

ABSTRACTIONS DE DIFFÉRENCES EXACTES DE RÉSEAUX DE RÉACTIONS :

AMÉLIORER LA PRÉCISION DE PRÉDICTION DE CHANGEMENTS DE SYSTÈMES BIOLOGIQUES

Emilie ALLART

THÈSE

pour obtenir le grade de Docteur de
l'Université de Lille

Spécialité : INFORMATIQUE

soutenue le 29 janvier 2021

Composition du jury :

<i>Présidente :</i>	Luce BROTCORNE	Inria Lille
<i>Rapporteurs :</i>	Loïc PAULEVÉ	CNRS, Bordeaux
	Jun PANG	Université de Luxembourg
<i>Examineur :</i>	Francois FAGES	Inria Saclay
<i>Directeur :</i>	Joachim NIEHREN	Inria Lille
<i>Co-encadrant :</i>	Cristian VERSARI	Université de Lille

Équipe BioComputing du Laboratoire CRISAL
École Doctorale Sciences Pour l'Ingénieur
Centre de Recherche en Informatique, Signal et Automatique de Lille

Résumé

Des prédictions de changements pour des réseaux de réactions avec information cinétique partielle peuvent être obtenues par raisonnement qualitatif avec l'interprétation abstraite. Un problème de prédiction classique en biologie systémique est de savoir quels knock-outs de gènes peuvent, ou doivent, augmenter le flux de sortie d'une espèce ciblée à l'état stationnaire. Répondre à une telle question pour un réseau de réactions donné demande de raisonner avec des différences abstraites telles que "augmenter" et "diminuer".

Une tâche fondamentale pour des prédictions de changements a été présentée par Niehren et al. (2016). Il s'agit du problème de calcul de l'abstraction de différences d'un ensemble de solutions positives d'un système d'équations linéaires avec des contraintes de différences non linéaires. Précédemment, des algorithmes de surapproximation pour cette tâche ont été proposés en utilisant différentes heuristiques, par exemple basées sur la réécriture des équations linéaires. Dans cette thèse, nous présentons les premiers algorithmes exacts pouvant résoudre cette tâche pour les deux abstractions de différences utilisées dans la littérature.

En guise de première contribution, nous montrons pour un système d'équations linéaires, comment caractériser l'abstraction booléenne de l'ensemble de ces solutions positives. Cette abstraction associe 1 à n'importe quel réel strictement positif, et 0 à 0. La caractérisation est donnée par l'ensemble des solutions booléennes d'un autre système d'équations, qui est obtenu à partir des modes élémentaires. Les solutions booléennes de ce système caractéristique peuvent être calculées en pratique à l'aide de la programmation par contraintes sur les domaines finis. Nous pensons que ce résultat est intéressant pour l'analyse de programmes fonctionnels avec arithmétiques linéaires.

Comme seconde contribution, nous présentons deux algorithmes qui calculent, pour un système d'équations linéaires et de contraintes à différences non linéaires donné, l'abstraction de différences en Δ_3 et respectivement en Δ_6 . Ces algorithmes s'appuient sur la caractérisation des abstractions booléennes pour les systèmes d'équations linéaires issue de la première contribution. Le lien entre ces abstractions est défini en logique du premier-ordre, tel que l'abstraction peut être calculée par la programmation par contraintes sur les domaines finis également.

Nous avons implémenté nos algorithmes exacts et les avons appliqués à la prédiction de knock-outs de gènes qui peuvent mener à une surproduction de leucine dans *B. Subtilis*, nécessaire pour la surproduction de surfactine en biotechnologie. Le calcul des prédictions précises avec l'algorithme exact peut tout de même prendre plusieurs heures. Pour cela, nous présentons aussi une nouvelle heuristique, basée sur les modes élémentaires, pour le calcul de l'abstraction de différences. Celle-ci fournit un bon compromis entre précision et efficacité en temps.

Mots clés: Interprétation abstraite, biologie systémique, équations linéaires, programmation par contraintes, réseaux de réactions avec cinétique partielle.

Abstract

Change predictions for reaction networks with partial kinetic information can be obtained by qualitative reasoning with abstract interpretation. A typical change prediction problem in systems biology is which gene knockouts may, or must, increase the outflow of a target species at steady state. Answering such questions for reaction networks requires to reason about abstract difference such as “increases” and “decreases”.

A task fundamental for change predictions was introduced by Niehren et al. (2016). It is the problem to compute for a given system of linear equations with nonlinear difference constraints, the difference abstraction of the set of its positive solutions. Previous approaches provided overapproximation algorithms for this task based on various heuristics, for instance by rewriting the linear equations. In this thesis we present the first algorithms that can solve this task exactly for the two difference abstractions used in the literature so far.

As a first contribution, we show how to characterize for a linear equation system the boolean abstraction of its set of positive solutions. This abstraction maps any strictly positive real numbers to 1 and 0 to 0. The characterization is given by the set of boolean solutions for another equation system, that we compute based on elementary modes. The boolean solutions of the characterizing equation system can then be computed based on finite domain constraint programming in practice. We believe that this result is relevant for the analysis of functional programs with linear arithmetics.

As a second contribution, we present two algorithms that compute for a given system of linear equations and nonlinear difference constraints, the exact difference abstraction into Δ_3 and Δ_6 respectively. These algorithms rely on the characterization of boolean abstractions for linear equation systems from the first contribution. The bridge between these abstractions is defined in first-order logic. In this way, the difference abstraction can be computed by finite set constraint programming too.

We implemented our exact algorithms and applied them to predicting gene knockouts that may lead to leucine overproduction in *B. Subtilis*, as needed for surfactin overproduction in biotechnology. Computing the precise predictions with the exact algorithm may take several hours though. Therefore, we also present a new heuristics for computing difference abstraction based on elementary modes, that provides a good compromise between precision and time efficiency.

Keywords: Abstract interpretation, systems biology, linear equation systems, constraint programming, reaction networks with partial kinetic information.

Table des matières

1	Introduction	13
1.1	Contexte général	14
1.1.1	Biologie des systèmes	14
1.1.2	Modélisation de systèmes biologiques	14
1.1.3	Modèles mathématiques déterministes	15
1.1.4	Modèles stochastiques	17
1.1.5	Modèles par réseaux de réactions	17
1.1.6	Réseaux sans information cinétique	21
1.2	Contexte local	24
1.2.1	Réseaux avec information cinétique partielle	24
1.2.2	Sémantique de différences des réseaux	25
1.2.3	Tâche de prédiction de changements	28
1.2.4	Algorithme de prédiction de changements	28
1.2.5	Application à la surproduction de la Surfactine	29
1.3	Problématique	30
1.3.1	L’approximation de John	30
1.3.2	Un exemple de surapproximation	31
1.3.3	Calcul exact d’abstractions versus heuristiques	32
1.3.4	L’ajout de contraintes cinétiques	32
1.4	Contributions	32
1.4.1	Réécriture exacte pour l’abstraction booléenne	33
1.4.2	Calcul exact d’abstractions de différence	34
1.4.3	Outil de prédiction de changements et heuristiques	35
1.5	Autres travaux liés	36
1.5.1	Prédiction de knock-out par l’optimisation à deux niveaux	36
1.5.2	Interprétation abstraite	37
1.5.3	Réseaux booléens	37
1.5.4	Projet BestBioSurf	38

2	Préliminaires	43
2.1	Ensembles et fonctions	43
2.2	Σ -algèbres	44
2.3	Σ -structures	45
2.4	Σ -abstractions	47
2.5	Ajout de constantes	48
2.6	Σ -expressions	48
2.7	Polynômes	49
2.8	Équations	50
2.9	Systèmes d'équations	51
2.10	Modes élémentaires	53
2.11	La logique du premier-ordre	55
3	Descriptions logiques de différences	59
3.1	La Σ -algèbre de n -uplets S^n	60
3.2	Abstraction de différences	61
3.2.1	De partition vers abstraction	61
3.2.2	L'abstraction vers Δ_3	62
3.2.3	L'abstraction vers Δ_6	63
3.3	Logique du premier-ordre avec n -uplets	63
3.3.1	Syntaxe et sémantique	64
3.3.2	Liens avec l'algèbre des n -uplets	65
3.3.3	Application aux équations polynomiales	67
3.3.4	Encoder les n -uplets en logique standard	68
3.3.5	Propriété de commutation	71
3.4	Abstraire les différences de solutions	71
3.5	Description logique d'application de fonctions	75
4	Réseaux de réactions avec cinétique partielle	79
4.1	Systèmes de réactions chimiques	80
4.1.1	Sans cinétique	80
4.1.2	Avec cinétique	84
4.1.3	Avec cinétique modulo similarité	88
4.2	Cinétique partielle des modificateurs	89
4.3	Réseaux avec cinétiques partielles: le langage de modélisation	90
4.4	Syntaxe graphique	92
4.5	Exemple du réseau de régulation <i>Pilv – Leu</i>	93
4.5.1	Modélisation biologique	93
4.5.2	Modélisation formelle	94
4.6	Exemple de réseau métabolique	96
4.7	Modes élémentaires	100

5	Prédiction de changements de réseau	103
5.1	Raisonnement intuitif	104
5.2	Abstraction de différences exacte	106
5.3	Approximation de John	108
5.4	Programmation par contraintes à domaine fini	108
5.5	Résoudre des contraintes à différences sur Δ_6	109
5.6	Des solutions de contraintes à la prédiction	110
5.6.1	Algorithme	110
5.6.2	Exemple d'application	111
5.7	Heuristiques	112
5.7.1	Réécriture	112
5.7.2	Conséquences immédiates des modes élémentaires	113
5.8	Prédiction pour la surproduction de surfactine	113
5.9	L'outil BioComputing's Reaction-networks	114
6	Abstraction booléenne de système mixte réel	119
6.1	Introduction	119
6.2	Exemple d'analyse de programme	121
6.3	Exactitude de l'abstraction	122
6.4	Invariance de l'abstraction	127
6.5	Abstraction booléenne de systèmes mixtes	132
6.6	Calcul de l'abstraction de signe	134
7	Calcul exact des abstractions de différences	137
7.1	Introduction	137
7.2	Exact Δ_3 -abstraction	138
7.2.1	Systèmes linéaires	139
7.2.2	Les Structures Mixtes	141
7.2.3	Avec contraintes de différences	141
7.3	Exact Δ_6 -abstraction	144
7.3.1	Systèmes linéaires	144
7.3.2	Avec contraintes de différences	147
8	Heuristique et comparaison	151
8.1	Heuristique des conséquences à support minimal	151
8.2	Résultats expérimentaux	153
9	Conclusion et perspectives	155

Remerciements

Une thèse est une expérience de vie, chaque personne rencontrée au cours de celle-ci y participe que ce soit par leur présence, leur partage, leur bienveillance, ... Je la perçois comme un tableau où chacun y ajoute sa touche de couleur, ou encore une musique bercée par la douce mélodie de chaque personne apparaissant à différents moments.

Naturellement cette thèse ne serait rien sans mes encadrants Joachim Niehren et Cristian Versari que je remercie chaleureusement pour leur confiance, leur soutien et leur encadrement tout au long de ces 4 années. Ils ont su être présents à tour de rôle pour m'accompagner et aller jusqu'au bout de cette aventure comme une équipe. Nous avons tous des méthodes de travail différentes, ils ont su m'épauler avec patience en prenant cela en compte.

Je remercie ensuite grandement Loïc Paulevé et Jun Pang d'avoir accepté de rapporter cette thèse en période de fêtes. Merci également à Luce Brotcorne et François Fages d'avoir pris part au jury, leur présence m'était importante.

Un grand merci ensuite à mon équipe de recherche BioComputing. J'ai passé une grande partie de mon temps de recherche seule et toute personne me connaissant sait à quel point c'était un immense challenge pour moi. Donc avoir ces moments d'échanges avec chaque membre était précieux pour moi, que ce soit Guillaume, Mirabelle et Raphael, Athénaïs, Cédric, Maxime et Debarun. Bien sûr je n'oublie pas Arnaud qui, en plus de me tenir compagnie sur les dernières années de ma thèse, a été mon assistant technique lors de ma soutenance.

Merci Michel pour ces belles discussions autour d'un café, ce sont des échanges enrichissants et en or pour moi. Merci à Quentin, avec qui j'ai fait une bonne partie du chemin universitaire, et sans qui mes pauses auraient manqué de belles discussions diverses et variées !

Au cours de mon cursus universitaire j'ai été accueillie par l'équipe Dolphin et Bonsai qui m'ont transmis l'amour, la passion et la curiosité de la recherche. J'y ai rencontré des femmes en or qui m'ont servies d'exemple; Laetitia, Maude,

Clarisse et Hélène merci à vous.

L'expérience de l'enseignement est sans conteste la plus belle chose qui me soit arrivée de faire. Je remercie mes étudiants pour cela. Et je remercie tout le service d'enseignement que ce soit en tant qu'étudiante, qu'en tant qu'enseignante.

Cette thèse n'aurait pas été si belle sans l'aide de plusieurs magnifiques personnes qui m'ont soutenues, relues et encouragées lors de la dernière étape de rédaction. Julien, Leila, Eglantine, Anne-So, Isabelle, Léa, je vous remercie du fond du coeur.

C'était une épreuve sur la durée, je n'aurai pas su la supporter sans mes amis du cirque, les Intrépides, ils étaient et sont la lumière qui éclaire mon chemin de bonheur. Eglantine, Amaury, Manon, Bastien, Angélique, Anne-So, Fred, Georges, Jérémie, Juliette, ... mission accomplie, on peut prévoir le futur spectacle à présent!

