \in S$, pour tout $(a,b) \in A(s)\times B(s)$, pour tout $t \in S$, $P_{st}(a,b) \geq 0$ et $\sum\limits_{t\in S} P_s(a,b) \leq 1$.

Stop ou encore

On définit un jeu de la manière suivante : à chaque état $s \in S$, à chaque date k, les joueurs peuvent arrêter de jouer ou continuer à jouer et un nouvel état t est tiré sous une probabilité P_{st}.

- Si le joueur $J1$ est le seul à décider de s'arrêter alors il paie une somme α_s au joueur $J2$;
- si le joueur $J2$ décide seul de s'arrêter alors, le joueur $J1$ doit lui payer une somme β_s ;
- si les deux joueurs décident de s'arrêter en même temps, le joueur $J1$ paie une somme γ_s au joueur $J2$.
- si les deux joueurs décident de continuer à jouer, le joueur $J1$ paie au joueur $J2$ une somme R_s.

On peut supposer que les paiements d'arrêts sont éventuellement à valeurs dans $\overline{\mathbb{R}}$.

On peut représenter le jeu à un état s sous la forme matricielle :

$G_s =$

$J2\backslash J1$	Continue	Arrête
Continue	$\sum\limits_{t\in S} P_{st}G_t + R_s$	α_s
Arrête	β_s	γ_s

La notation $\sum\limits_{t\in S} P_{st}G_t + R_s$ signifie qu'avec probabilité P_{st} les joueurs entrent dans le jeu d'arrêt G_t et que pour le couple d'actions $(continue, continue)$ $J1$ paie R_s à $J2$.

Remplaçons arbitrairement $(G_t)_{t\in S}$ par un vecteur $x = (x_t)_{t\in S}$ et notons $G_s(x)$ le nouveau jeu matriciel, ainsi :

$$G_s(x) = \begin{array}{|c|c|c|} \hline J2\backslash J1 & \text{Continue} & \text{Arrête} \\ \hline \text{Continue} & \sum_{t\in S} P_{st}x_t + R_s & \alpha_s \\ \hline \text{Arrête} & \beta_s & \gamma_s \\ \hline \end{array}$$

La proposition est un résultat bien connu de jeu matriciel 2×2, elle permet de caractériser les jeux matriciels 2×2 qui ne possèdent pas de valeur en actions pures.

Proposition 9.1 (*Valeur de $G_t(x)$*)

Soit $s \in S$ et soit $x \in \mathbb{R}^S$. Nous posons $h_s(x) = \sum_{t\in S} P_{st}x_t + R_s$. Le jeu matriciel $G_s(x)$ ne possède pas de valeur en action pure si et seulement, l'une des conditions est satisfaite :

1. $h_s(x) > \alpha_s, \beta_s$ et $\gamma_s > \alpha_s, \beta_s$;
2. $\alpha_s > h_s(x), \gamma_s$ et $\beta_s > h_s(x), \gamma_s$.

Si le jeu $G_s(x)$ possède une valeur $v_s(x)$ en action pure et si le jeu $G = (G_s)_{s\in S}$ possède une valeur $v = (v_s)_{s\in S}$ alors v est un point fixe de l'opérateur F défini par :

$$x \mapsto F_s(x) = \begin{cases} \max(\min(\sum_{t\in S} P_{st}(x_t + R_t), \beta_s), \min(\alpha_s, \gamma_s)) \\ \min(\max(\sum_{t\in S} P_{st}(x_t + R_t), \alpha_s), \max(\beta_s, \gamma_s)) \end{cases}$$

En supposant les paiements d'arrêts du joueur $J1$ positifs, c'est-à-dire, pour tout $s \in S$, $\alpha_s, \gamma_s \geq 0$, tout point fixe de F est positif, en effet, grâce à l'égalité min-max, x est un point fixe de F, implique que pour tout $s \in S$, $x_s = \max(\min(\sum_{t\in S} P_{st}(x_t + R_t), \beta_s), \min(\alpha_s, \gamma_s))$ et donc que $x_s \geq \min(\alpha_s, \gamma_s)) \geq 0$ pour tout $s \in S$.

Par ailleurs, en supposant α_s et γ_s finis pour tout $s \in S$, F est majorée par $M = (\max(\alpha_s, \gamma_s))_{s\in S}$ et donc $\text{fix}(F) \subseteq [0, M_1] \times [0, M_2] \times \ldots \times [0, M_d]$. Finalement, la suite $(F^k(0))_{k\geq 0}$ est croissante majorée et par continuité de F converge vers un point fixe fini de F d'où $\text{fix}(F) \neq \emptyset$.

En conclusion, d'après les résultats de Filar et Vrieze énoncés plus tôt sur les jeux positifs, la limite de $(F^k(0))_{k\geq 0}$ est la valeur du jeu et la valeur du jeu est le plus petit point fixe de F.

Exemple 9.1 (Application numérique à deux états)
Considérons le jeu suivant :

1. A l'état 1, la matrice de paiements est la suivante :

$$G_1 = \begin{array}{|c|c|} \hline 3/4G_1 + 1/4G_2 + 5 & 25 \\ \hline 50 & 30 \\ \hline \end{array}$$

2. A l'état 2, la matrice de paiements est la suivante :

$$G_2 = \begin{array}{|c|c|} \hline G_2 & 20 \\ \hline 40 & 30 \\ \hline \end{array}$$

Les jeux G_1 et G_2 admettent une valeur en action pure.

L'opérateur de Shapley F est la fonction :

$$x \mapsto F(x) = \begin{pmatrix} \min(\max(3/4x_1 + 1/4x_2 + 5, 25), 50) \\ \min(\max(x_2, 20), 40) \end{pmatrix}$$

En choisissant les paiements d'arrêts comme politique initiale, on trouve le point fixe $(u_1, u_2) = (50, 40)$. La valeur 40 est atteinte par la fonction les deux actions à l'état 2 et pour l'état 1, une seule action atteint la valeur 50, la semidifférentielle en (u_1, u_2) est la fonction : $(h_1, h_2) \mapsto (0, \min(h_2, 0))$, la semidifférentielle admet donc le point fixe négatif non nul $(0, -1)$ qui est donné par l'itération du vecteur $e = (-1, -1)$. Ce point fixe conduit à une nouvelle politique associée à la fonction f^{π^1} définie par $(x_1, x_2) \mapsto (50, \max(x_2, 20))$. Par programmation linéaire, $\text{lfp}(f^{\pi^1}) = (50, 20)$ qui n'est pas un point fixe de F car $0.75 \times 50 + 0.25 \times 20 + 5 = 47.5 < 50$ et donc, nous en déduisons une nouvelle politique π^2, associée à la fonction f^{π^2} définie par $(x_1, x_2) \mapsto (\max(3/4x_1 + 1/4x_2 + 5, 25), \max(x_2, 20))$ et dont $\text{lfp}(f^{\pi^2}) = (40, 20)$ qui est un point fixe de F que nous notons (v_1, v_2). La semidifférentielle g en (v_1, v_2) est $(h_1, h_2) \mapsto (3/4h_1 + 1/4h_2, \max(h_2, 0))$. En utilisant l'algorithme 9, on trouve, comme $(g^k)_1(e) = -((3/4)^k)$ et $(g^k)_2(e) = 0$, que $\rho(g) < 1$ et on conclut que $(40, 20)$ est le plus petit point fixe de F.

Exemple 9.2 (Exemple avec trois états et paiements instantanés nuls)
Ce type de jeu se rapproche du modèle des jeux récursifs d'Everett [Eve57] où, lorsque les joueurs continuent à jouer les paiements instantanés sont nuls, seuls les paiements d'arrêts sont non nuls. De plus, dans les modèles d'Everett, l'ensemble des points fixes de l'opérateur de programmation dynamique n'est pas nécessairement un singleton.

Considérons le jeu suivant :

1. A l'état 1, la matrice de paiements est la suivante :

$$G_1 = \begin{array}{|c|c|} \hline 1/2G_1 + 1/2G_2 & 50 \\ \hline 90 & 90 \\ \hline \end{array}$$

2. A l'état 2, la matrice de paiements est la suivante :

$$G_2 = \begin{array}{|c|c|} \hline 1/4G_1 + 1/2G_2 + 1/4G_3 & 60 \\ \hline 90 & 70 \\ \hline \end{array}$$

3. A l'état 3, la matrice de paiements est la suivante :

$$G_3 = \begin{array}{|c|c|} \hline 1/4G_1 + 3/4G_3 & 70 \\ \hline 100 & 75 \\ \hline \end{array}$$

Les jeux G_1, G_2 et G_3 admettent une valeur en action pure. Maintenant calulons le plus petit point fixe de la fonction :

$$x \mapsto F(x) = \begin{pmatrix} \min(\max(1/2x_1 + 1/2x_2, 50), 90) \\ \min(\max(1/4x_1 + 1/2x_2 + 1/4x_3, 60), 90) \\ \min(\max(1/4x_1 + 3/4x_3, 70), 100) \end{pmatrix}$$

Choisissons les paiements d'arrêts comme politique initiale f^{π^0}, nous obtenons le vecteur $x^0 = (90, 90, 100)$, or $F(x^0) = (90, 90, 97.5)$ et donc nous changeons de politique et nous considérons la fonction f^{π^1} définie par :

$$x \mapsto f^{\pi^1}(x) = \begin{pmatrix} 90 \\ 90 \\ \max(1/4x_1 + 3/4x_3, 70) \end{pmatrix}$$

Par programmation linéaire, nous trouvons le vecteur $x^1 = (90, 90, 90)$, et comme $F(x^1) = (90, 90, 90)$, x^1 est un point fixe de F. Par ailleurs, $1/2x_1^1 + 1/2x_2^1 = 90$ et $1/4x_1^1 + 1/2x_2^1 + 1/4x_3^1 = 90$, la semidifférentielle g au point x^1 est par conséquent :

$$h \mapsto g(h) = \begin{pmatrix} \min(1/2h_1 + 1/2h_2, 0) \\ \min(1/4h_1 + 1/2h_2 + 1/4h_3, 0) \\ 1/4h_1 + 3/4h_3 \end{pmatrix}$$

Sur l'orthant négatif, la fonction g coïncide avec l'application linéaire $x \mapsto Px$ où P est la matrice de transition, le rayon spectral vaut 1 puisque la matrice est stochastique et on conclut qu'il existe un point fixe négatif non-nul pour la semidifférentielle (par exemple e) et que le point fixe x^1 n'est pas le plus petit point fixe de F. On en déduit une nouvelle politique π^2 associée à la fonction f^{π^2} définie par :

$$x \mapsto f^{\pi^2}(x) = \begin{pmatrix} \max(1/2x_1 + 1/2x_2, 50) \\ \max(1/4x_1 + 1/2x_2 + 1/4x_3, 60) \\ \max(1/4x_1 + 3/4x_3, 70) \end{pmatrix}$$

Par programmation linéaire, on trouve $x^2 = (70, 70, 70)$. La semidifférentielle h en x^2 est la fonction :

$$x \mapsto h(x) = \begin{pmatrix} 1/2h_1 + 1/2h_2 \\ 1/4h_1 + 1/2h_2 + 1/4h_3 \\ \max(1/4h_1 + 3/4h_3, 0) \end{pmatrix}$$

Par un simple calcul de point fixe, on trouve que l'unique point fixe négatif de h est 0. En conclusion, $x^2 = (70, 70, 70)$ est le plus petit point fixe de F et donc la valeur du jeu.