Sophie, tu étais, tu es et tu seras toujours là ... Safi pas besoin de mots.

Maman, papa, Noémie, Fredo cette thèse est pour vous. S'il y a bien des personnes présentes depuis le début c'est vous. Vous m'avez donné un exemple de vie remplie d'amour, de joie, de curiosité, de créativité et de bienveillance. Ce sont les valeurs qui me suivent et qui m'ont amenées avec toutes ces belles personnes là où je suis à présent.

A toute autre personne avec qui j'ai pu échanger des sourires, des discussions, des danses, des chants, des musiques, des balades, ... Si vous vous reconnaissez dans ces mots croyez moi je vous remercie infiniment.

Chapitre 1

Introduction

Sommaire

1.1	Contexte général	14
1.1.1	Biologie des systèmes	14
1.1.2	Modélisation de systèmes biologiques	14
1.1.3	Modèles mathématiques déterministes	15
1.1.4	Modèles stochastiques	17
1.1.5	Modèles par réseaux de réactions	17
1.1.6	Réseaux sans information cinétique	21
1.2	Contexte local	24
1.2.1	Réseaux avec information cinétique partielle	24
1.2.2	Sémantique de différences des réseaux	25
1.2.3	Tâche de prédiction de changements	28
1.2.4	Algorithme de prédiction de changements	28
1.2.5	Application à la surproduction de la Surfactine	29
1.3	Problématique	30
1.3.1	L'approximation de John	30
1.3.2	Un exemple de surapproximation	31
1.3.3	Calcul exact d'abstractions versus heuristiques	32
1.3.4	L'ajout de contraintes cinétiques	32
1.4	Contributions	32
1.4.1	Réécriture exacte pour l'abstraction booléenne	33
1.4.2	Calcul exact d'abstractions de différence	34
1.4.3	Outil de prédiction de changements et heuristiques	35
1.5	Autres travaux liés	36
1.5.1	Prédiction de knock-out par l'optimisation à deux niveaux	36

1.5.2	Interprétation abstraite	37
1.5.3	Réseaux booléens	37
1.5.4	Projet BestBioSurf	38

1.1 Contexte général

1.1.1 Biologie des systèmes

Le fonctionnement de tout organisme vivant repose sur ses interactions moléculaires. La compréhension de ces interactions et de leurs dynamiques est le sujet de la *biologie des systèmes* (Kitano, 2002). C'est un domaine interdisciplinaire alliant mathématiques, informatique, physique, chimie, biologie, et biotechnologie, dont l'approche est d'analyser les systèmes biologiques à l'aide de *modèles*. La biologie des systèmes en informatique est apparue récemment, même si Turing (1952) a commencé des travaux bien avant, la conférence principale sur le domaine CMSB (*Conference on Computational Methods in Systems Biology*) n'a été fondée qu'en 2005.

La connaissance du génome, du transcriptome et du protéome est fondamentale pour la biologie de systèmes, mais insuffisante étant donné qu'il s'agit uniquement de données statiques, et non dynamiques. Ces connaissances sont fournies par la *bioinformatique*. Pour analyser le génome, une approche fréquente est la recherche de motifs et l'alignement de séquences. La *bioinformatique* a par la suite été étendue à d'autres types de données statiques venant de l'étude du *transcriptome* et du *protéome*. On cherche par exemple à savoir quelle partie du génome encode un certain ARN messager (le transcriptome), qui lui-même permet d'exprimer des protéines (le protéome). De cette façon on peut déterminer quelles protéines peuvent être produites à partir de ce gène.

1.1.2 Modélisation de systèmes biologiques

Un problème de la biologie des systèmes est de reconstruire des connaissances pertinentes de la dynamique de systèmes biologiques à partir de données expérimentales. Par exemple, on peut acquérir la connaissance de dépendances entre les molécules par des méthodes d'apprentissage (Rao et al., 2007). Même si celles-ci peuvent permettre la compréhension de la dynamique des systèmes biologiques, elles restent insuffisantes pour les simuler ou prédire les conséquences d'un changement de systèmes, comme par exemple un knock-out de gènes.

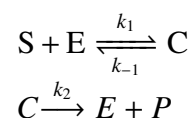
Une idée générale de la biologie des systèmes est donc de créer des modèles de système biologique, qui peuvent être simulés ou qui permettent de faire des prédictions du comportement de ce système. La modélisation est l'art de créer de tels

modèles. Ceci est souvent fait par la coopération interdisciplinaire de biologistes, de chimistes, d'informaticiens et de mathématiciens. Différentes techniques de modélisation existent pour répondre à différents problèmes (De Jong, 2002), qui peuvent être faites à de multiples échelles (Cohen and Harel, 2007) : organisme, tissu, cellule, molécule, gène, etc ... Ces interactions se passent souvent au cours d'une boucle de création de modèles, suivie par une validation via des simulations ou des prédictions. Si la validation échoue, il faut comprendre les raisons, et passer de nouveau à la création d'un meilleur modèle. Ceci peut aussi nécessiter de refaire des expériences pour valider ou réfuter des hypothèses biologiques. Mais dans la plupart des cas, le problème a déjà été résolu dans la littérature biologique, qu'on doit donc parcourir pour chercher et extraire les connaissances manquantes.

1.1.3 Modèles mathématiques déterministes

Les modèles les plus fréquents de la dynamique de systèmes biologiques sont des modèles mathématiques qui sont des systèmes d'*équations différentielles*.

Les équations les plus fréquentes sont les équations différentielles ordinaires (EDOs). Elles permettent de représenter l'évolution au cours du temps des concentrations de chaque espèce d'un système de réactions chimiques de façon déterministe et continue. Lorsque les paramètres des EDOs et les concentrations initiales des espèces sont fixés (par connaissance ou approximation), il est possible de simuler l'évolution des concentrations au cours de temps par ordinateur (Thieffry and De Jong, 2002). On peut voir, à titre d'exemple, avec la FIGURE 1.1 le modèle d'EDOs du cas classique de la dégradation enzymatique. Il décrit la transformation d'un substrat S en un produit P , en présence d'une enzyme E , avec la formation d'un complexe intermédiaire C . La vitesse des réactions dépend de la valeur réelle positive que prennent leurs paramètres cinétiques k_1, k_{-1} et k_2 :

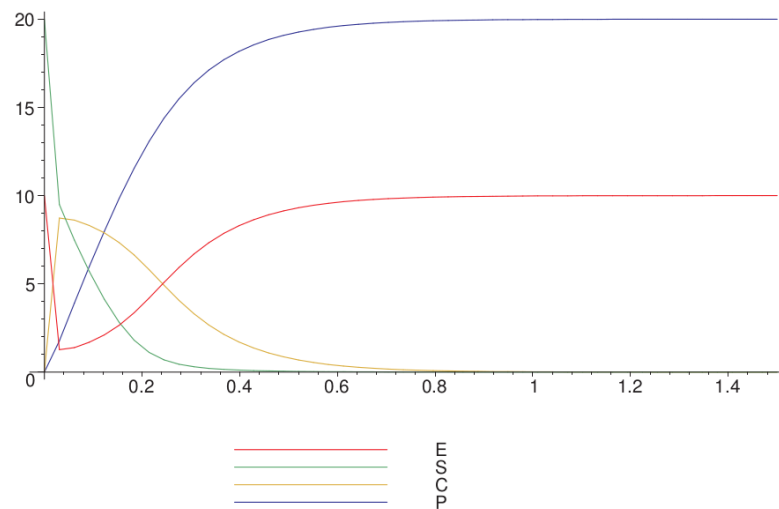


La FIGURE 1.2 montre une simulation de ce modèle, chaque courbe représente l'évolution de la concentration d'une espèce du système dans le temps. Pour cette simulation nous avons fixé la valeur des concentrations initiales comme suit : $C(0) = 0, P(0) = 0, E(0) = 10, S(0) = 20$; et les valeurs des paramètres : $k_1 = 10.0, k_{-1} = 5.0, k_2 = 8.0$.

On retrouve les EDOs par exemple avec le contrôle du cycle cellulaire (Tyson and Novak, 2001), ou encore, pour l'analyse de réseau moléculaire impliqué dans la formation des rythmes circadiens (Leloup and Goldbeter, 2000). Des bases de données, telles que *BioModels* (Chelliah et al., 2013), fournissent des collections

$$\begin{cases} \frac{dE(t)}{dt} = -k_1 E(t)S(t) + k_{-1}C(t) + k_2C(t) \\ \frac{dS(t)}{dt} = -k_1 E(t)S(t) + k_{-1}C(t) \\ \frac{dC(t)}{dt} = k_1 E(t)S(t) - k_{-1}C(t) - k_2C(t) \\ \frac{dP(t)}{dt} = k_2C(t) \end{cases}$$

FIGURE 1.1 – Modèle d'EDOs de la dégradation enzymatique.

FIGURE 1.2 – Trace de la simulation du modèle d'EDOs de la FIGURE 1.1 avec les valeurs des concentrations initiales fixées comme suit : $C(0) = 0$, $P(0) = 0$, $E(0) = 10$, $S(0) = 20$; et les valeurs des paramètres : $k_1 = 10.0$, $k_{-1} = 5.0$, $k_2 = 8.0$.

de modèles de ce type, elles sont souvent exprimées dans le langage SBML (*systems biology markup language*) (Hucka et al., 2003).

Les EDOs peuvent aussi modéliser des aspects de systèmes biologiques, qui ne sont pas directement dérivés à partir d'un réseau de réactions. Par exemple, le pancréas peut être modélisé comme un contrôleur de la glycémie – la concentration du glucose dans le sang. Quand la glycémie augmente au-delà d'un seuil, de l'insuline est sécrétée par le pancréas pour la réduire. Dalla Man et al. (2007) utilise des systèmes d'EDOs pour modéliser le pancréas comme un contrôleur de la glycémie, tel que les modèles puissent simuler l'évolution de la glycémie postprandiale humaine – après un repas.

Quand les paramètres d'un système d'EDOs sont inconnus, on peut essayer de les estimer à partir de données expérimentales. C'est le cas, par exemple, dans le modèle de Dalla Man et al. (2007). Cependant, en général, la qualité des va-

leurs estimées des paramètres ne sera bonne que si les données expérimentales disponibles sont suffisantes pour bien caractériser les paramètres manquants.

Nous notons finalement que des systèmes d'équations différentielles partielles (EDPs) sont utilisés pour modéliser des systèmes biologiques spatiales (voir par exemple [Batmanov \(2014\)](#)). Les systèmes d'équations de réaction-diffusion qui prennent en compte des données spatiales sont des EDPs. On peut voir l'exemple simple de l'équation de Kolmogorov–Petrovsky–Piskunov ([Kolmogorov et al., 1937](#)) qui considère la concentration d'une espèce dans un espace à une dimension. Les solutions des équations de réaction-diffusion permettent de représenter de nombreux comportements, [Turing \(1952\)](#) par exemple décrit l'apparition autonome et naturelle de *patterns* à partir d'un état uniforme et homogène.

1.1.4 Modèles stochastiques

Les approches de modélisation déterministes basées sur des équations différentielles font l'hypothèse implicite que les concentrations des espèces considérées sont présentes en grande quantité, elles ne varient pas énormément sur plusieurs instances d'un même système biologique modélisé. Cependant, lorsque des espèces sont présentes en petit nombre cette hypothèse ne s'applique plus, et il faut passer à la modélisation stochastique.

Les modèles stochastiques les plus fréquents sont les chaînes de Markov à temps continu (CTMC) [Kemeny and Snell \(1976\)](#). Elles formalisent l'évolution d'un système par des transitions entre différents états, et à chaque transition est affectée une probabilité de se produire. Puis, il est possible simuler l'évolution dynamique d'une CTMC à l'aide d'un algorithme de Gillespie ([Gillespie, 1976](#)).

Un exemple pour un modèle stochastique basé sur des CTMCs est l'étude de la régulation de l'expression du bactériophage lambda ([Arkin et al., 1998](#)), ou l'activité des promoteurs transcriptionnels de faible activité ([Kaern et al., 2005](#)). [Kuttler and Niehren \(2006\)](#) montre, qu'à partir des CTMCs de ces modèles ont également été produits des programmes dans le π -calcul stochastique ([John et al., 2008](#) ; [Regev et al., 2000](#)).

1.1.5 Modèles par réseaux de réactions

Des systèmes biologiques peuvent être aussi modélisés directement par des réseaux de réactions chimiques.

Modèles de graphes informels

Des biologistes et biochimistes ont représenté des réseaux de réactions chimiques sous de multiples formes graphiques, pour regrouper leurs connaissances.

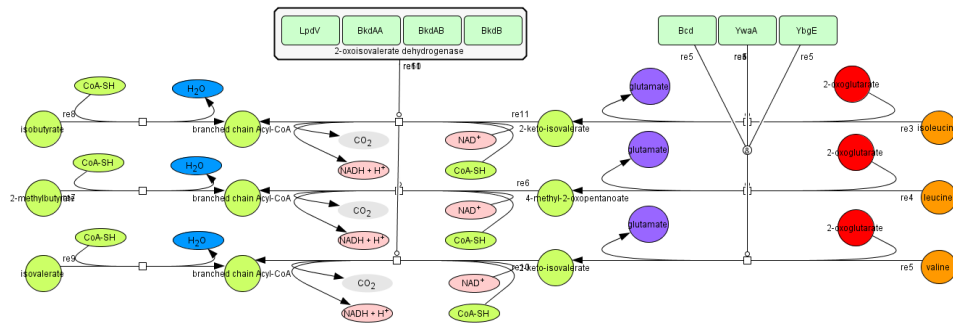


FIGURE 1.3 – Exemple de modèle informel du SubtiWiki, représentant une partie de la voie de synthèse des acides gras chez la bactérie *B. Subtilis*.