Jeux stochastiques à information parfaite

Ces jeux sont aussi appelés jeux stochastiques à information parfaite. Le groupe nominal information parfaite signifie que les joueurs connaissent les actions jouées par l'autre joueur à chaque état, la connaissance de cette information influence donc l'action du joueur. On peut ainsi supposer que le jeu est séquentiel et, qu'en ajoutant des états, on peut décomposer l'espace d'état en deux types d'états, les états controlés par le joueur $J1$ et les états controlés par le joueur $J2$. On peut aussi ajouter des états non contrôlés qui désignent les calculs intermédiaires.

Finalement, on appelera jeux stochastiques à information parfaite un jeu stochastique tel que pour chaque état $s \in S$, l'un des deux joueurs ne possède qu'une action disponible. On peut ainsi assimiler un jeu stochastique à information parfaite, à un graphe d'ensemble de sommets V et un ensemble d'arcs E. L'ensemble V représente les états du jeu qui est l'union

disjointe des ensembles $S_{\max}$, $S_{\min}$, S_{op} où $S_{\max}$ est l'espace des états contrôlés par $J2$, $S_{\min}$ est l'espace des états contrôlés par $J1$ et S_{op} les autres états. Pour un état $s \in S_{\max} \cup S_{\min}$, les ensembles d'actions du joueur $J1$ sont les ensemble d'arcs $st \in E$, on ajoute également des possibilités d'arrêts $\alpha_s \in \mathbb{R}_+ \cup \{-\infty\}$ pour le maximiseur et $\beta_s \in \mathbb{R}_+ \cup \{+\infty\}$ pour le minimiseur.

L'opérateur de Shapley F s'écrit donc :

$$\begin{aligned} s \in S_{\max} \quad & x \mapsto F_s(x) = \max(\max_{st \in E} x_t + R_t, \alpha_s) \\ s \in S_{\min} \quad & x \mapsto F_s(x) = \min(\min_{st \in E} x_t + R_t, \beta_s) \\ s \in S_{\text{op}} \quad & x \mapsto F_s(x) = P_s \cdot x \end{aligned}$$

On retrouve un jeu stochastique à paiements positifs dès que R_t, α_s et β_s le sont. La valeur du jeu est le plus petit point fixe positif de F et la valeur du jeu est la limite de la suite $(F^n(0))_{n \geq 0}$ dès qu'elle est finie.

Pour assurer que le plus petit point fixe est positif, on peut modifier l'opérateur de Shapley de sorte que les points fixes soit positifs :

$$\begin{aligned} s \in S_{\max} \quad & x \mapsto \tilde{F}_s(x) = \max(\max_{st \in E} x_t + R_t, \alpha_s) \\ s \in S_{\min} \quad & x \mapsto \tilde{F}_s(x) = \min(\min_{st \in E} \max(x_t + R_t, R_t), \beta_s) \\ s \in S_{\text{op}} \quad & x \mapsto \tilde{F}_s(x) = \max(P_t \cdot x, 0) \end{aligned}$$

La fonction $\tilde{F}$ coïncide avec F sur le cône positif et le plus petit point fixe de $\tilde{F}$ est le plus petit point fixe positif de F.

Exemple 9.3 (Application numérique)
Considérons le jeu représenté par le graphe décrit à la figure 9.1.

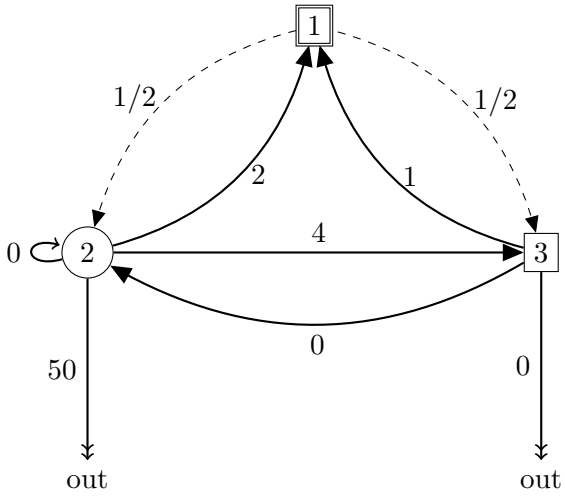


Figure 9.1 – Jeu stochastique à information parfaite de l'exemple 9.3

L'état 1 représente l'ensemble aléatoire et avec probabilité 1/2 le jeu passe de l'état 1 à l'état 2 et avec la même probabilité le jeu passe de l'état 1 à l'état 3. L'état 2 est l'état contrôlé par le minimiseur et l'état 3 est contrôlé par le maximiseur.

Actions de $J1$	
1	aller à l'état 1 en payant 2
2	rester à l'état 2 et ne rien payer
3	aller à l'état 3 et payer 1
4	arrêter le jeu et payer 50

Actions de $J2$	
1	aller à l'état 1 et recevoir 1
2	aller à l'état 2 et ne rien recevoir
3	arrêter le jeu et recevoir 0

L'opérateur de Shapley modifié est donc :

$$x \mapsto F(x) = \begin{pmatrix} \max(1/2x_2 + 1/2x_3, 0) \\ \min(\max(x_1+2,2), \max(x_2,0), \max(x_3+4,4), 50) \\ \max(x_1+1, x_2, 0) \end{pmatrix}$$

Pour utiliser l'itération sur les politiques, il faut initialiser avec les actions du joueur $J1$ et nous choisissons le paiement d'arrêt comme politique initiale.

On doit résoudre le programme linéaire : minimiser $x_1 + x_2 + x_3$ sous les contraintes

$$\begin{gathered} (1/2)x_2 + (1/2)x_3 \leq x_1,\ 0 \leq x_1 \\ 50 \leq x_2 \\ x_1 + 1 \leq x_3,\ x_2 \leq x_3,\ 0 \leq x_3 \end{gathered}$$

On trouve $x_1 = 51$, $x_2 = 50$ et $x_3 = 52$. Ce vecteur est un point fixe de F que l'on note $\overline{x}$. Comme, au point $\overline{x}$, la fonction $x \mapsto \max(x_2, 0)$ atteint le minimum dans $\min(\max(x_1 + 2, 2), \max(x_2, 0), \max(x_3 + 4, 4), 50)$, la semidifférentielle au point $\overline{x}$ est la fonction $f'_{\overline{x}}$ définie par :

$$h \mapsto f'_{\overline{x}}(h) = \begin{pmatrix} 1/2h_2 + 1/2h_3 \\ \min(h_2, 0) \\ h_1 \end{pmatrix}$$

La fonction $f'_{\overline{x}}$ restreinte sur le cône négatif est une application linéaire dont la matrice est stochastique, elle admet donc un point fixe négatif non-nul (par exemple $h_0 = (-1, -1, -1)$). Au point h_0, pour la deuxième coordonnée seule la fonction $h \mapsto h_2$ donne la valeur -1 et on change la politique en considérant $x \mapsto x_2$, on résout le programme linéaire : minimiser $x_1 + x_2 + x_3$ sous les contraintes $(1/2)x_2 + (1/2)x_3 \leq x_1$, $x_2 \leq x_2$, $x_1 + 1 \leq x_3$, $x_2 \leq x_3$ et enfin $x_1, x_2, x_3 \geq 0$, on trouve $x_1 = 1$, $x_2 = 0$ et $x_3 = 2$. Posons $\underline{x} = (1, 0, 2)$.

La semidifférentielle au point $\overline{x}$ est la fonction $f'_{\overline{x}}$ définie par :

$$h \mapsto f'_{\underline{x}}(h) = \begin{pmatrix} 1/2h_2 + 1/2h_3 \\ \max(h_2, 0) \\ h_1 \end{pmatrix}$$

Par l'algorithme 9, on calcule $\gamma_k = \max\limits_{i=1,2,3} \dfrac{(f'_{\underline{x}})^k_i(h)}{h_i}$ en prenant $h \equiv -1$, on trouve pour $k = 1$, $\gamma_k = 1$, $k = 2$, $\gamma_k = 1$ et enfin pour $k = 3$, $\gamma_k = 1/2$, on conclut que $\rho(f'_{\underline{x}}) < 1$ et que $\underline{x}$ est le plus petit point fixe de $\tilde{F}$.

9.2 Exemples et applications à l'analyse statique de programmes

Dans cette section, nous considérons des exemples simples de programmes dont la fonction sémantique abstraite sur les intervalles est une fonction croissante affine par morceaux et contractante. On peut voir la fonction sémantique abstraite sur les intervalles comme un opérateur dynamique de jeu : les unions réprésentent l'opérateur max, les intersections l'opérateur min, les affectations et les constantes les paiements. Dans cette sous-section, pour simplifier les écritures, nous notons $\vee$ à la place de min et $\wedge$ à la place de max.

On rappelle que l'on note un intervalle $I = [-i^-, i^+]$ et les variables (y_i) représentent les intervalles au point de contrôle i.

Exemple 9.4 (Introductif)
Nous avons présenté comme exemple introductif le programme de la figure 9.2.

```
int x,
x=0                 //1
while(x<=99){       //2
      x=1-x;        //3
}                   //4
```

$$\begin{aligned} x_1 &= [0,0] \\ x_2 &= (x_1 \cup x_3) \cap]-\infty, 99] \\ x_3 &= 1 - x_2 \\ x_4 &= (x_1 \cup x_3) \cap [99, +\infty[\end{aligned}$$

Figure 9.2 – Exemple introductif

La difficulté vient uniquement du point de contrôle 2, on s'intéressera uniquement à cet intervalle. On commence par réduire l'équation sémantique du point de contrôle 2 en réécrivant x_3 en fonction de x_2, on obtient $-x_3^- = 1 - x_2^+$ ce qui entraîne $x_3^- = x_2^+ - 1$, et $x_3^+ = 1 - (-x_2^-) = 1 + x_2^-$.

Les bornes de l'intervalle x_2 sont données par les équations min-max suivantes au point de contrôle 2 :

$$\begin{aligned} x_2^- &= 0 \vee (x_2^+ - 1) \\ x_2^+ &= 99 \wedge (0 \vee (x_2^- + 1)) \end{aligned}$$

Pour alléger les notations, on pose, pour (x, y) dans $\mathbb{R}^2$:

$$F(x,y) = \begin{pmatrix} 0 \vee (y-1) \\ 99 \wedge (0 \vee (x+1)) \end{pmatrix}.$$

et trouver x_2^- et x_2^+ revient à chercher (x, y) tel que $F(x, y) = (x, y)$ les plus petits possibles, la variable x jouant le rôle de x_2^- et y celui de x_2^+.

En suivant l'heuristique classique c'est-à-dire, en sélectionnant la fonction $(x, y) \mapsto (0 \vee (y-1), 99)$, on trouve par programmation linéaire, $\bar{x} = 98$ et $\bar{y} = 99$. On vérifie que $(\bar{x}, \bar{y})$ est un point fixe de F.

F est le min-max de fonctions affines contractantes et croissantes, elle est donc F croissante, affine par morceaux et croissante. Nous allons maintenant souligner les fonctions affines qui atteignent le min-max pour le point $(\bar{x}, \bar{y})$.

$$F(\bar{x},\bar{y}) = \begin{pmatrix} 0 \vee \underline{(\bar{y}-1)} \\ \underline{99} \wedge (0 \vee \underline{(\bar{x}+1)}) \end{pmatrix} .$$

D'après la proposition 8.5, F a pour semidifférentielle au point $(\bar{x},\bar{y})$:

$$F'_{(\bar{x},\bar{y})}(h_1,h_2) = \begin{pmatrix} h_2 \\ 0 \wedge h_1 \end{pmatrix} .$$

La fonction $h \mapsto F'_{(\bar{x},\bar{y})}(h)$ est une fonction min-max homogène et on utilise la proposition 8.10 pour calculer le rayon spectral de $F'_{(\bar{x},\bar{y})}$ et un point fixe négatif non-nul s'il existe.

Posons $e \equiv -1$. Comme $F'_{(\bar{x},\bar{y})}(e) = e$ alors le rayon spectral vaut 1 et e est un point fixe négatif non nul, on choisit alors la fonction qui atteint le minimum dans $0 \wedge e_1$, la fonction est donc $h \mapsto h_1$ et donc la nouvelle politique est

$$F(\bar{x},\bar{y}) = \begin{pmatrix} 0 \vee y-1 \\ 0 \vee x+1 \end{pmatrix}$$

et par programmation linéaire, on trouve que $(\bar{x},\bar{y}) = (0,1)$ qui est un point fixe de F. La semidifférentielle au point $(\bar{x},\bar{y})$ est donc la fonction

$$F'_{(\bar{x},\bar{y})}(h_1,h_2) = \begin{pmatrix} 0 \vee h_2 \\ h_1 \end{pmatrix} .$$

En prenant $e \equiv -1$, on a $F'_{(\bar{x},\bar{y})}(e) = (0,-1)$ puis que $F'_{(\bar{x},\bar{y})}(0,-1) = (0,0)$ ce qui implique que le rayon spectral est 0 et qu'en conclusion $(\bar{x},\bar{y}) = (0,1)$ est le plus petit point fixe de F.

Exemple 9.5 (Tiré de [CGG+05])
On considère un simple programme avec une boucle et deux variables.

```
void main(){
int i,j;
i=1;                    //1
j=10;                   //2
while (j>= i){          //3
      i=i+2;            //4
      j=j-1;            //5
      }                 //6
}
```

Comme dans l'exemple 9.4, il n'y a que quelques points de contrôle importants : les points de contrôle 3 et 6 pour ces deux points de contrôle, un calcul d'intersections est nécessaire alors que pour les autres points de contrôle, l'ensemble des politiques est un singleton. Nous exprimons les intervalles des points de contrôle 4 et 5 en fonction de i_3 et j_3. Nous obtenons comme $i_5 = i_3 + 2$ que, $-i_5^- = -i_3^- + 2$ ce qui équivaut à $i_5^- = i_3^- - 2$ et $i_5^+ = i_3^+ + 2$. De $j_5 = j_3 - 1$, nous déduisons que, $-j_5^- = -j_3^- - 1$ ce qui signifie $j_5^- = j_3^- + 1$ et $j_5^+ = j_3^+ - 1$.

Pour calculer les bornes des intervalles aux points de contrôle 3 et 6, il faut résoudre les systèmes d'équations min-max (9.1) et (9.2). On utilise l'heuristique classique et on souligne les politiques initiales.

Le système à résoudre au point de contrôle 3 est donné par le système d'équations (9.1).

$$\left\{\begin{array}{lcc} i_3^- & = & -1 \vee (i_3^- - 2) \\ i_3^+ & = & \underline{(10 \vee (j_3^+ - 1))} \wedge (1 \vee (i_3^+ + 2)) \\ j_3^- & = & \underline{(-1 \vee (i_3^- - 2))} \wedge ((-10 \vee (j_3^- + 1)) \\ j_3^+ & = & 10 \vee (j_3^+ - 1) \end{array}\right. \tag{9.1}$$

Le système à résoudre au point de contrôle 6 est donné par le système d'équations (9.2).

$$\left\{\begin{array}{lcc} i_6^- & = & (-1 \vee (i_3^- - 2)) \wedge \underline{((-10 \vee (j_3^- + 1)) - 1)} \\ i_6^+ & = & 1 \vee (i_3^+ + 2) \\ j_6^- & = & -10 \vee (j_3^- + 1) \\ j_6^+ & = & \underline{((1 \vee (i_3^+ + 2)) - 1)} \wedge (10 \vee (j_3^+ - 1)) \end{array}\right. \tag{9.2}$$

Par programmation linéaire, on obtient les vecteurs suivants :

$$\begin{array}{lcllcl} i_3^- & = & -1 & i_6^- & = & -1 \\ i_3^+ & = & 10 & i_6^+ & = & 12 \\ j_3^- & = & -1 & j_6^- & = & 0 \\ j_3^+ & = & 10 & j_6^+ & = & 11 \end{array}$$

Ces vecteurs sont solutions des systèmes (9.1) et (9.2) implique $j_6^+ = ((1 \vee (i_3^+ + 2)) - 1) \wedge (10 \vee (j_3^+ - 1))$. Or

$$(1 \vee (i_3^+ + 2)) - 1) \wedge (10 \vee (j_3^+ - 1)) = ((1 \vee 12) - 1) \wedge (10 \vee 9) = 11 \wedge 10 = 10$$

et $j_6^+ = 11$.

Le point trouvé précemment n'est pas un point fixe et cela nous conduit à un changement de politique, on prend maintenant $j_2^+ \vee (j_3^+ - 1)$ à la place de $((1 \vee (i_3^+ + 2)) - 1)$ pour la coordonnée j_6^+. Pour trouver le point fixe associé à cette nouvelle politique, on résout un programme linéaire et on trouve :

$$\begin{array}{lcllcl} i_3^- & = & -1 & i_6^- & = & -1 \\ i_3^+ & = & 10 & i_6^+ & = & 12 \\ j_3^- & = & -1 & j_6^- & = & 0 \\ j_3^+ & = & 10 & j_6^+ & = & 10 \end{array}$$

qui est sont des solutions des systèmes (9.1) et (9.2). On note :

$$\begin{array}{rcl} (\bar{\imath}, \bar{\jmath}) & = & (i_3^-, i_3^+, i_6^-, i_6^+, j_3^-, j_3^+, j_6^-, j_6^+) \\ & = & (-1, 10, -1, 12, -1, 10, 0, 10) \end{array} .$$

Nous introduisons la fonction F définie par :

$$F\begin{pmatrix} i \\ j \end{pmatrix} = F\begin{pmatrix} i_3^- \\ i_3^+ \\ i_6^- \\ i_6^+ \\ j_3^- \\ j_3^+ \\ j_6^- \\ j_6^+ \end{pmatrix} = \begin{pmatrix} -1 & \vee & (i_3^- - 2) \\ 10 \vee (j_3^+ - 1) & \wedge & 1 \vee (i_3^+ + 2) \\ -1 \vee (i_3^- - 2) & \wedge & (-10 \vee (j_3^- + 1)) - 1 \\ 1 & \vee & (i_3^+ + 2) \\ -1 \vee (i_3^- - 2) & \wedge & (-10 \vee (j_3^- + 1) \\ 10 & \vee & (j_3^+ - 1) \\ -10 & \vee & (j_3^- + 1) \\ (1 \vee (i_3^+ + 2)) - 1 & \wedge & 10 \vee (j_3^+ - 1) \end{pmatrix}$$

F est croissante contractante et affine par morceaux en tant que min-max d'applications affines croissantes et contractante. Nous soulignons les fonctions qui atteignent le min-max au point $(\bar{\imath}, \bar{\jmath})$.

$$F\begin{pmatrix} \bar{\imath} \\ \bar{\jmath} \end{pmatrix} = \begin{pmatrix} \underline{-1} & \vee & (\overline{\imath_3^-} - 2) \\ \underline{10} \vee (\overline{\jmath_3^+} - 1) & \wedge & 1 \vee (\overline{\imath_3^+} + 2) \\ \underline{-1} \vee (\overline{\imath_3^-} - 2) & \wedge & (-10 \vee \underline{(\overline{\jmath_3^-} + 1)}) - 1 \\ 1 & \vee & \underline{(\overline{\imath_3^+} + 2)} \\ \underline{-1} \vee (\overline{\imath_3^-} - 2) & \wedge & (-10 \vee (\overline{\jmath_3^-} + 1) \\ \underline{10} & \vee & (\overline{\jmath_3^+} - 1) \\ -10 & \vee & \underline{(\overline{\jmath_3^-} + 1)} \\ (1 \vee (\overline{\imath_3^+} + 2)) - 1 & \wedge & \underline{10} \vee (\overline{\jmath_3^+} - 1) \end{pmatrix}$$

Le calcul de la semidifférentielle de F au point $(\bar{\imath}, \bar{\jmath})$ donne, par la proposition 8.5 :

$$F'_{(\bar{\imath},\bar{\jmath})}\begin{pmatrix} \delta\bar{\imath} \\ \delta\bar{\jmath} \end{pmatrix} = F'_{(\bar{\imath},\bar{\jmath})}\begin{pmatrix} \delta\overline{\imath_3^-} \\ \delta\overline{\imath_3^+} \\ \delta\overline{\imath_6^-} \\ \delta\overline{\imath_6^+} \\ \delta\overline{\jmath_3^-} \\ \delta\overline{\jmath_3^+} \\ \delta\overline{\jmath_6^-} \\ \delta\overline{\jmath_6^+} \end{pmatrix} = \begin{pmatrix} 0 & & \\ 0 & & \\ 0 & \wedge & \delta\overline{\jmath_3^-} \\ & \delta\overline{\imath_3^+} & \\ & 0 & \\ & 0 & \\ & \delta\overline{\jmath_3^-} & \\ & 0 & \end{pmatrix}$$

En prenant, $e \equiv -1$, $F'_{(\bar{\imath},\bar{\jmath})}(e) = (0, 0, -1, -1, 0, 0, -1, 0)$ et comme $F'_{(\bar{\imath},\bar{\jmath})}(0, 0, -1, -1, 0, 0, -1, 0) = 0$ et donc le rayon spectral sur $\mathbb{R}^8_-$ de $F'_{(\bar{\imath},\bar{\jmath})}$ est 0 et $(\bar{\imath}, \bar{\jmath})$ est le plus petit point fixe de F.