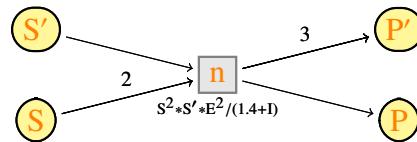


FIGURE 1.4 – Graphe de la réaction chimique avec cinétique (n).

On retrouve, par exemple, le *SubtiWiki* (Mäder et al., 2012) qui regroupe l'ensemble des connaissances biologiques concernant le métabolisme de la bactérie *Bacillus Subtilis*, ou encore la base de données *KEGG* (Kanehisa et al., 2007) et *EcoCyc* (Karp et al., 2002) qui regroupent un ensemble de données relatives aux génomes, aux voies métaboliques et aux composés biochimiques. Un graphe exemple du SubtiWiki est illustré dans la FIGURE 1.3.

Il est possible de faire de la simulation de ces graphes lorsque les données numériques sont disponibles, par l'utilisation d'outils tels que E-cell, Gepasi, BioS-pice. Cependant, il n'est pas possible d'appliquer des raisonnements biologiques sur ces modèles, même pour ceux contenant des valeurs quantitatives, ils ne fournissent pas de sémantique formelle permettant de simuler ou analyser le comportement des systèmes biologiques modélisés.

Modèles formels

Une approche formelle de la définition de systèmes de réactions chimiques a été introduite par Feinberg (1987, 2019). Son formalisme considère un ensemble d'espèces, dont on fait l'abstraction de toute structure interne (on ne regarde ni sa structure, ni sa composition atomique). Une réaction chimique avec de telles

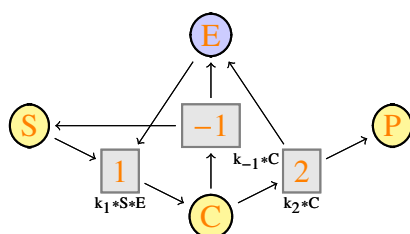
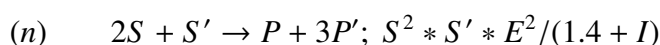


FIGURE 1.5 – Graphe du réseau de réactions avec cinétique de la dégradation enzymatique, avec les paramètres cinétiques k_1, k_2, k_{-1} respectifs des réactions nommées 1, 2, et -1 .

espèces transforme un multiensemble d'espèces comme $2S + S'$ dans un autre multiensemble d'espèces $P + 3P'$ avec une vitesse définie par une expression cinétique, comme par exemple $S^2 * S' * E^2 / (1.4 + I)$. Ici les espèces agissent comme des variables pour l'évolution de leurs concentrations dans le temps, donc du type $\mathbb{R}_+ \rightarrow \mathbb{R}_+$. Cette réaction, qui est illustrée graphiquement par la FIGURE 1.4 s'écrit comme suit :



Les espèces S et S' sont appelées les substrats, comme elles sont consommées par la réaction, et les espèces P et P' sont appelées les produits, qui sont produites par la réaction. Les espèces E et I ne sont ni un substrat, ni un produit, mais elles apparaissent dans l'expression cinétique. On dit qu'elles sont des modificateurs de la réaction. L'expression cinétique de cet exemple exprime que E est une enzyme, qui active et accélère la réaction, tandis que I un inhibiteur, il ralentit la réaction. Bien entendu, il est possible d'avoir de multiples enzymes et inhibiteurs dans ce type de réaction chimique abstraite. Finalement, nous assumons que chaque réaction porte un nom unique, ici (n) , pour pouvoir les identifier plus facilement dans des représentations graphiques.

Un réseau de réactions est un ensemble de réactions chimiques. Chaque réseau peut être représenté graphiquement par un graphe biparti, suivant l'esprit des graphes de réseaux de Petri (1966), aussi appliqué en biologie des systèmes (Herrajy et al., 2017). Le graphe du réseau, avec l'unique réaction décrite ci-dessus, est donné en FIGURE 1.4. Dans le graphe biparti, à chaque réaction est assigné un nœud "boîte", dans lequel le nom de la réaction est inscrit, et auquel l'expression cinétique est annotée. Puis, à chaque espèce est associé un nœud "cercle" dans le graphe, dans lequel le nom de l'espèce est inscrit. Chaque substrat est lié à sa réaction par une arête orientée vers la réaction, sur laquelle la multiplicité du substrat est annotée, si elle est différente de 1. Symétriquement, chaque produit est lié

à sa réaction par une arête orientée vers le produit, sur laquelle la multiplicité du produit est annotée, si elle est différente de 1.

Les graphes bipartis d'un réseau représentent toute l'information du réseau sans perte. De ce fait, nous pouvons et allons définir des réseaux de réactions chimiques par leur graphe. Par exemple le réseau de la dégradation enzymatique en FIGURE 1.5 contient quatre espèces: un substrat S , un complexe C , un produit P , et une enzyme E . Et il a trois réactions: la réaction nommée 1 qui combine un substrat S avec une enzyme E dans un complexe C , sa réaction inverse nommée -1 , et la réaction nommée 2 qui transforme le complexe C en un produit P en libérant une enzyme E .

Sémantiques des réseaux

Un modèle formel est obtenu par le choix de la sémantique d'un réseau de réaction. Fages and Soliman (2008) montrent qu'à chaque réseau de réactions chimiques peut être assigné plusieurs sémantiques : stochastique, déterministe, non-déterministe, booléenne; et que ces sémantiques sont de plus en plus abstraites.

La *sémantique déterministe* est donnée par l'ensemble d'équations différentielles inférées du réseau de réactions, qui décrit l'évolution des concentrations de chaque espèce du système dans le temps. Ainsi, pour chaque espèce A le système contient une équation différentielle qui correspond à la dérivé $\frac{dA}{dt}$. Le système d'EDOs obtenu à partir du réseau de la dégradation enzymatique a été donné auparavant avec la FIGURE 1.1. De plus, une simulation de ce système d'EDOs avec des valeurs fixées pour les paramètres et les concentrations initiales est donnée par la FIGURE 1.2.

La *sémantique stochastique* en terme de CTMC peut être définie pour tous les réseaux de réactions. Les expressions cinétiques servent dans les transitions des CTMCs, voir par exemple (Fages et al., 2015).

Une *sémantique non-déterministe* est obtenue en considérant des réactions chimiques comme des règles de réécriture de solutions chimiques, et en ignorant les expressions cinétiques. Cela est élaboré par Madelaine (2017) ; Madelaine et al. (2014). Il s'agit de la sémantique des réseaux de Petri (1966) classiques. Si lors de la réécriture, plusieurs réactions peuvent être applicables en même temps, on doit choisir une des possibilités d'une manière asynchrone, et regarder tous ses choix possibles. Ainsi, on construit un graphe de réécriture représentant l'ensemble de tous les comportements possibles, et l'on regarde si un attracteur peut ou doit être atteint.

La *sémantique booléenne* offre une sémantique de réécriture non-déterministe alternative, où l'on ne regarde pas le nombre d'espèces à chaque instant, mais uniquement leur existence.

Analyses des réseaux

Selon la sémantique utilisée nous pouvons faire différentes analyses de ces réseaux de réactions, qui servent à comprendre leurs fonctionnements.

Simulation. L'analyse la plus répandue est la *simulation de la dynamique* d'un réseau de réactions. Des algorithmes de simulation différents s'appliquent à la sémantique déterministe différentielle et à la sémantique stochastique avec des CTMC, cela nécessite que les paramètres cinétiques soient connus, et que les concentrations ou le nombre d'espèces initiales soient donnés. La sémantique non-déterministe par réécriture, peut également être simulée étape par étape, en produisant un digraphe de solutions chimiques, et non pas une trace réelle du même type que celle de la FIGURE 1.2.

Vérification de propriétés de modèles. L'outil BioCham permet la vérification de propriétés de modèles, qui ici correspondent à la sémantique, déterministe, stochastique ou booléenne d'un réseau de réactions chimiques (Fages et al., 2004). Ces propriétés peuvent être définies dans une logique temporelle comme LTL, CTL, ou pCTL (Baier and Katoen, 2008) adaptée à la sémantique considérée.

Dans le cas de la sémantique déterministe, on peut vérifier par exemple l'*atteignabilité* d'une solution chimique telle que la concentration d'une espèce d'intérêt soit plus grande que 5.1, ou encore, vérifier si la concentration d'une espèce va *osciller*, étant donné un état de départ. Puis, il est possible de corriger des réseaux pour qu'une propriété logique devienne vraie, ou encore, de chercher la valeur de paramètres cinétiques permettant aux traces du réseau de satisfaire une propriété logique.

Dans le cas de la sémantique stochastique, on peut vérifier par exemple, si la probabilité d'atteindre une solution chimique avec une propriété logique est plus grande que 0.9.

Analyse de sensibilité. L'outil Copasi (Hoops et al., 2006) offre une analyse de sensibilité des concentrations à l'état stable à partir de la valeur des paramètres du réseau. Il permet aussi d'estimer ces paramètres de sorte que les traces de simulation du réseau s'approchent au mieux des données expérimentales.

1.1.6 Réseaux sans information cinétique

Les expressions cinétiques sont souvent inconnues pour réactions chimiques. Cela est vrai en particulier pour des réactions de réseaux informels, sous forme graphique, comme on peut en trouver dans le *SubtiWiki* (Mäder et al., 2012). Une des raisons vient du fait que certaines de leurs réactions sont utilisées pour représenter de façon simplifiée des sous-réseaux plus complexes.

Un autre problème apparaît pour les réseaux métaboliques qui peuvent être fortement régulés mais dont la régulation n'est connue que partiellement. De ce

fait, il n'est pas possible de connaître l'expression cinétique de façon à n'avoir plus qu'à estimer les paramètres.

De plus, les données expérimentales mesurables en métabolomique concernent uniquement les concentrations des métabolites, et non les vitesses de flux qui nous intéresseraient.

Pour toutes ces raisons, nous nous sommes intéressés aux réseaux de réactions chimiques sans information cinétique. Par contre, les analyses quantitatives par simulation ne sont plus applicables. En réponse à cela d'autres types d'analyses basées sur des états stables ont été proposées.

Analyse de l'équilibre des flux

Des réseaux de réactions chimiques, dont les expressions cinétiques sont inconnues, sont utilisés, par exemple, pour décrire les réactions purement métaboliques sans aucune régulation. A l'état stable, ces réseaux décrivent un système de flux à l'équilibre.

L'analyse de flux à équilibre est une approche très utilisée (Orth et al., 2010) pour les réseaux métaboliques sans information cinétique. Son objectif est de prédire le taux de croissance d'un organisme ou le taux de production d'un métabolite d'intérêt. Pour ce faire, la sémantique déterministe du réseau, en termes de EDOs, peut être simplifiée par un système d'équations linéaires dont les variables représentent les vitesses des réactions. Les équations linéaires caractérisent l'état stable : la vitesse de production de chaque espèce doit être égale à la vitesse de sa consommation. Les équations linéaires peuvent aussi être exprimées de manière équivalente par une équation matricielle $\mathbf{Ax} = 0$ où \mathbf{A} est la matrice stœchiométrique du réseau et \mathbf{x} le vecteur de variables de vitesse de ces réactions. Cette approche utilise donc la stœchiométrie du réseau pour inférer des contraintes sur les flux métaboliques. Grâce à la programmation linéaire, on cherche à optimiser une fonction objective fixée selon la prédiction que l'on souhaite faire. Cette approche basée sur l'optimisation est la plus couramment utilisée (Bonarius et al., 1997 ; Kauffman et al., 2003 ; Lee et al., 2006). Cependant, la solution n'est pas forcément unique, elle retourne souvent un ensemble de solutions optimales.