Exemple 9.6 (Exemple sur un programme avec boucle imbriquée)
Le programme est un programme simple avec une boucle imbriquée.

```
int x,int y,
x=[0,2];y=[10,15]   //1
while(x<=y){        //2
      x=x+1;        //3
     while(5<=y){   //4
          y=y-1;    //5
           }        //6
     }              //7
```

Les points de contrôle intéressants sont les points de contrôle 2, 4, 6 et 7 car ils font intervenir des intersections. On commence par exprimer les autres intervalles en fonction des intervalles de ces points de contrôle, $x_4 = x_3$ et $x_3 = x_2 + 1$, on a aussi $x_4 = x_5 = x_6$. Par $x_3 = x_2 + 1$, on entend, d'une part, $-x_3^- = -x_2^- + 1$ c'est-à-dire $x_3^- = x_2^- - 1$ et $x_3^+ = x_2^+ + 1$.Par ailleurs, $y_5 = y_4 - 1$ ce qui veut dire, d'une part, $-y_5^- = -y_4^- - 1$ qui est équivalant à $y_5^- = y_4^- + 1$ et d'autre part, que $y_5^+ = y_4^+ - 1$

On souligne, les politiques initiales données par l'heuristique classique Pour le point de contrôle 2, on a le système d'équations suivantes :

$$\left\{\begin{array}{lcc} x_2^- & = & x_1^- \vee x_6^- \\ x_2^+ & = & (x_1^+ \vee x_6^+) \wedge \underline{(y_1^+ \vee y_6^+)} \\ y_2^- & = & \underline{(x_1^- \vee x_6^-)} \wedge (y_1^- \vee y_6^-) \\ y_2^+ & = & y_1^+ \vee y_6^+ \end{array}\right.$$

Au point de contrôle 4, le système :

$$\left\{\begin{array}{lcc} y_4^- & = & (y_3^- \vee y_5^-) \wedge \underline{-5} \\ y_4^+ & = & y_3^+ \vee y_5^+ \end{array}\right.$$

Au point de contrôle 6, le système :

$$\left\{\begin{array}{lcc} y_6^- & = & y_3^- \vee y_5^- \\ y_6^+ & = & (y_3^+ \vee y_5^+) \wedge \underline{4} \end{array}\right.$$

Et enfin, au point de contrôle 7, le système :

$$\left\{\begin{array}{lcc} x_7^- & = & (x_1^- \vee x_6^-) \wedge \underline{((y_1^- \vee y_6^-) - 1)} \\ x_7^+ & = & (x_1^+ \vee x_6^+) \\ y_7^- & = & (y_1^- \vee y_6^-) \\ y_7^+ & = & (y_1^+ \vee y_6^+) \wedge \underline{((x_1^+ \vee x_6^+) - 1)} \end{array}\right.$$

Par programmation linéaire, on trouve :

$$\begin{array}{lcllcllcl} x_2^- & = & 0 & y_4^- & = & -5 & x_7^- & = & -1 \\ x_2^+ & = & 15 & y_4^+ & = & 15 & x_7^+ & = & 16 \\ y_2^- & = & 0 & y_6^- & = & 0 & y_7^- & = & 0 \\ y_2^+ & = & 15 & y_6^+ & = & 4 & y_7^+ & = & 15 \end{array}$$

On note :

$$\begin{array}{rcl} (\bar{x}, \bar{y}) & = & (\bar{x_2^-}, \bar{x_2^+}, \bar{x_7^-}, \bar{x_7^+}, \bar{y_2^-}, \bar{y_2^+}, \bar{y_4^-}, \bar{y_4^+}, \bar{y_6^-}, \bar{y_6^+}, \bar{y_7^-}, \bar{y_7^+})^\intercal \\ & = & (0, 15, -1, 16, 0, 15, -5, 15, 0, 4, 0, 15)^\intercal \end{array}$$

On introduit la fonction :

$$F\begin{pmatrix} x \\ y \end{pmatrix} = F\begin{pmatrix} x_2^- \\ x_2^+ \\ x_7^- \\ x_7^+ \\ y_2^- \\ y_2^+ \\ y_4^- \\ y_4^+ \\ y_6^- \\ y_6^+ \\ y_7^- \\ y_7^+ \end{pmatrix} = \begin{pmatrix} 0 & \vee & (x_2^- - 1) \\ 2 \vee (x_2^+ + 1) & \wedge & 15 \vee y_6^+ \\ 0 \vee (x_2^- - 1) & \wedge & (-10 \vee y_6^-) - 1 \\ 0 & \vee & (x_2^+ + 1) \\ 0 \vee (x_2^- - 1) & \wedge & -10 \vee y_6^- \\ 15 & \vee & y_6^+ \\ y_2^- \vee (y_4^- + 1) & \wedge & -5 \\ y_2^+ & \vee & y_4^+ - 1 \\ y_2^- & \vee & y_4^- + 1 \\ y_2^+ \vee (y_4^+ - 1) & \wedge & 4 \\ -10 & \vee & y_6^- \\ 15 \vee y_6^+ & \wedge & (2 \vee (x_2^+ + 1)) - 1 \end{pmatrix}$$

F est croissante, affine par morceaux et contractante en tant que min-max vectoriel d'applications affines croissante et contractante. Par ailleurs, le point $(\bar{x}, \bar{y})$ est un point fixe de F.

On va, comme précédemment souligner les fonctions qui atteignent le min-max pour le point $(\bar{x}, \bar{y})$.

$$F\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = \begin{pmatrix} \underline{0} & \vee & (\bar{x}_2^- - 1) \\ 2 \vee (\bar{x}_2^+ + 1) & \wedge & \underline{15} \vee \bar{y}_6^+ \\ 0 \vee (\bar{x}_2^- - 1) & \wedge & (-10 \vee \underline{\bar{y}_6^-}) - 1 \\ 0 & \vee & \underline{(\bar{x}_2^+ + 1)} \\ \underline{0} \vee (\bar{x}_2^- - 1) & \wedge & -10 \vee \underline{\bar{y}_6^-} \\ \underline{15} & \vee & \bar{y}_6^+ \\ \bar{y}_2^- \vee (\bar{y}_4^- + 1) & \wedge & \underline{-5} \\ \underline{\bar{y}_2^+} & \vee & \bar{y}_4^+ - 1 \\ \underline{\bar{y}_2^-} & \vee & \bar{y}_4^- + 1 \\ \bar{y}_2^+ \vee (\bar{y}_4^+ - 1) & \wedge & \underline{4} \\ -10 & \vee & \underline{\bar{y}_6^-} \\ \underline{15} \vee \bar{y}_6^+ & \wedge & (2 \vee \underline{(\bar{x}_2^+ + 1)}) - 1 \end{pmatrix}$$

La semidifférentielle de F au point $(\bar{x}, \bar{y})$ est, par la proposition 8.5 :

$$F'_{(\bar{x},\bar{y})}\begin{pmatrix} \delta\bar{x} \\ \delta\bar{y} \end{pmatrix} = F\begin{pmatrix} \delta\bar{x}_2^- \\ \delta\bar{x}_2^+ \\ \delta\bar{x}_7^- \\ \delta\bar{x}_7^+ \\ \delta\bar{y}_2^- \\ \delta\bar{y}_2^+ \\ \delta\bar{y}_4^- \\ \delta\bar{y}_4^+ \\ \delta\bar{y}_6^- \\ \delta\bar{y}_6^+ \\ \delta\bar{y}_7^- \\ \delta\bar{y}_7^+ \end{pmatrix} = \begin{pmatrix} 0 & & \\ 0 & & \\ \delta\bar{y}_6^- & & \\ \delta\bar{x}_2^+ & & \\ 0 & \wedge & \delta\bar{y}_6^- \\ 0 & & \\ 0 & & \\ \delta\bar{y}_2^+ & & \\ \delta\bar{y}_2^- & & \\ 0 & & \\ \delta\bar{y}_6^- & & \\ 0 & \wedge & \delta\bar{x}_2^+ \end{pmatrix}$$

En prenant $e \equiv -1$, on obtient successivement que :

$$F'_{(\bar{x},\bar{y})}(e) = (0, 0, -1, -1, -1, -1, 0, 0, -1, -1, 0, -1, -1) = e^1.$$

puis

$$F'_{(\bar{x},\bar{y})}(e^1) = (0, 0, -1, 0, -1, 0, 0, 0, -1, 0, -1, 0) = e^2$$

enfin :

$$F'_{(\bar{x},\bar{y})}(e^2) = (0, 0, -1, 0, -1, 0, 0, 0, -1, 0, -1, 0) = e^2$$

et donc e^2 est un point fixe négatif non nul de $F'_{(\bar{x},\bar{y})}$, le point fixe $(\bar{x},\bar{y})$ n'est pas le plus petit point fixe de F et nous soulignons les applications qui atteignent le min-max dans $F'_{(\bar{x},\bar{y})}(e^2)$:

$$F'_{(\bar{x},\bar{y})}\begin{pmatrix}\delta\bar{x}\\ \delta\bar{y}\end{pmatrix} = \begin{pmatrix} & 0 & \\ & 0 & \\ & \delta\bar{y}_6^- & \\ & \delta\bar{x}_2^+ & \\ 0 & \wedge & \underline{\delta\bar{y}_6^-} \\ & 0 & \\ & 0 & \\ & \delta\bar{y}_2^+ & \\ & \delta\bar{y}_2^- & \\ & 0 & \\ & \delta\bar{y}_6^- & \\ \underline{0} & \wedge & \underline{\delta\bar{x}_2^+} \end{pmatrix}$$

ceci détermine une nouvelle application associée à une politique :

$$g\begin{pmatrix}x\\ y\end{pmatrix} = g\begin{pmatrix} x_2^- \\ x_2^+ \\ x_7^- \\ x_7^+ \\ y_2^- \\ y_2^+ \\ y_4^- \\ y_4^+ \\ y_6^- \\ y_6^+ \\ y_7^- \\ y_7^+ \end{pmatrix} = \begin{pmatrix} 0 & \vee & x_2^- - 1 \\ 15 & \vee & y_6^+ \\ -11 & \vee & y_6^- - 1 \\ 0 & \vee & x_2^+ + 1 \\ -10 & \vee & y_6^- \\ 15 & \vee & y_6^+ \\ & -5 & \\ y_2^+ & \vee & y_4^+ - 1 \\ y_2^- & \vee & y_4^- + 1 \\ & 4 & \\ -10 & \vee & y_6^- \\ 1 & \vee & x_2^+ \end{pmatrix}$$

et par programmation linéaire nous trouvons :

$$(\tilde{u},\tilde{v}) = (0, 15, -5, 16, -4, 15, -5, 15, -4, 4, -4, 15)^\intercal$$

qui est un point fixe de F.

Par la proposition 8.5, le calcul de la semidifférentielle de F au point $(\tilde{u},\tilde{v})$ conduit à la fonction :

$$F'_{(\tilde{u},\tilde{v})}\begin{pmatrix}\delta\tilde{u}\\ \delta\tilde{v}\end{pmatrix} = F\begin{pmatrix}\delta\tilde{u}_2^-\\ \delta\tilde{u}_2^+\\ \delta\tilde{u}_7^-\\ \delta\tilde{u}_7^+\\ \delta\tilde{v}_2^-\\ \delta\tilde{v}_2^+\\ \delta\tilde{v}_4^-\\ \delta\tilde{v}_4^+\\ \delta\tilde{v}_6^-\\ \delta\tilde{v}_6^+\\ \delta\tilde{v}_7^-\\ \delta\tilde{v}_7^+\end{pmatrix} = \begin{pmatrix} & 0 & \\ & 0 & \\ & \delta\tilde{v}_6^- & \\ & \delta\tilde{u}_2^+ & \\ & \delta\tilde{v}_6^- & \\ & 0 & \\ & 0 & \\ & \delta\tilde{v}_2^+ & \\ \delta\tilde{v}_2^- & \vee & \delta\tilde{v}_4^- \\ & 0 & \\ & \delta\tilde{v}_6^- & \\ 0 & \wedge & \delta\tilde{u}_2^+\end{pmatrix}$$

En prenant $e \equiv -1$, on trouve successivement :

$$F'_{(\tilde{u},\tilde{v})}(e) = (0,0,-1,-1,-1,0,0,-1,-1,0,-1,-1) = e^2$$

puis,

$$F'_{(\tilde{u},\tilde{v})}(e^2) = (0,0,-1,0,-1,0,0,0,0,0,-1,0) = e^3$$

et enfin :

$$F'_{(\tilde{u},\tilde{v})}(e^3) = (0,0,0,0,0,0,0,0,0,0,0,0).$$

En conclusion, le point fixe $(\tilde{u},\tilde{v})$ est le plus petit point fixe de F.

9.3 Un mot sur l'implémentation de l'analyseur

La méthode a été implémentée pour les problèmes d'analyse statique de programmes par interprétation abstraite. Le code actuel reprend une précédente implémentation C de l'itération sur les politiques dans le domaine des intervalles. Le prototype ne gère que les programmes C dont les affectations sont les fonctions coordonnées sont des fonctions affines croissantes. Ceci implique que les problèmes de points fixes sur les bornes inférieures et sur les bornes supérieures sont des problèmes indépendants.

Nous allons décrire une exécution du prototype sur l'exemple 9.6. Comme le prototype prend en argument des véritables programme C, nous avons considérons une version légèrement modifiée du programme 9.6, seule l'initialisation change et le programme C analysé est le programme "imbrique.c" décrit à la figure 9.3

Ceci ne change pas la sémantique du programme 9.6 pour les variables x et y mais ajoute uniquement de nouvelles variables qui ne modifient le comportement des variables x et y.

La commande "./parserz imbrique.c" fournit les équations sémantiques en fonction des variables x, y, k et j même si nous n'étudions que les comportements des variables x et y. La sortie tronquée uniquement aux variables x et y de la commande est la suivante :

```
void main()
{
  int x;
  int y;
  int k;
  int j;
  if (k<0)
    x=0;
  else
    x=2;
  if (j<0)
    y=10;
  else
    y=15;
  while (x <= y)
     { x = x+1;
       while (y >= 5)
 y = y+(-1);
      }
}
```

Figure 9.3 – Programme "imbrique.c" analysé par le prototype

```
( x y )[1] = ( x[0], y[0] )
( x y )[2] = ( [0,0], y[1] )
( x y )[3] = ( x[0], y[0] )
( x y )[4] = ( [2,2], y[3] )
( x y )[5] = ( ( x[2] U x[4] ), ( y[2] U y[4] ) )
( x y )[6] = ( x[5], y[5] )
( x y )[7] = ( x[6], [10,10] )
( x y )[8] = ( x[5], y[5] )
( x y )[9] = ( x[8], [15,15] )
( x y )[10] = ( ( x[7] U x[9] ), ( y[7] U y[9] ) )
( x y )[11] = ( ( [-oo, max( y[10] U y[15] )] ^ ( x[10] U x[15] ) ),
              ( [min( x[10] U x[15] ), +oo] ^ ( y[10] U y[15] ) ) )
( x y )[12] = ( ( x[11] + [1, 1] ), y[11] )
( x y )[13] = ( ( x[12] U x[14] ), ( [5, +oo] ^ ( y[12] U y[14] ) ) )
( x y )[14] = ( x[13], ( y[13] + [-1, -1] ) )
( x y )[15] = ( ( x[12] U x[14] ), ( [-oo, 4] ^ ( y[12] U y[14] ) ) )
( x y )[16] = ( ( [min( ( y[10] U y[15] ) + [1, 1] ), +oo] ^ ( x[10] U x[15] ) ),
              ( [-oo, max( ( x[10] U x[15] ) - [1, 1] )] ^ ( y[10] U y[15] ) ) )
```

La deuxième partie concerne les équations min-max à chaque point de contrôle à chaque borne, nous ne donnons que les équations min-max essentielles, pour la variable x :

```
x(-)[2] = 0
x(+)[2] = 0
x(-)[4] = -2
x(+)[4] = 2
x(-)[5] = max(x(-)[2],x(-)[4])
x(+)[5] = max(x(+)[2],x(+)[4])
x(-)[6] = x(-)[5]
x(+)[6] = x(+)[5]
x(-)[7] = x(-)[6]
x(+)[7] = x(+)[6]
x(-)[8] = x(-)[5]
x(+)[8] = x(+)[5]
x(-)[9] = x(-)[8]
x(+)[9] = x(+)[8]
x(-)[10] = max(x(-)[7],x(-)[9])
x(+)[10] = max(x(+)[7],x(+)[9])
x(-)[11] = min(oo,max(x(-)[10],x(-)[15]))
x(+)[11] = min(max(y(+)[10],y(+)[15]),max(x(+)[10],x(+)[15]))
x(-)[12] = (x(-)[11] + -1)
x(+)[12] = (x(+)[11] + 1)
x(-)[13] = max(x(-)[12],x(-)[14])
x(+)[13] = max(x(+)[12],x(+)[14])
x(-)[14] = x(-)[13]
x(+)[14] = x(+)[13]
x(-)[15] = max(x(-)[12],x(-)[14])
x(+)[15] = max(x(+)[12],x(+)[14])
x(-)[16] = min((max(y(-)[10],y(-)[15]) + -1),max(x(-)[10],x(-)[15]))
x(+)[16] = min(oo,max(x(+)[10],x(+)[15]))
```

Ainsi que pour la variable y :

```
y(-)[7] = -10
y(+)[7] = 10
y(-)[9] = -15
y(+)[9] = 15
y(-)[10] = max(y(-)[7],y(-)[9])
y(+)[10] = max(y(+)[7],y(+)[9])
y(-)[11] = min(max(x(-)[10],x(-)[15]),max(y(-)[10],y(-)[15]))
y(+)[11] = min(oo,max(y(+)[10],y(+)[15]))
y(-)[12] = y(-)[11]
y(+)[12] = y(+)[11]
y(-)[13] = min(-5,max(y(-)[12],y(-)[14]))
y(+)[13] = min(oo,max(y(+)[12],y(+)[14]))
y(-)[14] = (y(-)[13] + 1)
y(+)[14] = (y(+)[13] + -1)
y(-)[15] = min(oo,max(y(-)[12],y(-)[14]))
y(+)[15] = min(4,max(y(+)[12],y(+)[14]))
y(-)[16] = min(oo,max(y(-)[10],y(-)[15]))
y(+)[16] = min((max(x(+)[10],x(+)[15]) - 1),max(y(+)[10],y(+)[15]))
```

Le prototype se met en pause jusque la touche "entrée" soit enfoncée et l'itération sur les politiques se met en route. Le prototype utilise l'heuristique classique de la définition 3.10 pour choisir la politique initiale : c'est-à-dire à chaque point de contrôle dès qu'il y a une intersection. Par exemple, au point de contrôle 11, pour la variable x, le prototype retourne le texte suivant :

```
Choix Politique Initiale
Politique gauche pour l'inf pour x[11]: oo
Politique droite pour l'inf pour x[11]: max(x(-)[10],x(-)[15])
Choix politique pour l'inf pour x[11]: max(x(-)[10],x(-)[15])
 -------------------------
Politique gauche pour le sup pour x[11]: max(y(+)[10],y(+)[15])
Politique droite pour le sup pour x[11]: max(x(+)[10],x(+)[15])
Choix politique pour le sup pour x[11]: max(y(+)[10],y(+)[15])
 -------------------------
```

Le prototype appelle maintenant le solveur de programmation linéaire `glpk` pour calculer le plus petit point fixe de la fonction associée à la politique initiale. Il résout en réalité deux problèmes d'optimisation indépendants, le premier traitant uniquement les bornes inférieures et le second uniquement les bornes supérieures :

```
glp_write_lp: writing problem data to 'toto'...
      0: obj =  -4.600000000e+01  infeas =  5.000e+08 (0)
*    23: obj =   2.199999932e+09  infeas =  0.000e+00 (0)
*    39: obj =   2.199999927e+09  infeas =  0.000e+00 (0)
OPTIMAL SOLUTION FOUND

glp_write_lp: writing problem data to 'tata'...
      0: obj =   5.100000000e+01  infeas =  5.000e+08 (0)
*    42: obj =   2.200000201e+09  infeas =  0.000e+00 (0)
OPTIMAL SOLUTION FOUND
```

Un simple test d'égalité dont l'exécution n'est pas affichée par l'analyseur montre que le vecteur trouvé par `glpk` est un point fixe de la fonction min-max. Le prototype retourne alors le vecteur d'intervalles solution :

Le calcul de la semidifférentielle n'est pas affiché par le prototype mais correspond au calcul de l'exemple 9.6. Chaque problème de point fixe est résolu indépendamment. Le prototype calcule la semidifférentielle pour la fonction min-max pour les bornes inférieures ainsi que la semidifférentielle pour la fonction min-max pour les bornes supérieures. La semidifférentielle de la première fonction possède un point fixe négatif non-nul alors que la semidifférentielle de la seconde n'en possède pas. Le prototype affiche le texte suivant :

```
La semidifférentielle a un point fixe négatif non nul, le point fixe
n'est pas le plus petit.
Le seul point fixe négatif pour la semidifférentielle est nul,
l'itération sur les politiques est terminée.
*****************************************************************************
```

La non-minimalité conduit à un changement de politique et le prototype appelle `glpk` pour trouver le plus petit point fixe de cette nouvelle fonction :

```
x[2] = [0,0]
x[4] = [2,2]
x[5] = [0,2]
x[6] = [0,2]
x[7] = [0,2]
x[8] = [0,2]
x[9] = [0,2]
x[10] = [0,2]
x[11] = [0,15]
x[12] = [1,16]
x[13] = [1,16]
x[14] = [1,16]
x[15] = [1,16]
x[16] = [1,16]
```

```
y[7] = [10,10]
y[9] = [15,15]
y[10] = [10,15]
y[11] = [0,15]
y[12] = [0,15]
y[13] = [5,15]
y[14] = [4,14]
y[15] = [0,4]
y[16] = [0,15]
```

```
glp_write_lp: writing problem data to 'toto'...
      0: obj =  -4.600000000e+01  infeas =  5.000e+08 (0)
*    23: obj =   2.199999932e+09  infeas =  0.000e+00 (0)
*    40: obj =   2.199999907e+09  infeas =  0.000e+00 (0)
OPTIMAL SOLUTION FOUND
```