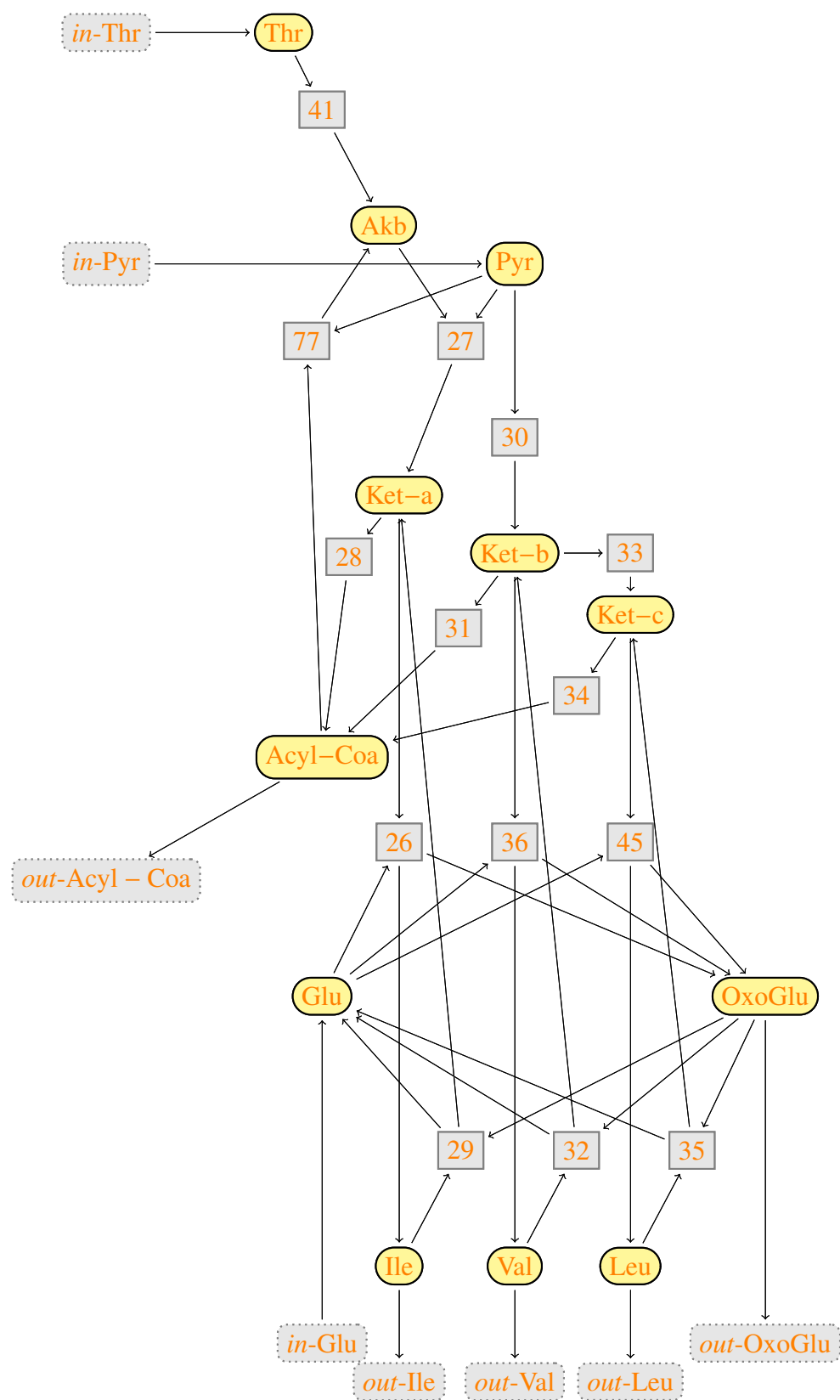


FIGURE 1.6 – Graphe du sous-réseau métabolique de la bactérie *B. subtilis* produisant la leucine à partir de pyruvate.

1.2 Contexte local

Nous présentons dans cette section les travaux de l'équipe BioComputing auxquels cette thèse fait suite. Il s'agit d'une méthode d'analyse qualitative mise en place pour faire la prédiction de knock-out de gènes à partir de réseaux de réactions avec information cinétique partielle ayant pour cible l'augmentation d'une espèce produite par l'organisme modélisé.

1.2.1 Réseaux avec information cinétique partielle

Nous pouvons modéliser les réactions purement métaboliques par un réseau de réactions chimiques sans information cinétique. La FIGURE 1.6 représente graphiquement une partie du métabolisme, sans sa régulation, de la bactérie *B. Subtilis*: la voie de production des acides aminés branchés: valine, leucine et isoleucine à partir du pyruvate et de la thréonine.

Flux d'entrée et flux de sortie. Certaines réactions sont distinguées graphiquement comme des flux d'entrée et d'autres comme des flux de sortie. Dans la FIGURE 1.6, on voit par exemple le flux d'entrée de pyruvate *xPyr* et le flux de sortie de la leucine *yLeu*. La vitesse des flux d'entrée est contrôlée par l'environnement, qui peut modéliser le milieu dans le bioréacteur ou un réseau adjacent. Les vitesses des flux de sortie sont, par contre elles, contrôlées par la concentration de l'espèce qui sort.

Régulation. Le métabolisme de tout organisme est régulé par l'intervention d'enzymes régulatrices, qui peuvent activer, accélérer ou réprimer une réaction. Pour représenter cela, nous allons considérer des réactions avec des modificateurs qui peuvent être des :

inhibiteurs qui diminuent la vitesse d'une réaction,

activateurs qui activent et augmentent la vitesse d'une réaction,

accélérateurs qui augmentent la vitesse d'une réaction.

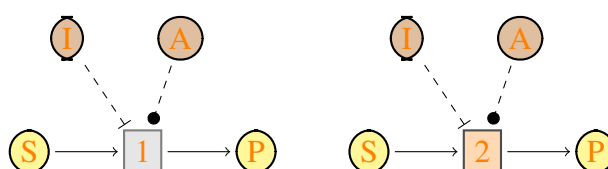


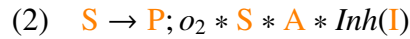
FIGURE 1.7 – La réaction nommée 1 contient un substrat *S*, un produit *P*, un activateur *A* et un inhibiteur *I*. La réaction 1 est en gris et n'est donc pas candidate pour un knock-out. La réaction 2 est similaire sauf qu'elle est en orange et est donc candidate pour un knock-out.

La FIGURE 1.7 donne la représentation graphique d'une réaction nommée 1, qui consomme un substrat **S** et produit **P** sous le contrôle inhibiteur de **I** et l'activation de **A**. Notre formalisme graphique permet aussi de prendre en compte des réactions chimiques possédant plusieurs substrats, produits, activateurs et répresseurs.

Cette réaction est colorée en gris, cela signifie qu'elle n'est pas candidate pour un knock-out. Elle peut donc s'écrire sous forme textuelle comme suit :



où $\text{Inh} : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ est la fonction telle que $\text{Inh}(r) = 1/(1+r)$ pour tout $r \in \mathbb{R}_+$. Des réactions comme la réaction nommée 2, dont la boîte la représentant est orangée, sont candidates pour un knock-out. Elles s'écrivent quant à elles de la sorte :



où o_2 est une variable booléenne, telle que $o_2 \doteq 1$ si et seulement si la réaction 2 est active, et donc ne subit pas de knock-out.

Expressions cinétiques similaires. Cependant, avec ce type de réseaux, nous n'avons pas d'information cinétique précise pour savoir de combien la vitesse d'une réaction est impactée par un de ces régulateurs. Nous pouvons juste constater que si la concentration de **I** diminue, son impact répresseur diminuera et donc la vitesse de la réaction augmentera. La vraie expression cinétique de la réaction 1 est donc similaire à l'expression:

$$\mathbf{S} * \mathbf{A} * \text{Inh}(\mathbf{I})$$

dans le sens qu'elle a le même comportement monotonique. Par exemple, si la concentration du produit **S** augmente alors la valeur de la vraie expression cinétique augmente. De même dans le cas où la concentration de l'activateur **A** augmente, ou si la concentration du répresseur **I** diminue.

1.2.2 Sémantique de différences des réseaux

Les réseaux sont étudiés à l'état stationnaire, on peut donc de façon classique obtenir un ensemble d'équations linéaires pour la partie métabolique, et des contraintes non linéaires pour la régulation modulo similarité. Ceci peut être formalisé en se basant sur les notions de différences concrètes et abstraites.

Différences concrètes et abstraites. Une différence concrète est une paire de réels positifs $(v_{\text{avant}}, v_{\text{après}}) \in \mathbb{R}_+^2$. Elle peut décrire, relativement à un changement d'état stable, la paire composée de la valeur assignée à une variable avant v_{avant} et après $v_{\text{après}}$ un changement. Les variables pourraient par exemple être la vitesse d'un flux d'entrée, d'un flux de sortie, d'une réaction, ou encore la concentration d'une espèce.

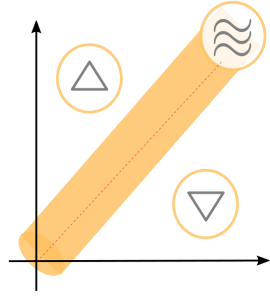


FIGURE 1.8 – L'abstraction de différences $h_{\Delta_3} : \mathbb{R}_+^2 \rightarrow \Delta_3$.

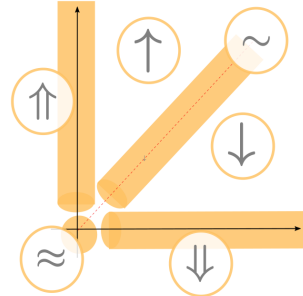


FIGURE 1.9 – L'abstraction de différences $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_6$.

Les différences abstraites partitionnent les différences concrètes dans un ensemble de classe finie. Nous considérons le domaine $\Delta_3 = \{\Delta, \nabla, \approx\}$ avec trois différences abstraites, représentant une augmentation, une diminution, et le non changement respectivement. La partition des différences concrètes par les différences abstraites en Δ_3 est illustrée par la FIGURE 1.8. Elle est définie par la fonction d'abstraction $h_{\Delta_3} : \mathbb{R}_+^2 \rightarrow \Delta_3$, telle que pour tout $(r, r') \in \mathbb{R}_+^2$:

$$h_{\Delta_3}(r, r') = \begin{cases} \Delta & \text{si } r < r' & \text{augmentation} \\ \nabla & \text{si } r > r' & \text{diminution} \\ \approx & \text{si } r = r' & \text{pas de changement} \end{cases}$$

L'abstraction que nous considérons dans notre outil de prédiction est plus fine, elle spécifie le cas où un changement commence à zéro ou finit à zéro. Cette abstraction partitionne les différences concrètes dans les six différences concrètes en $\Delta_6 = \{\uparrow, \downarrow, \approx, \uparrow, \downarrow, \approx\}$ comme l'illustre la FIGURE 1.9. Plus formellement, la partition est définie par la fonction d'abstraction $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_6$, telle que pour tout $r, r' \in \mathbb{R}_+$, nous avons:

$$h_{\Delta_6}(r, r') = \begin{cases} \uparrow & \text{si } 0 \neq r < r' & \text{augmentation, mais pas à partir de 0} \\ \downarrow & \text{si } r > r' \neq 0 & \text{diminution, mais pas jusqu'à 0} \\ \approx & \text{si } r = r' \neq 0 & \text{pas de changement, à valeur non 0} \\ \uparrow & \text{si } 0 = r < r' & \text{augmentation à partir de 0} \\ \downarrow & \text{si } r > r' = 0 & \text{diminution jusqu'à 0} \\ \approx & \text{si } r = r' = 0 & \text{pas de changement à 0} \end{cases}$$

Sémantique de différences. Pour tout affectation de variables $\alpha, \alpha' : \mathcal{V} \rightarrow \mathbb{R}_+$ nous définissons leur différence concrète comme l'affectation $diff(\alpha, \alpha') : \mathcal{V} \rightarrow \mathbb{R}_+^2$ tel que pour toute variable $x \in \mathcal{V}$:

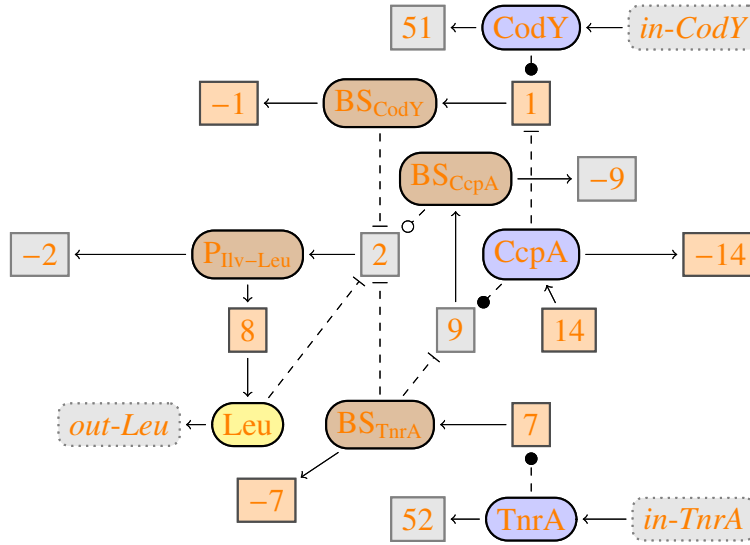


FIGURE 1.10 – Représentation graphique du réseau de réactions avec cinétique partielle pour la régulation d'un promoteur du métabolisme de la bactérie *B. Subtilis*.

$$\text{diff}(\alpha, \alpha')(x) =_{\text{def}} (\alpha(x), \alpha'(x))$$

Considérons maintenant un réseau R avec information cinétique partielle. Soit L son ensemble d'équations linéaires décrivant ses états stables et C l'ensemble de ses contraintes cinétiques. Soit $\Delta \in \{\Delta_3, \Delta_6\}$ un domaine de différences abstraites avec partition $h_\Delta : \mathbb{R}_+^2 \rightarrow \Delta$. La sémantique de Δ -différences de R est définie par:

$$\text{sem}(R) =_{\text{def}} \{h_\Delta \circ \text{diff}(\alpha, \alpha') \mid \alpha, \alpha' \in \text{sol}^{\mathbb{R}_+}(L)\} \cap \text{sol}^\Delta(C)$$

L'intersection avec $\text{sol}^\Delta(C)$ signifie que nous ne considérons que les affectations de variables vers des différences abstraites, qui satisfont les contraintes cinétiques de C . Plus formellement, nous considérons Δ ici comme une structure relationnelle définissant une opération d'addition $+\Delta$, une opération de multiplication $*\Delta$, et une opération d'inversion Inv^Δ . Les tables de ces relations sont données pour Δ_3 et Δ_6 en FIGURE 3.3 et FIGURE 3.4 respectivement. Les résultats de ces opérations peuvent être non-déterministes, en particulier les valeurs de $\Delta +^{\Delta_3} \nabla$ et $\Delta *^{\Delta_3} \nabla$ sont égales, de façon non déterministe, à n'importe quel élément de Δ_3 .