Le simple test d'égalité pour déterminer si le vecteur trouvé par `glpk` est un point fixe de la fonction min-max sur les bornes inférieures. Le vecteur trouvé par `glpk` est bien un point fixe, le prototype retourne ce vecteur (uniquement les bornes inférieures).

```
x(-)2 = 0
x(-)4 = -2
x(-)5 = 0
x(-)6 = 0
x(-)7 = 0
x(-)8 = 0
x(-)9 = 0
x(-)10 = 0
x(-)11 = 0
x(-)12 = -1
x(-)13 = -1
x(-)14 = -1
x(-)15 = -1
x(-)16 = -5
```

```
y(-)7 = -10
y(-)9 = -15
y(-)10 = -10
y(-)11 = -4
y(-)12 = -4
y(-)13 = -5
y(-)14 = -4
y(-)15 = -4
y(-)16 = -4
```

Le prototype calcule la semi-différentielle à ce point (en silence) et la semidifférentielle ne possède pas de point fixe négatif non-nul, le prototype affiche donc le texte suivant :

```
La politique a fourni un point fixe
Itération sur les politiques terminée
Le seul point fixe négatif pour la semidifférentielle est nul,
```

```
l'itération sur les politiques est terminée.
********************************************************************************
Plus petit point fixe trouvé.
```

Finalement, le prototype retourne le vecteur d'intervalles solution :

```
x[2] = [0,0]
x[4] = [2,2]
x[5] = [0,2]
x[6] = [0,2]
x[7] = [0,2]
x[8] = [0,2]
x[9] = [0,2]
x[10] = [0,2]
x[11] = [0,15]
x[12] = [1,16]
x[13] = [1,16]
x[14] = [1,16]
x[15] = [1,16]
x[16] = [5,16]
```

```
y[7] = [10,10]
y[9] = [15,15]
y[10] = [10,15]
y[11] = [4,15]
y[12] = [4,15]
y[13] = [5,15]
y[14] = [4,14]
y[15] = [4,4]
y[16] = [4,15]
```

CHAPITRE 10

CONCLUSION DE LA PARTIE MINIMALITÉ

Ce chapitre est la conclusion sur la partie "A la recherche du plus petit point fixe". Notons que le chapitre 8 a fait l'objet d'une publication [AGG08a]. Nous décomposons ce chapitre de conclusion en deux sections : la première section 10.1 porte sur la caractérisation du plus petit point fixe, cette section est elle-même découpée en quatre sous-sections, la première 10.1.1 étant un résumé du travail sur les caractérisations de minimalité locale, la deuxième sous-section 10.1.2 portant sur les résultats obtenus sur les applications affines par morceaux, la troisième traitant de la caractérisation du plus petit point fixe, enfin, la sous-section 10.1.4 évoquant problèmes à résoudre ainsi que sur les travaux futurs ; la seconde section 10.2 porte, quant à elle, sur le calcul du rayon spectral, cette section est, composée d'un résumé de travaux sur le calcul du rayon spectral à la sous-section 10.2.1 puis d'une sosu-section 10.2.2 concernant les perspectives.

10.1 Caractérisation du plus petit point fixe

10.1.1 Locale minimalité

L'approche utilisée ici est la caractérisation du problème de plus petit point fini comme un problème d'optimisation. On s'est intéressé dans un premier temps à la locale minimalité (définition 8.6), c'est-à-dire que le point fixe u n'admet aucun point fixe plus petit dans un certain voisinage. Les points fixes dans ce voisinage sont soit incomparables soit plus grands que u. De plus, nous considérons des fonctions $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ semidifférentiables (définition 8.5) en tout point fixe de f : dans un certain voisinage d'un point fixe u, la fonction s'approche par une fonction homogène continue f'_u.

Dans un premier théorème 8.19 nous montrons qu'une condition suffisante pour qu'un point fixe soit localement minimal est l'existence d'un point fixe négatif non nul pour la semidifférentielle (une application homogène et continue) au point fixe. Pour une différentielle classique, l'existence d'un point fixe est équivalente à 1 est valeur propre. Pour décider si 1 est valeur propre, il suffit de montrer que le rayon spectral est strictement plus petit que 1. Dans notre cas, puisque la fonction n'est pas nécessairement différentiable, il faut utiliser une définition du rayon spectral compatible avec la semidifférentielle : cet outil noté ρ_C (définition 8.2) existe et est bien défini (proposition 8.2) pour des cônes C convexes fermés et

pointés. Ainsi, toujours dans le théorème 8.3, nous montrons que $\rho_{\mathbb{R}^d_-}(f'_u) < 1$ fournit une condition suffisante de locale minimalité d'un point fixe u.

De plus, dans le théorème 8.2, nous avons montré que si la fonction était croissante et si pour un point fixe u, $\rho_{\mathbb{R}^d_-}(f'_u) > 1$ alors u n'est pas le plus petit point fixe.

10.1.2 Applications affines par morceaux

Les applications affines par morceaux sont des applications semidifférentiables dont le calcul de la semidifférentielle est facile (proposition 8.5). De plus, ces applications ont un comportement similaire aux fonctions affines pour la théorie de la différentielle classique : elles ont un développement limité exact (lemme 8.3). Ce lemme implique le corollaire 8.1 : pour un point fixe u, l'existence d'un point fixe pour f'_u équivaut au fait que u n'est pas localement minimal.

10.1.3 Plus petit point fixe

L'hypothèse importante pour passer du local au global est la contraction au sens large. L'hypothèse de contraction au sens large implique que $\rho_{\mathbb{R}^d_-}(f'_u) \leq 1$. Le théorème 8.3 combiné avec ce résultat entraîne la proposition 8.7 : l'équivalence entre, pour un point fixe u, $\rho_{\mathbb{R}^d_-}(f'_u) < 1$ et f'_u n'admet pas de point fixe négatif non nul. Cette proposition permet de conclure que soit $\rho_{\mathbb{R}^d_-}(f'_u) = 1$ et u n'est pas localement minimal dans le cas affine par morceaux soit $\rho_{\mathbb{R}^d_-}(f'_u) < 1$ et u est localement minimal.

Le point crucial de la contraction au sens large (lemme 8.4) est l'existence d'un retract non-linéaire de $\mathbb{R}^d$ vers l'ensemble des points fixes d'une fonction contractante croissante. Ce retract permet de relier tous les points fixes au plus petit point fixe : le chemin parcourt l'ensemble des points fixes. Puisque le chemin a pour but le plus petit point fixe, il traverse tous les voisinages de tout point fixe, ce qui montre que l'unique point fixe localement minimal est le plus petit point fixe (théorème 8.4). La contraction au sens large implique la connexivité par arcs de l'ensemble des points fixes.

En théorie des jeux, l'hypothèse de contraction au sens large est loin d'être farfelue puisqu'elle concerne l'opérateur de Shapley et les autres opérateurs de programmation dynamique. En analyse statique, la classe des programmes est plus limitée, la restriction porte essentiellement sur les affectations présentes dans le programme qui doivent être des applications contractantes au sens large.

10.1.4 Perspectives

Globale v.s Locale minimalité

Lorsque la fonction n'est pas contractante au sens large, on ne peut que caractériser les points fixes localement minimaux. Dans le cas des contractions, un chemin relie les points fixes, dans le cas général, il faut pouvoir gérer les sauts entre points fixes localement minimaux. Ce problème de saut est illustré par l'exemple 10.1.

Exemple 10.1 (Problème d'ensemble de points fixes discret)
Considérons la fonction f suivante :

$$x \mapsto f(x) = \begin{cases} 1 & \text{si } x \leq 2 \\ 3x - 5 & \text{si } x \in [2, 3] \\ 4 & \text{si } x \geq 3 \end{cases}$$

Le graphe de f est représenté par la figure 10.1.

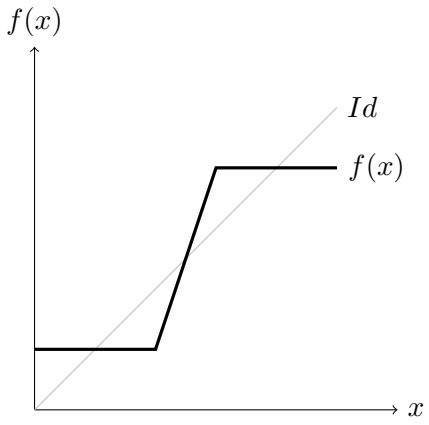


Figure 10.1 – Graphe de f

Il serait également intéressant d'étudier la complexité du problème de recherche de plus petit point fixe.

Locale minimalité et applications affines par morceaux

Un point fixe localement minimal est exactement caractérisé par l'existence d'un point fixe pour la semidifférentielle quand la fonction est affine par morceaux. En optimisation, les optima locaux sont caractérisés par des conditions du second ordre, une généralisation du corollaire 8.1 pourrait passer par une étude des semidifférentielles secondes.

Applications affines par morceaux et théorème de Danskin

Le calcul semidifférentiel des applications affines par morceaux est un cas particulier du théorème de Danskin, il serait intéressant de se baser sur théorème pour caractériser des points fixes localement minimaux En généralisant la classe d'applications, nous pourrons considérer des opérateurs de programmation dynamique de jeu à espaces d'états compacts qui contiennent l'opérateur de Shapley général i.e avec actions mixtes.

Généralisation aux AM-espaces

Dans le passage de locale minimalité au plus petit point fixe, on utilise le fait que $\mathbb{R}^d$ muni de la norme supérieure définit un M-espace abstrait. Une généralisation du théorème 8.4 aux AM-espaces munis de cette structure de treillis semble une bonne perspective de travail.

10.2 Calcul du rayon spectral

10.2.1 Résumé

Deux théorèmes permettent de conclure lorsque le rayon spectral est strictement inférieur à 1 et lorsque le rayon spectral est strictement supérieur à 1. Le réel problème vient de la limite des deux problèmes, c'est-à-dire le cas où le rayon spectral vaut exactement 1. Si le rayon spectral est strictement plus petit ou plus grand que 1 , les méthodes décrites dans la section 8.3.1 peuvent être utilisées même si pour le moment, elles ne sont pas implémentables telles quelles.

Le seul cas positif lorsque le rayon spectral vaut exactement 1 demeure le cas des fonctions min-max homogènes. Le rayon spectral ne prend que deux valeurs : 0 ou 1. Le calcul se basant sur l'itération de l'image du vecteur $e \equiv -1$. Le temps de calcul du rayon spectral est fini et dépend uniquement de la dimension de l'espace vectoriel.

D'un point de vue jeu, les fonctions min-max correspondent à des opérateurs dynamiques de jeu déterministe : les probabilités de transition sont en réalité des masses de Dirac. En analyse statique, la restriction porte essentiellement sur le type d'affectations présentes dans le programme : les affectations sont du type incrémentation de variable, changement de variables, *etc.*

10.2.2 Perspectives

Quand le rayon spectral vaut 1, une première extension serait d'adapter la proposition 8.10 aux fonctions stables sur les rationnels. On utiliserait la structure discrète des rationnels. De plus, on pourrait construire une itération sur les politiques (croissante) pour calculer un point fixe négatif de la semidifférentielle. Lorsqu'un point fixe est le plus petit, l'itération sur les politiques retournerait le point fixe nul, dans le cas contraire, cette itération fournirait une direction de descente.

BIBLIOGRAPHIE

[AB06] C. D. Aliprantis and K. C. Border. *Infinite Dimensional Analysis 3rd Edition*. Springer, 2006.

[AGG] A. Adjé, S. Gaubert, and E. Goubault. Coupling policy iteration with semi-definite relaxation to compute accurate numerical invariants in static analysis. Submitted to Logical Methods in Computer Science.

[AGG08a] A. Adjé, S. Gaubert, and E. Goubault. Computing the smallest fixed point of nonexpansive mappings arising in game theory and static analysis of programs. Technical report, arXiv :0806.1160, Proceedings of MTNS'08, Blacksburg, Virginia, July 2008.

[AGG08b] Xavier Allamigeon, Stéphane Gaubert, and Eric Goubault. Inferring Min and Max Invariants Using Max-plus Polyhedra. In María Alpuente and Germán Vidal, editors, *Proceedings of the 15th International Static Analysis Symposium (SAS'08)*, volume 5079 of *Lecture Notes in Computer Science*, pages 189–204, Valencia, Spain, July 2008. Springer Verlag.

[AGG10] A. Adjé, S. Gaubert, and E. Goubault. Coupling policy iteration with semi-definite relaxation to compute accurate numerical invariants in static analysis. In Andrew Gordon, editor, *Programming Languages and Systems*, volume 6012 of *Lecture Notes in Computer Science*, pages 23–42. Springer Berlin / Heidelberg, 2010.

[AGN] M. Akian, S. Gaubert, and R.D. Nussbaum. Uniqueness of fixed point of non-expansive semidifferentiable maps. *preprint*.

[All09] Xavier Allamigeon. *Static analysis of memory manipulations by abstract interpretation — Algorithmics of tropical polyhedra, and application to abstract interpretation*. PhD thesis, Ecole Polytechnique, Palaiseau, France, November 2009.

[AT07] C. D. Aliprantis and R. Tourky. *Cones and Duality*. American Mathematical Society, 2007.

[Bar77] A. Y. Barraud. A numerical algorithm to solve $a^t x a - x = q$. *Transactions on Automatic Control*, 22 :883–885, 1977.

[BBO92] M. Blair, P. Bridickas, and Sally Obenski. Patriot missile defense, software problem led to system failure at dharhan, saudi arabia. Technical report, US

Government Accountability Office (GAO), February 1992. http://archive.gao.gov/t2pbat6/145960.pdf.

[BCC+02] B. Blanchet, P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Miné, D. Monniaux, and X. Rival. Design and implementation of a special-purpose static program analyzer for safety-critical real-time embedded software, invited chapter. In T. Mogensen, D.A. Schmidt, and I.H. Sudborough, editors, *The Essence of Computation : Complexity, Analysis, Transformation. Essays Dedicated to Neil D. Jones*, LNCS 2566, pages 85–108. Springer-Verlag, October 2002.

[Bec07] A. Beck. Quadratic matrix programming. *SIAM Journal on Optimization*, 17(4) :1224–1238, 2007.

[Bec09] A. Beck. Convexity properties associated with nonconvex quadratic matrix functions and applications to quadratic programming. *Journal of Optimization Theory and Applications*, 142 :1–29, 2009. 10.1007/s10957-009-9539-y.

[BEGFB94] S Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994.

[Bel52] R. E. Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the U.S.A*, 38 :716–719, 1952.

[Bel57] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[BJ72] T.S. Blyth and M.F. Janowitz. *Residuation Theory*. Pergamon press, 1972.

[Bla95] F. Blanchini. Nonquadratic lyapunov functions for robust control. *Automatica*, 31 :451–461, 1995.

[Boa96] Ariane 5 Inquiry Board. Ariane 5 – flight 501 failure. Technical report, European Space Agency, 1996. http://esamultimedia.esa.int/docs/esa-x-1819eng.pdf.

[Bon06] F. Bonnans. *Optimisation continue*. Dunod, 2006.

[Bou93] F. Bourdoncle. Efficient chaotic iteration strategies with widenings. In *In Proceedings of the International Conference on Formal Methods in Programming and their Applications*, pages 128–141. Springer-Verlag, 1993.

[BS95] R.A. Brualdi and B.L. Shader. *Matrices of Sign-Solvable Linear Systems*. Number 116 in Cambridge Tracts in Mathematics. Cambridge University Press, 1995.

[BS00] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer, 2000.

[BTBIR79] A. Ben-Tal, A. Ben-Israel, and E. Rosinger. A helly-type theorem and semi-infinite programming. volume 43, pages 127–135. Academic Press, New York, 1979.

[BTT96] A. Ben-Tal and M. Teboulle. Hidden convexity in some nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 72 :51–63, 1996. 10.1007/BF02592331.

[But30] S. Butterworth. On the theory of filter amplifiers. *Wireless Engineer*, 7 :536–541, 1930.

[BV04] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Press University, 2004.

[CC76] P. Cousot and R. Cousot. Static determination of dynamic properties of programs. In *Proceedings of the Second International Symposium on Programming*, pages 106–130. Dunod, Paris, France, 1976.

[CC77] P. Cousot and R. Cousot. Abstract interpretation : a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the Fourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, California, 1977. ACM Press, New York, NY.

[CC92a] P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation*, 2(4) :511–547, aug 1992.

[CC92b] P. Cousot and R. Cousot. Comparing the Galois connection and widening/narrowing approaches to abstract interpretation, invited paper. In M. Bruynooghe and M. Wirsing, editors, *Proceedings of the International Workshop Programming Language Implementation and Logic Programming, PLILP '92,*, Leuven, Belgium, 13–17 August 1992, Lecture Notes in Computer Science 631, pages 269–295. Springer-Verlag, Berlin, Germany, 1992.

[CE82] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In Dexter Kozen, editor, *Logics of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer Berlin / Heidelberg, 1982.

[CGG+05] A. Costan, S. Gaubert, E. Goubault, M. Martel, and S. Putot. A policy iteration algorithm for computing fixed points in static analysis of programs. In *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05)*, volume 3576 of *LNCS*, pages 462–475, Edinburgh, Jul 2005. Springer.

[CH78] P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *Conference Record of the Fifth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 84–97, Tucson, Arizona, 1978. ACM Press, New York, NY.

[Con93] A. Condon. On algorithms for simple stochastic games. In *Advances in Computational Complexity Theory, volume 13 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 51–73. American Mathematical Society, 1993.

[CTCG+98] J Cochet-Terrasson, G Cohen, S Gaubert, M McGettrick, and J-P Quadrat. Numerical computation of spectral elements in max-plus algebra. In *In Proc. IFAC Conf. on Syst. Structure and Control*, 1998.

[CTGG99] J. Cochet-Terrasson, S. Gaubert, and J. Gunawardena. A constructive fixed point theorem for min-max functions. *Dynamics and Stability of Systems*, 14(4) :407–433, 1999.

[Dan67] J. M. Danskin. *The Theory of Max Min.* Springer, 1967.

[DG06] Vishesh Dhingra and Stéphane Gaubert. How to solve large scale deterministic games with mean payoff by policy iteration. In *Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, valuetools '06, New York, NY, USA, 2006. ACM.

[Dij70] E. Dijkstra. Notes on structured programming. Technical report, Technological University Eindhoven, 1970.

[Doo93] J. L. Doob. *Measure Theory*. Springer, 1993.

[Eve57] H. Everett. Recursive games. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games, Vol III*, volume 39 of *Annals of Mathematics Studies*, pages 47–78. Princeton University Press, 1957.

[FA08a] E Feron and F. Alegre. Control software analysis, part I : Open-loop properties. Technical report, arXiv :0809.4812, 2008.

[FA08b] E. Feron and F. Alegre. Control software analysis, part II : Closed-loop analysis. Technical report, arXiv :0812.1986, 2008.

[Fer04] J. Feret. Static analysis of digital filters. In *European Symposium on Programming (ESOP'04)*, number 2986 in LNCS. Springer-Verlag, 2004. © Springer-Verlag.

[For91] K. Forsman. Construction of lyapunov functions using gröbner bases. In *In Proc. of the 30th Conf. on Decision and Control*, pages 798–799. IEEE, 1991.

[FR91] J. A. Filar and T. E. S Raghavan. Algorithms for stochastic games- a survey. *Zeitschrift für Operations Research*, 35 :437–472, 1991.

[Fri09] Oliver Friedmann. An exponential lower bound for the parity game strategy improvement algorithm as we know it. In *LICS*, pages 145–156. IEEE Computer Society, 2009.

[FV97] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.

[GG04] S. Gaubert and J. Gunawardena. The perron-frobenius theorem for homogeneous monotone functions. *Transactions of AMS*, 356(12) :4931–4950, 2004.

[GGTZ07] S. Gaubert, E. Goubault, A. Taly, and S. Zennou. Static analysis by policy iteration on relational domains. In *Proceedings of European Symposium Of Programming 2007*, Lecture Notes in Computer Science 4421, pages 237–252. Springer, 2007.

[GHK+80] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D.S. Scott. *A compendium of continuous lattices*. Springer, 1980.

[Gil57] D. Gillette. Stochastic games with zero stop probabilities. In M. Dresher, A.W Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games, Volume 3*, pages 179–188. Princeton University Press, 1957.

[Gim06] H. Gimbert. *Jeux positionnels*. PhD thesis, Université Denis Diderot, 2006.

[GJ79] M. R. Garey and D. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H.Freeman & Co Ltd, 1979.

[GKS10] S. Gaubert, R. Katz, and S. Sergeev. Tropical linear programming and parametric mean payoff games. In *Third international Workshop on Invariant Generation (WING'2010)*, July 2010.

[GL02] M. A. Goberna and M. A. López. Linear semi-infinite programming theory : An updated survey. *European Journal of Operations Research*, 143 :390–405, 2002.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[GP] E. Goubault and S. Putot. A zonotopic framework for functional abstraction. preprint.

[GPBG08] E. Goubault, S. Putot, P. Baufreton, and J. Gassino. Static analysis of the accuracy in control systems : Principles and experiments. In *Formal Methods for Industrial Critical System (FMICS 2007)*, volume 4916 of *LNCS*, pages 3–20, 2008.

[GS07a] T. Gawlitza and H. Seidl. Precise fixpoint computation through strategy iteration. In Rocco De Nicola, editor, *Programming Languages and Systems*, volume 4421 of *Lecture Notes in Computer Science*, pages 300–315. Springer Berlin / Heidelberg, 2007. http://dx.doi.org/10.1007/978-3-540-71316-6_21.

[GS07b] T. Gawlitza and H. Seidl. Precise relational invariants through strategy iteration. In Jacques Duparc and Thomas A. Henzinger, editors, *CSL*, volume 4646 of *Lecture Notes in Computer Science*, pages 23–40. Springer, 2007.

[GS10] Thomas Martin Gawlitza and Helmut Seidl. Computing relaxed abstract semantics w.r.t. quadratic zones precisely. In Radhia Cousot and Matthieu Martel, editors, *SAS*, volume 6337 of *Lecture Notes in Computer Science*, pages 271–286. Springer, 2010.

[GSA$^+$] T. Gawlitza, H. Seidl, A. Adjé, S. Gaubert, and E. Goubault. Abstract interpretation meets convex optimization. Submitted to Journal Symbolic of Computation.

[Gun94] J. Gunawardena. Min-max functions. *Discrete Event Dynamic Systems*, 4 :377–406, 1994.