1.2.3 Tâche de prédiction de changements

Un changement est une modification apportée à un flux d'entrée, à savoir une diminution ou une augmentation de la valeur de sa vitesse, ou le knock-out d'une réaction candidate, qui revient à réduire à zéro sa vitesse. Les variables de contrôle d'un réseau sont donc les variables pour les flux d'entrée et les réactions candidates pour un knock-out.

On va chercher à prédire quel changement ou combinaison de changements doivent être appliqués au réseau pour qu'il ait le comportement souhaité. Pour cela on doit fixer la contrainte cible qui doit être satisfaite, dans notre cas il s'agira de l'augmentation du flux de sortie d'une espèce ciblée. Dans cette contrainte, nous ajouterons aussi une limite maximale qui permet de fixer le nombre de changements que l'on autorise. Dans le cas de la prédiction de knock-out unique, nous indiquerons que seulement un changement sur une réaction candidate au knock-out est autorisé. Cette contrainte cible T sera à interpréter sur Δ_6 .

Si l'on regarde le réseau de la FIGURE 1.10, ses variables de contrôle sont $\{v_{x\text{CodY}}, v_{x\text{TnrA}}, v_1, v_7, v_8, v_{14}\}$ et la contrainte $v_{y\text{Leu}} = \uparrow$ sera la contrainte cible qui vise à augmenter le flux de sortie de **Leu**. Soient V l'ensemble des variables de $\text{sem}(R)$ et $V_{\text{contrôle}} \subseteq V$ l'ensemble des variables de contrôle de R . Afin de prédire les changements sur les variables de contrôle qui satisfont la cible, nous allons devoir projeter l'ensemble de solutions abstraites générées sur celles-ci. Nous voulons donc calculer pour un réseau R et une cible T donnés l'ensemble de solutions suivant :

$$\text{sem}_R^{\text{contrôle}}(T) = \{\alpha|_{V_{\text{contrôle}}} \mid \alpha \in \text{sem}(R) \cap \text{sol}^{\Delta_6}(T)\}$$

Pour notre application nous sommes particulièrement intéressés à prédire quel knock-out de réaction unique permettra d'atteindre la cible. Les changements prédits devront donc contenir un knock-out seulement pour être valides.

1.2.4 Algorithme de prédiction de changements

La tâche de prédiction ne peut pas être résolue telle quelle, nous expliquerons pourquoi plus en détail dans la problématique. L'algorithme d'approximation de John (Niehren et al., 2015) nous permet de calculer une surapproximation de cet ensemble de solutions par l'utilisation de l'interprétation abstraite, le théorème montre que l'on a l'inclusion :

$$\text{sem}_R^{\text{contrôle}}(T) \subseteq \text{sol}^{\Delta_6}(\exists \mathcal{V} \setminus V_{\text{contrôle}}. (L \wedge C \wedge T))$$

Cet algorithme reste une heuristique puisqu'il approxime $\text{sem}_R^{\text{contrôle}}(T)$. Ainsi chaque contrainte du système sera interprétée sur la structure Δ_6 . L'algorithme mise en place prend donc en entrée :

- Un réseau de réactions R avec information cinétique partielle C
- Une contrainte cible T

Et il retourne l'ensemble de prédictions obtenues après interprétation des solutions de l'ensemble :

$$\text{sol}^{\Delta_6}(\exists \mathcal{V} \setminus V_{\text{contrôle}} \cdot (L \wedge C \wedge T))$$

Comme la structure Δ_6 est finie, cet ensemble peut être calculé par la programmation par contraintes. L'ensemble des solutions générées pourra être ensuite interprété comme des prédictions, il permet d'énumérer quel changement sur les variables de contrôle permettra d'atteindre la cible.

1.2.5 Application à la surproduction de la Surfactine

Un prototype de l'outil *Reaction-networks* a été développé par l'équipe Bio-Computing, pour tester et appliquer cet algorithme de prédiction sur des problèmes concrets de la biologie des systèmes.

L'application principale jusqu'à ce jour est la prédiction de knock-out de gènes menant à la surproduction d'un métabolite, la leucine, chez la bactérie *B.Subtilis*. La leucine est un acide aminé branché, c'est un des précurseurs de la surfactine, un peptide non ribosomique demandé dans bien des domaines (pharmacologie, agriculture, environnement). Il a été prouvé dans [Coutte et al. \(2015\)](#) qu'une surproduction de leucine chez cette bactérie entraînait une surproduction de surfactine. Cette application a été mise en place en partenariat avec l'équipe ProBioGEM. La [FIGURE 1.11](#) montre le graphe du réseau de réaction avec information cinétique partielle développé pour cette application. Il ne représente qu'une partie du métabolisme de la bactérie, il s'agit de la voie des acides aminés branchés: valine, leucine et isoleucine.

L'outil permet depuis 2018 de créer des réseaux de réactions avec information cinétique partielle depuis un navigateur Web, par l'utilisation d'une interface graphique affichant le graphe du réseau. Cette interface est accessible, pour preuve il a été utilisé avec succès par des biologistes sans formation en informatique.

L'outil permet de faire des prédictions de knock-out à partir de réseau de réaction avec information cinétique partielle. Cela demande par contre aux utilisateurs une expertise dans ce domaine de recherche en informatique pour pouvoir le manipuler.

L'application de l'algorithme de prédiction sur ce réseau de réactions, présenté dans [Niehren et al. \(2015\)](#) montre que si l'on fixe pour cible une augmentation du flux de sortie de la leucine avec un maximum de deux changements l'algorithme retourne 16 prédictions de knock-outs de réaction. Nous reviendrons dessus plus tard dans cette thèse.

1.3 Problématique

Notre objectif principal est le calcul exact de $sem_R^{\text{control}}(T)$ pour des réseaux de réactions à information cinétique partielle R et une cible T . Ceci nous demande, en particulier pour le système d'équations linéaires L de R , de calculer l'ensemble de différences abstraites obtenues à partir des différences concrètes de solutions réelles de L , en appliquant l'abstraction $h_\Delta : \mathbb{R}_+^2 \rightarrow \Delta$ où $\Delta \in \{\Delta_3, \Delta_6\}$, c'est-à-dire:

$$\{h_\Delta \circ \text{diff}(\alpha, \alpha') \mid \alpha, \alpha' \in \text{sol}^{\mathbb{R}_+}(L)\}$$

Il s'agit d'éviter l'interprétation abstraite basée sur l'approximation de John, pour arriver à un calcul exact de l'abstraction.

1.3.1 L'approximation de John

La partition $h_\Delta : \mathbb{R}_+^2 \rightarrow \Delta$ devient un homomorphisme, si nous équipons \mathbb{R}_+^2 avec l'addition et la multiplication appliquées sur chaque composante à l'aide de $+^{\mathbb{R}_+}$ et $*^{\mathbb{R}_+}$ respectivement. Puis, il n'est pas difficile de voir que :

$$\text{sol}^{\mathbb{R}_+^2}(L) = \{\text{diff}(\alpha, \alpha') \mid \alpha, \alpha' \in \text{sol}^{\mathbb{R}_+}(L)\}$$

L'objectif est donc de calculer l'ensemble des solutions abstraites :

$$h_\Delta \circ \text{sol}^{\mathbb{R}_+^2}(L)$$

Nous appelons un homomorphisme $h : S \rightarrow \Delta$, tel que $\text{dom}(\Delta) \subseteq \text{dom}(S)$, une *abstraction*. Puis, nous considérons l'inclusion $\Delta \subseteq \mathbb{R}_+^2$ en identifiant chaque élément $\delta \in \Delta$ par un élément $p \in \mathbb{R}_+^2$ fixé tel que $h_\Delta(p) = \delta$. Donc, $h_{\Delta_3} : \mathbb{R}_+^2 \rightarrow \Delta_3$ et $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_6$ sont bien des abstractions avec la définition donnée ci-dessus.

Notre première observation est que le Théorème de John peut être généralisé à toute abstraction $h : S \rightarrow \Delta$. Il montre que pour toute formule du premier-ordre ϕ , existentielle et positive, l'ensemble des abstractions des solutions de ϕ dans la structure concrète S est inclus dans l'ensemble des solutions de la structure abstraite Δ :

$$h_\Delta \circ \text{sol}^S(\phi) \subseteq \text{sol}^\Delta(\phi)$$

Ici nous sommes particulièrement intéressés dans le cas où ϕ est un système d'équations linéaires L , obtenu par les équations issues d'un réseau de réactions métaboliques à l'état stable sans connaissance des paramètres cinétiques.

1.3.2 Un exemple de surapproximation

En guise d'illustration de cette approximation, nous considérons l'abstraction de différences $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_6$, et l'application à la prédiction de changements au réseau métabolique en FIGURE 1.12. Ce réseau contient un cycle entre deux espèces **A** et **B** données par une réaction 1 qui transforme **A** en **B** et la réaction inverse 2. Puis, l'espèce **A** a un flux d'entrée et l'espèce **B** un flux de sortie.

Le flux d'entrée *in-A* est contrôlé à l'externe, tandis que toutes les autres réactions, incluant le flux de sortie *out-B*, sont contrôlées à l'interne du réseau (par les concentrations des espèces et les cinétiques du réseau). Les seuls changements sous contrôle externe pour ce réseau sont l'augmentation ou la diminution du flux d'entrée *in-A*. Ceci correspond aux équations avec des constantes de Δ_6 :

$$v_{in-A} \doteq \uparrow \quad \text{or} \quad v_{in-A} \doteq \downarrow$$

Si maintenant nous voulons augmenter le flux de sortie, c'est-à-dire $v_{out-B} \doteq \uparrow$, le seul changement qui peut nous amener à cette cible est l'augmentation du flux d'entrée *in-A*, alors : $v_{in-A} \doteq \uparrow$. Pour illustrer ce type de raisonnement avec notre méthode formelle, nous considérons le système d'équations linéaires ϕ qui décrit l'équilibre des flux de toutes les réactions à l'état stationnaire (Orth et al., 2010 ; Papin et al., 2004 ; Varma and Palsson, 1994).

$$v_1 \doteq in-A + v_2 \quad \wedge \quad v_1 \doteq out-B + v_2$$

La FIGURE 1.13 nous donne les trois solutions de $sol^{\Delta_6}(\phi)$ qui satisfont la cible $out-B \doteq \uparrow$. Les deux premières de ces trois solutions, par contre, ne sont pas des h_{Δ_6} -abstractions de solutions réelles, puisque toute solution réelle doit satisfaire :

$$in-A \doteq out-B$$

Comme cette égalité est une conséquence directe de ϕ sur les réels, donc aussi bien sur la structure \mathbb{R}_+ que sur la structure \mathbb{R}_+^2 : la vitesse du flux d'entrée doit être égale à la vitesse du flux de sortie à l'état stationnaire. Considérons le système linéaire ϕ' suivant, dans laquelle la cible du changement est incluse et les variables non contrôlables sont quantifiées existentiellement :

$$\exists v_1 \exists v_2 \exists v_{out-B}. (\phi \wedge v_{out-B} \doteq \uparrow)$$

Nous avons donc bien le fait que l'abstraction $h_{\Delta_6} \circ sol^{\mathbb{R}_+^2}(\phi') = \{[v_{in-A} / \uparrow]\}$ est surapproximée par l'interprétation abstraite $sol^{\Delta_6}(\phi') = \{[v_{in-A} / \uparrow], [v_{in-A} / \downarrow], [v_{in-A} / \sim]\}$ comme impliqué par l'approximation de John.

1.3.3 Calcul exact d'abstractions versus heuristiques

L'exemple montre aussi que la surapproximation de l'abstraction $h_{\Delta_6} \circ sol^{\mathbb{R}_+^2}(\phi)$ par l'interprétation abstraite $sol^{\Delta_6}(\phi)$ peut empêcher la prédiction de changements valides. La cause de la surapproximation est due ici au fait que ϕ n'est pas clos par conséquences linéaires, c'est-à-dire qu'il ne contient pas toutes les équations linéaires qui sont impliquées par ϕ dans la structure \mathbb{R}_+ , et donc également \mathbb{R}_+^2 . Puis, l'addition des conséquences linéaires manquantes à ϕ avant l'interprétation abstraite empêche la surapproximation.

Par contre, comme l'ensemble de conséquences linéaires est généralement infini, il n'est pas possible d'ajouter toutes les conséquences linéaires du système avant de calculer son interprétation abstraite. De ce fait, le problème du calcul exact ne peut pas être résolu comme tel. Nous pouvons seulement obtenir une meilleure approximation en nous basant sur des heuristiques qui ajoutent un nombre fini de conséquences linéaires. La qualité des heuristiques existantes, par contre, est difficile à argumenter, sans pouvoir calculer l'abstraction exactement.

1.3.4 L'ajout de contraintes cinétiques

Pour des réseaux de réactions avec cinétique partielle le problème est encore plus compliqué puisque les cinétiques possibles peuvent être contraintes. Ces contraintes peuvent être exprimées par des formules logiques à interpréter sur Δ_6 . Le problème à résoudre est donc de calculer exactement les h_{Δ_6} -abstractions des solutions de systèmes linéaires qui satisfont des contraintes supplémentaires. Et ceci, pas uniquement en théorie, mais de telle sorte que cela puisse être appliqué pour la prédiction de changement de réseau de réactions avec information cinétique partielle, et en particulier à la prédiction de knock-out de gènes pour de tels réseaux.

1.4 Contributions

Notre objectif principal est le calcul exact de l'abstraction de différence en Δ_3 ou Δ_6 de l'ensemble des solutions en \mathbb{R}_+^2 d'un système linéaire, et l'application de ce calcul à la prédiction de changement de réseau de réactions avec cinétique partielle.

Plus généralement, étant donné une abstraction h_{Δ} d'une structure concrète S vers une structure abstraite Δ , nous appelons une formule ϕ h_{Δ} -exacte si pour chaque solution abstraite de ϕ , il existe une solution concrète qui correspond après abstraction à cette solution abstraite. Donc, ϕ sera h_{Δ} -exacte si et seulement si :

$$h_{\Delta} \circ sol^S(\phi) = sol^{\Delta}(\phi)$$

Supposons maintenant que le domaine de la structure abstraite Δ est fini. Si nous pouvions en plus réécrire chaque formule ϕ comme une formule ϕ' qui est h_Δ -exacte, nous pourrions résoudre le problème du calcul exact de $h_\Delta \circ \text{sol}^S(\phi)$ en calculant $\text{sol}^\Delta(\phi')$ par l'intermédiaire de la programmation par contraintes à domaine fini. Malheureusement, après de longues réflexions faites au cours de ce projet de thèse, cela nous paraît difficilement envisageable pour les abstractions h_{Δ_3} et h_{Δ_6} même si les formules sont réduites à des systèmes d'équations linéaires.

1.4.1 Réécriture exacte pour l'abstraction booléenne

Dans cette optique, nous proposons d'étudier, en premier lieu, l'abstraction booléenne $h_{\mathbb{B}} : \mathbb{R}_+ \rightarrow \mathbb{B}$ définie pour tout $r \in \mathbb{R}_+$ par :

$$h_{\mathbb{B}}(r) = \begin{cases} 1 & \text{si } r > 0 \\ 0 & \text{si } r = 0 \end{cases}$$

Nous considérons comme formules des systèmes d'équations linéaires à coefficients entiers $\mathbf{A}\mathbf{y} \doteq 0$. Les solutions de telles formules dans la structure des réels positifs \mathbb{R}_+ peuvent donc être abstraites par l'abstraction booléenne $h_{\mathbb{B}}$.

Notre première observation est que tout système d'équations linéaires avec des matrices à coefficients *positifs* est $h_{\mathbb{B}}$ -exact. Si par contre nous prenons des matrices d'entiers possédant aussi des entiers négatifs, ce n'est plus nécessairement le cas. L'idée pour un calcul exact de l'abstraction booléenne est donc de réécrire des équations linéaires avec des coefficients négatifs de manière positive, sans changer les solutions en \mathbb{R}_+ . C'est ce que nous allons montrer en guise de première contribution.

En effet, nous montrerons que chaque système d'équations linéaires à coefficients entiers et sans terme constant, ou d'une manière équivalente, chaque équation matricielle :

$$\mathbf{A}\mathbf{y} \doteq 0$$

avec \mathbf{A} une matrice à coefficients entiers et \mathbf{y} un vecteur de variables distinctes, se réécrit dans un autre système linéaire \mathbb{R} -équivalent de la forme

$$\exists \mathbf{x}. E\mathbf{x} = \mathbf{n}\mathbf{y}$$

où E est une matrice d'entiers positifs, \mathbf{x} un vecteur de variables distinctes et \mathbf{n} un vecteur de naturels. Puis, nous montrerons que ce nouveau système est en effet $h_{\mathbb{B}}$ -exact. La réécriture dans un système $\exists \mathbf{x}. E\mathbf{x} = \mathbf{n}\mathbf{y}$ revient à faire le calcul des modes élémentaires (Zanghellini et al., 2013) du système $\mathbf{A}\mathbf{y} \doteq 0$. Dans notre cas, les modes élémentaires sont les directions extrêmes du cône des solutions positives de $\mathbf{A}\mathbf{y} \doteq 0$. Ils sont bien connus, et des bibliothèques de géométrie (Avis and Jordan, 2018 ; Fukuda, 1993) permettent de faire leur calcul de façon efficace

même dans le pire des cas qui est exponentiel en temps de calcul . Nous allons voir qu'il est possible de faire ce calcul dans des temps raisonnables pour les modèles biologiques qui nous intéressent, cela reste vrai tant que les modèles ont une taille raisonnable, cela ne serait probablement pas faisable si on lançait le calcul sur un modèle représentant l'ensemble du métabolisme d'une espèce par exemple.

Par la suite, nous aurions besoin d'étendre ce résultat des systèmes d'équations linéaires à d'autres systèmes d'équations que nous appelons $h_{\mathbb{B}}$ -mixtes. Pour notre application, ils auront la forme :

$$\mathbf{A}\mathbf{y} \doteq 0 \wedge \bigwedge_{i=1}^n p_i \doteq 0$$

où p_1, \dots, p_n sont des polynômes à coefficients entiers positifs et sans terme constant. Nous montrerons que chaque formule $h_{\mathbb{B}}$ -mixte peut être réécrite dans une formule $h_{\mathbb{B}}$ -exacte, par réécriture de sa partie linéaire comme discuté précédemment.

Avec l'implémentation de ce nouvel algorithme de réécriture, il est à présent possible de calculer de façon exacte l'abstraction booléenne de systèmes d'équations $h_{\mathbb{B}}$ -mixtes en calculant l'ensemble des solutions abstraites de :

$$\exists \mathbf{x}. E\mathbf{x} = \mathbf{ny} \wedge \bigwedge_{i=1}^n p_i \doteq 0$$

en se basant sur la programmation par contraintes à domaine fini. La complexité du calcul de $\exists \mathbf{x}. E\mathbf{x} = \mathbf{ny}$ est bornée par le temps du calcul des modes élémentaires de \mathbf{A} , et donc simplement exponentiel dans la taille de \mathbf{A} dans le pire des cas. Le calcul des solutions abstraites de $\exists \mathbf{x}. E\mathbf{x} = \mathbf{ny} \wedge \bigwedge_{i=1}^n p_i \doteq 0$ peut être fait en temps exponentiel de la taille de E , et donc doublement exponentiel dans la taille de A . En pratique, le pire des cas est souvent évité par les techniques de propagation de contraintes.

1.4.2 Calcul exact d'abstractions de différence

Notre contribution majeure est un nouvel algorithme pour le calcul exact des abstractions de différences h_{Δ_3} et h_{Δ_6} de systèmes d'équations linéaires interprétés sur des paires de réels positifs. Toute solution abstraite retenue sera l'abstraction de différences de deux solutions réelles positives du système linéaire.

Cet algorithme repose sur le fait que les deux abstractions de différences h_{Δ} , où $\Delta \in \{\Delta_3, \Delta_6\}$, peuvent être définies à partir de l'abstraction booléenne $h_{\mathbb{B}}$ en logique du premier ordre. En utilisant la réécriture exacte en formules $h_{\mathbb{B}}$ -mixtes et des transformations de logique du premier-ordre, nous montrerons que l'ensemble

des abstractions de différences, avec h_{Δ} , d'un système linéaire peut être décrit par une formule de logique de premier-ordre avec des variables booléennes et des variables pour des différences en Δ .

Nous remarquerons que le cas de Δ_6 est considérablement plus difficile que le cas de Δ_3 . Puis nous montrons que notre solution peut-être étendue au cas avec des contraintes cinétiques. L'ajout de telles contraintes, par contre, demande un effort supplémentaire non trivial dans les deux cas.

La formule produite sur les domaines finis peut être résolue par la programmation par contraintes à domaine fini. La production de cette formule pour un système $\mathbf{Ax} = 0$ demande le calcul des modes élémentaires d'une autre matrice qui contient deux copies de \mathbf{A} . Ceci reste en temps exponentiel dans le pire des cas, mais peut être considérablement plus coûteux que le calcul des modes élémentaires de \mathbf{A} .

1.4.3 Outil de prédiction de changements et heuristiques

J'ai implémenté mon algorithme de calcul exact d'abstractions d'un système d'équations linéaires sous contraintes cinétiques pour le cas $\Delta = \Delta_6$, et je l'ai intégré dans l'outil de prédiction de Reaction-networks de l'équipe BioComputing. Ceci m'a permis de tester mon algorithme exact pour la prédiction de changements de réseau de réactions avec information cinétique partielle. Sur l'exemple de la prédiction de knock-out pour la surproduction de leucine, le temps de calcul se fait en quelques heures. Ceci montre que les techniques exactes sont réalistes pour des réseaux de taille modérée, disons jusqu'à 100 réactions, si on est prêt à accepter quelques heures de calcul sur une seule machine du type PC standard. Je n'ai pas encore essayé de paralléliser le calcul sur plusieurs machines.

Pour réduire le temps de l'algorithme exact, j'ai donc mis en place une nouvelle heuristique, qui consiste à calculer toutes les conséquences linéaires avec un nombre minimal de variables, que nous appelons les conséquences à support minimal. Je montrerai que le calcul de cette heuristique peut aussi être réduit à un calcul de modes élémentaires, mais d'une manière assez différente. Par conséquent, la prédiction de cette nouvelle méthode heuristique se calcule en quelques minutes dans le cas de la leucine.

Finalement, j'ai comparé la précision des heuristiques existantes, et la nouvelle heuristique, avec la méthode exacte. Il se trouve que la précision de la nouvelle heuristique de conséquences linéaires avec support minimal est comparable avec la qualité du calcul exact pour le cas du modèle de la production de Leucine, et meilleure que les heuristiques précédentes de l'outil de BioComputing.

1.5 Autres travaux liés

1.5.1 Prédiction de knock-out par l'optimisation à deux niveaux

Il existe des approches permettant de résoudre les problèmes de knock-out de réactions grâce à l'utilisation d'optimisation à deux niveaux (Burgard et al., 2003 ; Patil et al., 2005 ; Pharkya et al., 2004 ; Pharkya and Maranas, 2006). Le premier niveau utilise l'équilibre de flux pour assurer que l'organisme étudié optimise ses chances de survie (Bonarius et al., 1997 ; Covert et al., 2001 ; Price et al., 2004 ; Varma and Palsson, 1994) (croissance de sa biomasse (Burgard et al., 2003 ; Varma and Palsson, 1993), ou encore une production d'ATP, pour avoir une énergie optimale (Ramakrishna et al., 2001)). Le deuxième niveau, quant à lui, se charge de trouver la stratégie de knock-out de gène qui augmente le taux de la réaction ciblée. Ces deux problèmes d'optimisation sont résolus en utilisant des approches de programmation linéaire sur les entiers (Burgard et al., 2003 ; Kim and Reed, 2010) ou des algorithmes génétiques (Patil et al., 2005).

L'utilisation de méthodes d'optimisation à deux niveaux permet d'utiliser un domaine mathématique bien connu. Cependant, par cette approche nous perdons le lien entre les concentrations des espèces et les taux des réactions. Par exemple, on perd l'information que l'augmentation de concentration d'une espèce mène à l'augmentation du taux de la réaction qui la consomme. En particulier, les *feedbacks* négatifs ne peuvent donc pas être représentés.

De plus, savoir si le choix des fonctions objectives est approprié reste discutable (Ranganathan et al., 2010 ; Segre et al., 2002). En effet, par cette approche on ne représente pas l'adaptabilité de l'organisme, sous de nouvelles contraintes il peut modifier son métabolisme de façon improbable. L'approche (Segre et al., 2002) tente d'y remédier en utilisant un raisonnement basé sur des bornes minimale et maximale pour les taux des réactions. Dans (Ranganathan et al., 2010), il est plutôt proposé de minimiser les ajustements métaboliques. Cette approche se rapproche de la nôtre, elle compare la valeur des taux de réaction avant et après une modification. Les premières valeurs sont obtenues avant modification par une optimisation de la croissance de l'organisme. Puis les deuxièmes résultent des modifications qui sont optimisées pour s'éloigner le moins possible des valeurs originales.

En supplément, les approches d'optimisation ne permettent pas d'appliquer un raisonnement local sur une partie du métabolisme. Elles ont besoin de modèles assez grands, contenant au moins le cycle TCA ou la voie de la glycolyse, pour fonctionner. Mais avec la taille du modèle, le taux d'incertitude augmente. En addition, ces approches ne prévoient pas les effets de bord non souhaités après la mutation (Flórez et al., 2011).

1.5.2 Interprétation abstraite

En interprétation abstraite il est possible d'abstraire la sémantique de programmes avec des opérations arithmétiques (Cousot and Halbwachs, 1978 ; Granger, 1991). Mais l'interprétation abstraite est aussi utilisée dans le domaine de la biologie des systèmes afin d'analyser le comportement des systèmes sans tenir compte de leur information cinétique qui est souvent incomplète, voir manquante. En effet, il est possible de faire de la modélisation basée sur les contraintes (Covert et al., 2003 ; Palsson, 2000 ; Price et al., 2004). Au lieu de prédire exactement ce que le système métabolique fait, la méthode basée sur les contraintes fournit l'ensemble des comportements que peut avoir le système si on lui impose certaines contraintes. A partir du réseau un ensemble de contraintes concernant sa stœchiométrie et sa régulation sont inférées.