[GVRSS08] F. Guerra Vázquez, J. J. Rückmann, O. Stein, and G. Still. Generalized semi-infinite programming : A tutorial. *J. Comput. Appl. Math.*, 217(2) :394–419, 2008.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6) :1115–1145, 1995.

[Hal74] P. R. Halmos. *Measure Theory*. Springer-Verlag New York, 1974.

[HK66] A. J. Hoffman and Karp. On nonterminating stochastic games. *Management sciences*, 12(5) :359–370, 1966.

[HK93] R. Hettich and K. O. Kortanek. Semi-infinite programming : theory, methods and applications. *SIAM Review*, 35(3) :380–429, 1993.

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10) :576–580, 1969.

[How60] R. Howard. *Dynamic Programming and Markov Processes*. Wiley, 1960.

[JCK07] C. Jansson, D. Chaykin, and C. Keil. Rigorous error bounds for the optimal value in semidefinite programming. *SIAM J. Numer. Anal.*, 46(1) :180–200, 2007.

[JR98] Mikael Johansson and Anders Rantzer. Computation of piecewise quadratic lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 43 :555–559, 1998.

[Kei05] C. Keil. Lurupa - rigorous error bounds in linear programming. In *Algebraic and Numerical Algorithms and Computer-assisted Proofs*, 2005. http://drops.dagstuhl.de/opus/volltexte/2006/445.

[Kil73] G. A. Kildall. A unified approach to global program optimization. In *In Conference Record of the ACM Symposium on Principles of Programming Languages*, pages 194–206. ACM Press, 1973.

[KK00] S. Kim and M. Kojima. Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software*, 15 :201–224, 2000.

[KK03] S. Kim and M. Kojima. Exact solutions of some nonconvex quadratic optimization problems via sdp and socp relaxations. *Computational Optimization and Applications*, 26 :143–154, 2003. 10.1023/A :1025794313696.

[Kle52] S.C. Kleene. *Introduction to Metamathematics*. Bibliotheca Mathematica. North-Holland, 1952.

[LS07a] Jérôme Leroux and Grégoire Sutre. Accelerated data-flow analysis. In *Static Analysis, 14th International Symposium, SAS 2007, Kongens Lyngby, Denmark, August 22-24, 2007, Proceedings*, volume 4634 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2007.

[LS07b] Jérôme Leroux and Grégoire Sutre. Acceleration in convex data-flow analysis. In *Foundations of Software Technology and Theoretical Computer Science, 27th International Conference, FSTTCS 2007, New Delhi, India, December 12-14, 2007, Proceedings*, volume 4855 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 2007.

[LS07c] M. A. López and G. Still. Semi-infinite programming. *European Journal of Operations Research*, 180 :491–518, 2007.

[LT93] N.G. Leveson and C.S. Turner. An investigation of the therac-25 accidents. *IEEE Computer*, 26(7) :18–41, July 1993.

[Lö04] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. http://control.ee.ethz.ch/~joloef/yalmip.php.

[Mar98] F. Martin. PAG – an efficient program analyzer generator. *International Journal on Software Tools for Technology Transfer*, 2(1) :46–67, 1998.

[Mar02] M. Martel. Static analysis of the numerical stability of loops. In Manuel Hermenegildo and Germán Puebla, editors, *Static Analysis*, volume 2477 of *Lecture Notes in Computer Science*, pages 3–14. Springer Berlin / Heidelberg, 2002.

[MBT+04] Glenford J. Myers, Tom Badgett, Todd M. Thomas, Corey Sandler, and Inc ebrary. *The Art of Software Testing*. John Wiley & Sons, Hoboken, N.J, 2nd ed edition, 2004.

[Min01a] A. Miné. A new numerical abstract domain based on difference-bound matrices. In *Proc. of the 2d Symp. on Programs as Data Objects (PADO II)*, volume 2053 of *Lecture Notes in Computer Science*, pages 155–172, Aarhus, Danemark, May 2001. Springer. http://www.di.ens.fr/~mine/publi/article-mine-padoII.pdf.

[Min01b] A. Miné. The octagon abstract domain. In *Proc. of the Workshop on Analysis, Slicing, and Transformation (AST'01)*, IEEE, pages 310–319, Stuttgart, Gernamy, October 2001. IEEE CS Press. http://www.di.ens.fr/~mine/publi/article-mine-ast01.pdf.

[Min04] A. Miné. *Weakly Relational Numerical Abstract Domains*. PhD thesis, École Polytechnique, Palaiseau, France, December 2004. http://www.di.ens.fr/~mine/these/these-color.pdf.

[Mor70] J. J. Moreau. Inf-convultion, sous-additivé, convexité des fonctions numériques. *Journal Mathématiques de Pures et Appliquées*, 49 :109–154, 1970.

[Mor93] J. J. Moré. Generalizations of the trust region problem. *Optimization Methods and Software*, 2 :189–209, 1993.

[MPN02] J. Mallet-Paret and R.D. Nussbaum. Eigenvalues for a class of homogeneous cone maps arising from max-plus operators. *Discrete and Continuous Dynamical Systems*, 8(3) :519–562, July 2002.

[MS96] A. P. Maitra and W. D. Sudderth. *Discrete Gambing and Stochastic Games*. Springer, 1996.

[Nes97] Y. Nesterov. Quality of semidefinite relaxation for nonconvex quadratic optimization. CORE Discussion Papers 1997019, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), March 1997.

[Nes98] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9 :141–160, 1998. 10.1080/10556789808805690.

[NN94] Y. Nesterov and A. Nemirovski. *Interior point polynomial algorithms in convex programming*. Society for Industrial and Applied Mathematics, 1994.

[Nus86] R.D. Nussbaum. Convexity and log convexity for spectral radius. *Linear Algebra And Its Applications*, pages 59–122, 1986.

[Nus88] R.D Nussbaum. Hilbert's projective metric and iterated nonlinear maps. *Memoirs of the AMS*, 75(391), August 1988.

[Ols91] G. J. Olsder. Eigenvalues of dynamical max-min systems. *Discrete Event Dynamic Systems*, 1 :177–207, February 1991.

[Ore44] O. Ore. Galois connexions. *Transactions of American Mathematical Society*, 55 :493–513, 1944.

[Ovc02] S. Ovchinnikov. Max-min representation of piecewise linear functions. *Contributions to Algebra and Geometry*, 43(1) :297–302, 2002.

[Pol98] B. T. Polyak. Convexity of quadratic transformations and its use in control and optimization. *Journal of Optimization Theory and Applications*, 99 :553–583, 1998. 10.1023/A :1021798932766.

[PR97] P.M. Pardalos and M.V. Ramana. Semidefinite programming. In *Interior Point Methods of Mathematical Programming*, pages 369–398. Kluwer Academic Publishers, 1997.

[Put94] M. L. Puterman. *Markov Decision Processes*. Wiley-Interscience, 1994.

[QS82] J. Queille and J. Sifakis. Specification and verification of concurrent systems in cesar. In Mariangiola Dezani-Ciancaglini and Ugo Montanari, editors, *International Symposium on Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Springer Berlin / Heidelberg, 1982.

[Ric53] H. G. Rice. On completely recursively enumerable classes and their key arrays. *Transaction of the American Mathematical Society*, 74 :358–366, 1953.

[Roc74] R. T. Rockafellar. *Conjugate Duality and Optimization*. SIAM, 1974.

[Roc84] R. T. Rockafellar. Directional differentiability of the optimal value function in a nonlinear programming problem. *Mathematical Programming Study*, 21 :213–226, 1984.

[Roc96] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.

[Rub00] A. M. Rubinov. *Abstract Convexity and Global optimization*. Kluwer Academic Publishers, 2000.

[RW98] R.T. Rockafellar and R.J-.B Wets. *Variational Analysis*, chapter 7. Springer, 1998.

[Sch71] H. H. Schaefer. *Topological Vector Spaces*. Springer-Verlag, 1971.

[Sha53] L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39 :1095–1100, 1953.

[Sha05] A. Shapiro. On duality theory of convex semi-infinite programming. *Optimization*, 54(6) :535 – 543, December 2005.

[Sha09] A. Shapiro. Semi-infinite programming, duality, discretization and optimality conditions. *Optimization*, 58 :133–161, 2009.

[Sho87] N. Z. Shor. Quadratic optimization problems. *Tekhnicheskaya Kibernetika*, 1 :128–139, 1987.

[Sim96] V. C. Sima. *Algorithm for Linear-quadratic optimization*. M. Dekker, Inc, New York, 1996.

[Sin97] I. Singer. *Abstract Convex Analysis*. Wiley-Interscience Publication, 1997.

[Sio58] M. Sion. On general minimax theorems. *Pacific Journal of mathematics*, 8 :171–176, 1958.

[Ske92] R. Skeel. Roundoff error and the patriot missile. *SIAM News*, 25(4) :11, July 1992.

[SSM05] S. Sankaranarayanan, H. B. Sipma, and Z. Manna. Scalable analysis of linear systems using mathematical programming. In *The Sixth International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05)*, volume 3385 of *LNCS*, pages 25–41, January 2005.

[Stu99] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12 :625–653, 1999.

[SW95] R. Stern and H. Wolkowicz. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. 5(2) :286–313, 1995.

[Tar55] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2) :285–309, 1955.

[TN01] A. Ben Tal and A. Nemirowski. *Lecture on Modern Convex Optimization : Analysis, Algorithm and Engineering Applications*. Society For Industrial Mathematics, 2001.

[Vav90] Stephen A. Vavasis. Quadratic programming is in np. *Information Processing Letters*, 36(2) :73 – 77, 1990.

[VF06] A. Ismael F. Vaz and Edite M.G.P. Fernandes. Solving semi-infinite programming problems by using an interface between matlab and sipampl. In *Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization*, pages 283–288, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).

[VFG04] A. Ismael F. Vaz, Edite M. G. P. Fernandes, and M. Paula S. F. Gomes. Sipampl : Semi-infinite programming with ampl. *ACM Trans. Math. Softw.*, 30(1) :47–61, 2004.

[VJ00] Jens Vöge and Marcin Jurdzinski. A discrete strategy improvement algorithm for solving parity games. In E. Allen Emerson and A. Prasad Sistla, editors, *CAV*, volume 1855 of *Lecture Notes in Computer Science*, pages 202–215. Springer, 2000.

[vN28] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100 :295–320, 1928.

[Vol06] Michel Volle. *De l'Informatique : Savoir vivre avec l'automate*. Economica, 2006.

[Vor77] N. N. Vorob'ev. *Game Theory. Lectures for Economists and Systems Scientists*. Springer Verlag, Berlin, 1977.

[Win93] G. Winskel. *The formal semantics of programming languages*. The MIT Press, 1993.

[Ye99a] Y Ye. Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming*, 84 :219–226, 1999. 10.1007/s10107980012a.

[Ye99b] Y. Ye. Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming*, 84 :219–226, 1999. 10.1007/s10107980012a.

[Zha00] S. Zhang. Quadratic maximization and semidefinite relaxation. *Mathematical Programming*, 87 :453–465, 2000. 10.1007/s101070050006.

[ZP96] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158 :343–359, 1996.

INDEX