Il existe en effet d'autres méthodes d'analyse de réseaux de réactions chimiques basées sur l'interprétation abstraite. Nous avons comme point commun l'abstraction de domaines de valeurs. En effet, il est possible d'identifier dans notre approche, avec l'abstraction $h_{\Delta} : S \rightarrow \Delta$, la classique connexion de Galois (Broy and Steinbrüggen, 1999).

Les autres approches se concentrent sur la dynamique du système, elles considèrent des changements dans le nombre d'espèces au cours du temps (Danos et al., 2010). Cela repose sur l'idée que les vecteurs de concentration d'espèces fournissent des états et que la présence de réactions représente des états de transitions. Dans (Danos et al., 2010), l'espace des états permet de donner une sémantique d'étape qui est ensuite abstraite en une sémantique de collection. Dans notre cas, nous ne considérons qu'un seul état de changement, un changement entre deux états stationnaires, qui est directement interprété sur notre domaine abstrait et ne tient pas, de ce fait, compte d'une sémantique d'étape.

1.5.3 Réseaux booléens

L'approche (Thomas, 1973) représente la concentration de toute espèce moléculaire par une variable logique, c'est-à-dire à valeur booléenne. Dans cette approche une variable ne peut prendre que deux valeurs, 0 ou 1 (cela peut être généralisé à plusieurs valeurs entières). Ces valeurs sont interprétées comme l'absence ou la présence de la molécule en question. Des équations logiques décrivent l'évolution au cours du temps de la valeur des variables. Ainsi, la valeur de chaque variable dépendra de la valeur des variables au temps précédent.

La sémantique booléenne est donc purement qualitative, elle néglige totalement les informations cinétiques et fournit la sémantique la plus abstraite des réseaux de réactions (Calzone et al., 2006 ; Fages et al., 2004 ; Florez et al., 2011 ;

Gianchandani et al., 2006 ; Paulevé et al., 2020 ; Paulevé and Richard, 2012). On en distingue plusieurs l'approche *synchrone*, l'approche *pleinement asynchrone* et l'approche *asynchrone*. L'approche *synchrone* actualise les valeurs de toutes les variables simultanément à chaque itération. A l'inverse l'approche *pleinement asynchrone* actualise à chaque itération uniquement une variable choisie de façon non-déterministe. L'approche *synchrone* regroupe la grande partie des autres sémantiques de mise à jour introduites dans la littérature des réseaux booléens.

Ces approches abstraient un grand nombre d'information mais permettent tout de même de relever les comportements caractéristiques dans la dynamique des systèmes biologiques.

Les récents travaux de recherche de Paulevé et al. (2020) démontrent que ces sémantiques booléennes peuvent manquer certains comportements, que l'on observe dans le système quantitatif. De plus, le coût important de leur analyse limite leur application pour de grands modèles. Pour répondre à cette problématique, il propose une nouvelle sémantique, la *sémantique MP* ("Most Permissive semantics"). Elle permet de donner une nouvelle interprétation des réseaux booléens, efficace en temps de calcul donc applicable aux grands modèles, et qui elle capture l'ensemble des comportements réalisables du système pour tout raffinement quantitatif sans nécessité d'ajout d'information ou de paramètres. Cette sémantique prend en compte deux états supplémentaires, dit *dynamiques*: \nearrow et \searrow représentant une augmentation, et respectivement une diminution. Ils servent de transition entre les états, de 0 vers 1 pour \nearrow , et inversement pour \searrow . De plus, il compare cette approche avec les réseaux multivalués et les EDOs qui rajoutent des informations sur les vitesses et les durées des réactions, ou, sur les quantités des molécules. En faisant une binarisation des configurations multivaluées, il montre que la sémantique MP est une abstraction correcte des raffinements multivalués.

Une autre analyse des réseaux booléens intéressante est celle de Mandon et al. (2019), qui s'intéresse à la reprogrammation séquentielle de réseaux booléens. Il cherche à savoir quel contrôle doit être apporté au modèle pour arriver à un état d'intérêt, à partir d'un certain état de départ. Cela s'applique à l'étude de la reprogrammation cellulaire, c'est une grande avancée pour la médecine régénératrice.

1.5.4 Projet BestBioSurf

Le projet *BestBioSurf* (Théâtre et al., 2020) s'intéresse à l'optimisation de la production d'un biosurfactant en minimisant son coût de production et son écotoxicité. Les biosurfactants, produits par des micro-organismes, en plus d'être biodégradables et compatibles avec l'environnement, partagent de nombreuses pro-

priétés intéressantes comme: émulsifiant, détergeant, aide à la solubilisation, la lubrification et la dispersion de phase.

Le projet fait donc appel aux domaines de la bioinformatique, la biologie synthétique et l'ingénierie moléculaire pour trouver une forme de biosurfactant la plus efficace tout en étant peu toxique et très biodégradable. Et il demande également de prédire et d'effectuer de l'ingénierie génétique sur les micro-organismes dans le but d'augmenter leur productivité à moindre coût. C'est à ce niveau que nous intervenons, nous faisons des prédictions de knock-out de gènes à effectuer sur la bactérie *B. Subtilis* visant à augmenter sa production de surfactine, un biosurfactant d'intérêt.

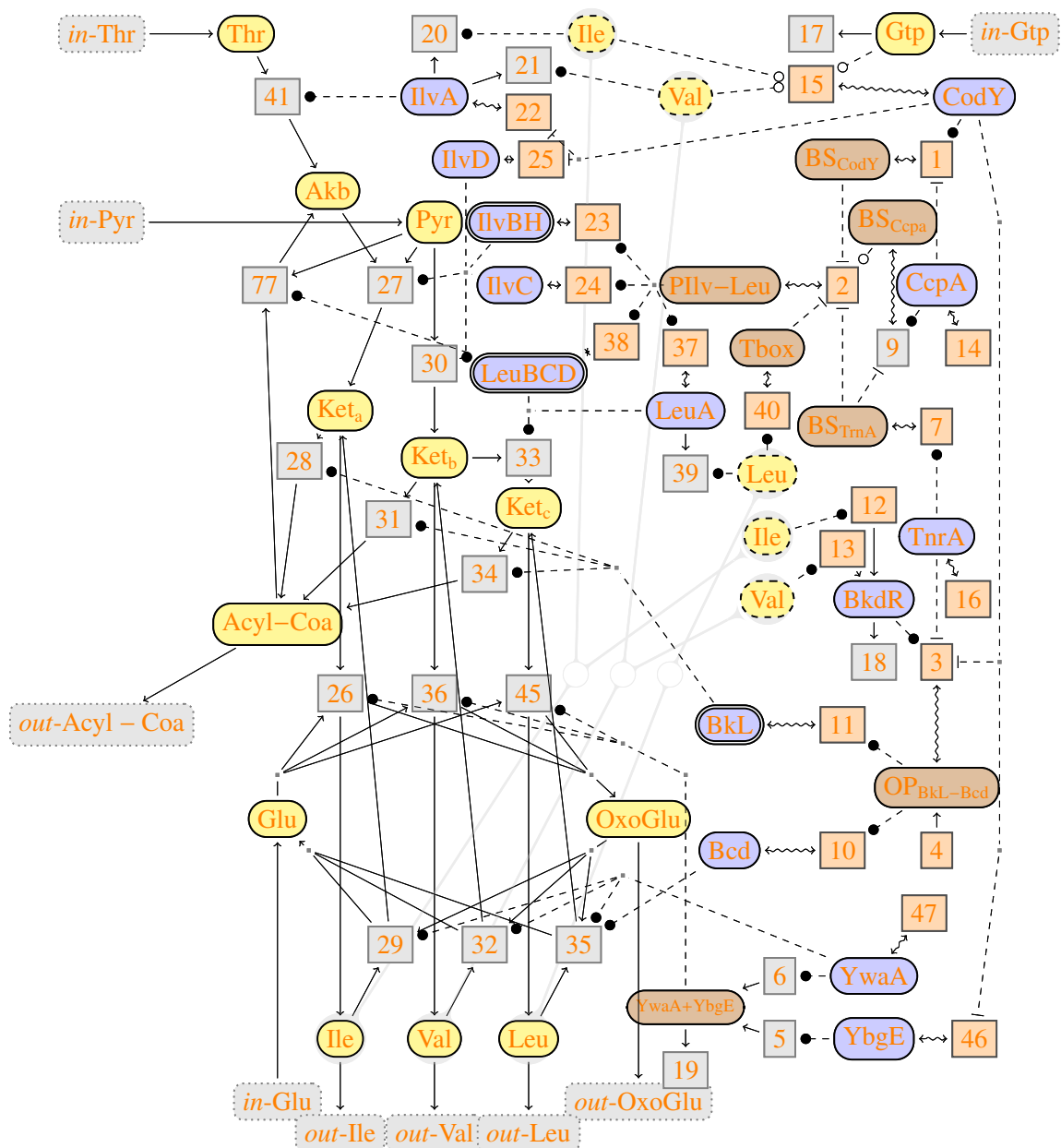


FIGURE 1.11 – Le graphe d’une partie d’un réseau métabolique de la bactérie *B. subtilis* avec sa régulation, produisant la Leucine.

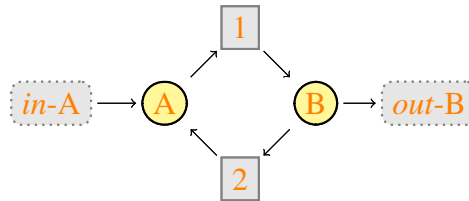


FIGURE 1.12 – Réseau métabolique avec un cycle simple.

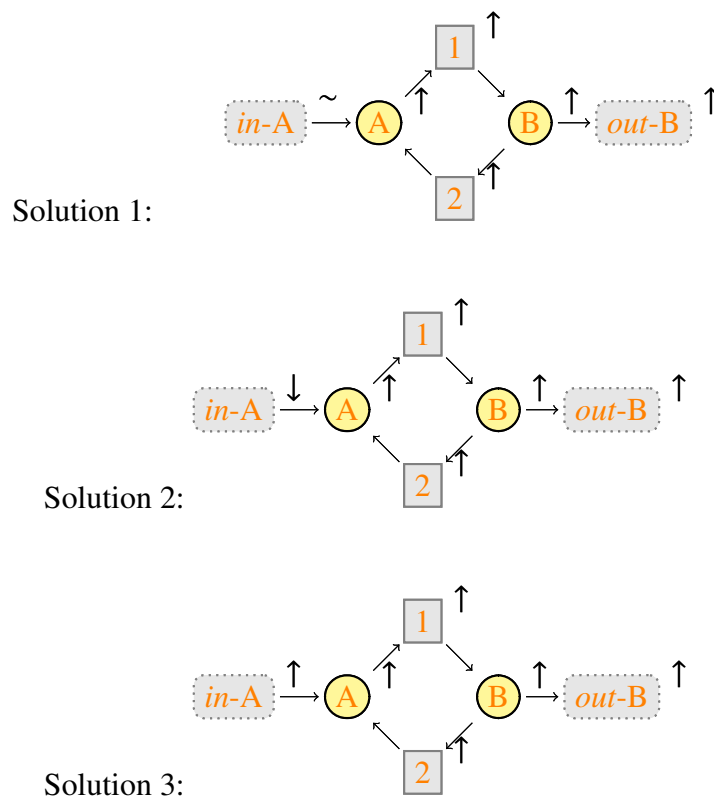


FIGURE 1.13 – Les trois solutions en Δ_6 du système linéaire ϕ compatibles avec $out-B \doteq \uparrow$.

Chapitre 2

Préliminaires

Nous rappelons les bases des outils de la logique du premier-ordre, que nous allons utiliser tout au long de ce manuscrit. Puis nous détaillons les systèmes d'équations linéaires et polynomiales qui nous intéressent, et qui sont des formules de la logique du premier-ordre. Et finalement, nous rappelons les notions de modes élémentaires pour des équations linéaires sous forme matricielle.

Sommaire

2.1	Ensembles et fonctions	43
2.2	Σ-algèbres	44
2.3	Σ-structures	45
2.4	Σ-abstractions	47
2.5	Ajout de constantes	48
2.6	Σ-expressions	48
2.7	Polynômes	49
2.8	Équations	50
2.9	Systèmes d'équations	51
2.10	Modes élémentaires	53
2.11	La logique du premier-ordre	55

2.1 Ensembles et fonctions

Commençons avec les notations usuelles sur les ensembles. Soit \mathbb{N} l'ensemble des nombres naturels et \mathbb{R}_+ l'ensemble des nombres positifs réels, tous deux com-

prennent 0.

Pour chaque ensemble A et $n \in \mathbb{N}$, l'ensemble des n -uplets d'éléments de A est noté A^n . La i -ème fonction de projection d'un élément de A sur un n -uplet, où $1 \leq i \leq n$ est la fonction $\pi_i : A^n \rightarrow A$ telle que $\pi_i(a_1, \dots, a_n) = a_i$ pour tout $a_1, \dots, a_n \in A$. Si A est fini, le nombre d'éléments de A est noté $|A|$.

Nous continuons avec la notion de fonction totale et partielle. Une fonction partielle est une relation $f \subseteq A \times B$ telle que pour tout $a \in A$ il existe au plus un $b \in B$ tel que $(a, b) \in f$. Dans ce cas, on écrit $f(a) = b$. Le domaine de la fonction partielle est $\text{dom}(f) = \{a \mid \exists b \in B. f(a) = b\}$ et son rang $\text{ran}(f) = \{b \mid \exists a \in A. f(a) = b\}$.

Étant données une fonction totale $f : A \rightarrow B$ et une fonction partielle $g : B \times C$ telles que $\text{ran}(f) \subseteq \text{dom}(g)$, on définit la fonction de composition comme la fonction totale $g \circ f : A \rightarrow C$ telle que $(g \circ f)(x) = g(f(x))$ pour tout $x \in A$. De plus, si $R \subseteq \{f : A \rightarrow B\}$ alors on définit :

$$g \circ R = \{g \circ f : A \rightarrow C \mid f \in R, \text{ran}(f) \subseteq \text{dom}(g)\}$$

A noter que les fonctions $f \in R$ ne respectant pas $\text{ran}(f) \subseteq \text{dom}(g)$ seront ignorées lors de la composition $g \circ R$. Cela vient du fait que l'on veut que toute fonction de $g \circ R$ soit totale même si g est partielle.

2.2 Σ -algèbres

Nous rappelons ensuite les notions de Σ -algèbre. Soit $\Sigma = \bigcup_{m \in \mathbb{N}} F^{(m)} \cup C$ une signature rangée. Les éléments de $f \in F^{(m)}$ sont appelés les symboles de fonction de Σ d'arité m et les éléments de $c \in C$ ses constantes.

Définition 1. Une Σ -algèbre $S = (\text{dom}(S), \cdot^S)$ consiste en un ensemble $\text{dom}(S)$ et une interprétation \cdot^S telle que $c^S \in \text{dom}(S)$ pour toute $c \in C$, et $f^S : \text{dom}(S)^m \rightarrow \text{dom}(S)$ pour tout $f \in F^{(m)}$ et $m \in \mathbb{N}$.

Nous réinterprétons ensuite les symboles de fonction de Σ d'arité m comme des symboles de relation d'arité $m+1$, ainsi nous pouvons réutiliser la même signature pour définir les Σ -structures.

La signature Σ la plus fréquente dans nos applications sera la signature arithmétique positive $\Sigma_{\text{pos-arith}} = F_{\text{pos-arith}}^{(2)} \cup F_{\text{pos-arith}}^{(1)} \cup C_{\text{pos-arith}}$ avec deux symboles de fonction binaire, un symbole de fonction monadique et deux constantes :

$$\begin{aligned} F_{\text{pos-arith}}^{(2)} &= \{+, *\} \\ F_{\text{pos-arith}}^{(1)} &= \{\text{inh}\} \\ C_{\text{pos-arith}} &= \{0, 1\} \end{aligned}$$

Pour toutes $\Sigma_{pos-arith}$ -algèbres S , les opérateurs $+^S$ et $*^S$ seront associatifs et commutatifs avec leur élément neutre respectif 0^S et 1^S . Et, si un inverse pour la multiplication est disponible pour toute valeur non nulle, et satisfait $1^S +^S r \neq 0^S$ pour tout $r \in dom(S)$, alors:

$$inh^S(r) = 1 /^{S \setminus \{0^S\}}(1 +^S r)$$

Exemple 2. L'ensemble des nombres réels positifs \mathbb{R}_+ peut être décrit par une $\Sigma_{pos-arith}$ -algèbre à domaine dans \mathbb{R}_+ , en interprétant $+$ comme l'addition sur les nombres réels positifs $+^{\mathbb{R}_+}$, $*$ comme la multiplication sur les nombres réels positifs $*^{\mathbb{R}_+}$, et en interprétant les constantes par elles-mêmes, à savoir, $0^{\mathbb{R}_+} = 0$ et $1^{\mathbb{R}_+} = 1$. Et pour tout $r \in dom(\mathbb{R}_+)$, inh est interprété comme ci-dessus, vue $r + 1 \neq 0$.

Par la suite pour toute Σ -algèbre, ou Σ -structure, le nom de celle-ci sera confondu avec son domaine dans le cas où ils sont identiques. Par exemple, nous allons délibérément confondre l'ensemble \mathbb{R}_+ avec la $\Sigma_{pos-arith}$ -algèbre $(\mathbb{R}_+, \cdot^{\mathbb{R}_+})$ dont le domaine $dom(\mathbb{R}_+)$ est égal à l'ensemble des réels positifs \mathbb{R}_+ .

Exemple 3. L'ensemble de booléens $\mathbb{B} = \{0, 1\} \subseteq \mathbb{R}_+$ peut être vu comme une $\Sigma_{pos-arith}$ -algèbre avec domaine \mathbb{B} en interprétant $+^{\mathbb{B}} = \vee^{\mathbb{B}}$ comme disjonction, $*^{\mathbb{B}} = \wedge^{\mathbb{B}}$ comme conjonction, et encore une fois les constantes par elles-mêmes $0^{\mathbb{B}} = 0$ et $1^{\mathbb{B}} = 1$. Et pour tout $b \in dom(\mathbb{B})$, inh est interprété par $inh^{\mathbb{B}}(b) = 1$.

2.3 Σ -structures

Nous rappelons la généralisation des Σ -algèbres à des Σ -structures, où les valeurs des opérateurs peuvent être interprétées d'une manière non-déterministe.

Définition 4. Une Σ -structure $\Delta = (dom(\Delta), \cdot^\Delta)$ consiste en un ensemble $dom(\Delta)$ et une interprétation \cdot^Δ telle que $c^\Delta \in dom(\Delta)$ pour tout $c \in C$ et $f^\Delta \subseteq dom(\Delta)^{m+1}$ pour tout $f \in F^{(m)}$ et $m \in \mathbb{N}$.

De cette façon, chaque Σ -algèbre est aussi une Σ -structure puisqu'une fonction d'arité n est une relation d'arité $n + 1$. A noter aussi que les symboles de $F^{(0)}$ sont interprétés comme des relations monadiques dans les Σ -structures, c'est-à-dire comme un sous-ensemble du domaine, contrairement aux constantes de C qui sont interprétées comme des éléments du domaine.

Exemple 5. Les nombres réels avec division forment une structure relationnelle mais pas une algèbre, étant donné que la division par zéro n'est pas définie. Plus

formellement, nous considérons la signature arithmétique $\Sigma_{arith} = F_{arith}^{(2)} \cup F_{arith}^{(1)} \cup C_{arith}$ où :

$$\begin{aligned} F_{arith}^{(2)} &= F_{pos-arith}^{(2)} \cup \{-, /\} \\ F_{arith}^{(1)} &= F_{pos-arith}^{(1)} \\ C_{arith} &= C_{pos-arith} \end{aligned}$$

L'ensemble des nombres réels \mathbb{R} peut être vu comme une Σ_{arith} -structure à domaine dans \mathbb{R} , en interprétant le symbole $/$ comme division, tel que pour tout $r_1, r_2 \in \mathbb{R}$:

$$r_1 /^{\mathbb{R}} r_2 = \begin{cases} \emptyset & \text{si } r_2 = 0 \\ \{r \in \mathbb{R} \mid r_2 *^{\mathbb{R}} r = r_1\} & \text{si } r_2 \neq 0 \end{cases}$$

Le résultat $r_1 /^{\mathbb{R}} 0 = \emptyset$ modélise le fait que la division par zéro n'est pas définie. Si par contre $r_2 \neq 0$, alors $r_1 /^{\mathbb{R}} r_2$ est toujours un singleton. L'interprétation de inh pour tout $r \in \mathbb{R}$ est :

$$inh^{\mathbb{R}}(r) = 1 /^{\mathbb{R}} (1 +^{\mathbb{R}} r)$$

Cet opérateur est donc redondant pour \mathbb{R} , mais ne l'était pas pour \mathbb{R}_+ . Nous le gardons tout de même dans la signature pour des raisons de cohérence. L'interprétation des autres opérateurs et constantes est définie comme à l'usuel.

Exemple 6. Les fonctions réelles du type $\mathbb{R}_+ \rightarrow \mathbb{R}$ avec division forment une structure relationnelle, qui elle non plus n'est pas une algèbre. Plus formellement, nous considérons la signature arithmétique $\Sigma_{deriv} = F_{deriv}^{(2)} \cup F_{deriv}^{(1)} \cup C_{deriv}$ où :

$$\begin{aligned} F_{deriv}^{(2)} &= F_{arith}^{(2)} \\ F_{deriv}^{(1)} &= \{\circ\} \cup F_{arith}^{(1)} \\ C_{deriv} &= C_{pos-arith} \end{aligned}$$

L'ensemble des fonctions réelles de type $\mathbb{R}_+ \rightarrow \mathbb{R}$ peut être vu comme une Σ_{deriv} -structure avec domaine $\{f \mid f : \mathbb{R}_+ \rightarrow \mathbb{R}\}$ en généralisant l'interprétation des opérateurs en $\{\odot^{\mathbb{R}} \mid \odot \in F_{pos-arith}^{(2)}\}$ sur des fonctions dans les réels, tel que pour tout $f_1, f_2 : \mathbb{R}_+ \rightarrow \mathbb{R}$:

$$f_1 \odot^{\mathbb{R}} f_2 = \{f \in \mathbb{R}_+ \rightarrow \mathbb{R} \mid \forall r \in \mathbb{R}_+. f(r) \in f_1(r) \odot^{\mathbb{R}} f_2(r)\}$$

La généralisation de l'opérateur monadique inh est :

$$inh^{\mathbb{R}}(f) = \{f \in \mathbb{R}_+ \rightarrow \mathbb{R} \mid \forall r \in \mathbb{R}_+. f(r) \in inh^{\mathbb{R}}(r)\}$$

Dans les deux cas ci-dessus, la division donnera un résultat non vide si on divise par une fonction non nulle partout.

L'interprétation de l'opérateur monadique de dérivation \circ dans cette structure envoie toute fonction dérivable $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ vers le singleton $\{f\}$ contenant sa dérivée, et toute autre fonction vers l'ensemble vide.

2.4 Σ -abstractions

Nous introduisons maintenant notre concept de Σ -abstraction pour des signatures g n rales. Elle est bas e sur la notation standard des homomorphismes entre Σ -structures. Le fait de pouvoir g n raliser le th or me de [John et al. \(2013a\)](#)   cette notion d'abstraction est une contribution de ce projet de th se.

D finition 7. *Un homomorphisme entre deux Σ -structures S et Δ est une fonction $h : \text{dom}(S) \rightarrow \text{dom}(\Delta)$ telle que pour $c \in C$, $n \in \mathbb{N}$, $f \in F^{(n)}$, et $s_1, \dots, s_{n+1} \in \text{dom}(S)$:*

1. $h(c^S) = c^\Delta$, et
2. si $(s_1, \dots, s_{n+1}) \in f^S$ alors $(h(s_1), \dots, h(s_{n+1})) \in f^\Delta$.

Si nous consid rons des relations d'arit  $n + 1$ comme des fonctions multivalu es d'arit  n , la seconde condition peut ˆtre r ecrite de fa on similaire comme :

$$h(f^S(s_1, \dots, s_n)) \subseteq f^\Delta(h(s_1), \dots, h(s_n))$$

Pour les Σ -alg bres, cette condition est  quivalente   :

$$h(f^S(s_1, \dots, s_n)) = f^\Delta(h(s_1), \dots, h(s_n))$$

D finition 8. *Une Σ -abstraction est un homomorphisme entre Σ -structures S et Δ tel que $\text{dom}(\Delta) \subseteq \text{dom}(S)$.*

Par exemple, nous pouvons abstraire les nombres positifs r els en bool ens en d finissant une fonction $h_{\mathbb{B}} : \mathbb{R}_+ \rightarrow \mathbb{B}$ telle que $h_{\mathbb{B}}(0) = 0$ et $h_{\mathbb{B}}(r) = 1$ pour tout $r \in \mathbb{R}_+ \setminus \{0\}$.

Lemme 9. *La fonction $h_{\mathbb{B}} : \mathbb{R}_+ \rightarrow \mathbb{B}$ est une $\Sigma_{\text{pos-arith}}$ -abstraction entre $\Sigma_{\text{pos-arith}}$ -alg bres.*

D monstration. Par d finition, nous avons $\mathbb{B} \subseteq \mathbb{R}_+$, comme nous identifions les bool ens avec 0 et 1. Il reste   montrer, que $h_{\mathbb{B}}$ est un homomorphisme. Pour tout $r, r' \in \mathbb{R}_+$ nous avons:

$$\begin{aligned} h_{\mathbb{B}}(r +^{\mathbb{R}_+} r') = 1 &\Leftrightarrow r +^{\mathbb{R}_+} r' \neq 0 &&\Leftrightarrow r \neq 0 \vee r' \neq 0 \\ &\Leftrightarrow h_{\mathbb{B}}(r) = 1 \vee h_{\mathbb{B}}(r') = 1 \\ h_{\mathbb{B}}(r *^{\mathbb{R}_+} r') = 1 &\Leftrightarrow r *^{\mathbb{R}_+} r' \neq 0 &&\Leftrightarrow r \neq 0 \wedge r' \neq 0 \\ &\Leftrightarrow h_{\mathbb{B}}(r) = 1 \wedge h_{\mathbb{B}}(r') = 1 \end{aligned}$$

Ainsi $h_{\mathbb{B}}(r +^{\mathbb{R}_+} r') = h_{\mathbb{B}}(r) +^{\mathbb{B}} h_{\mathbb{B}}(r')$ et $h_{\mathbb{B}}(r *^{\mathbb{R}_+} r') = h_{\mathbb{B}}(r) *^{\mathbb{B}} h_{\mathbb{B}}(r')$. En ce qui concerne l'op rateur inh , pour tout $r \in \mathbb{R}_+$ nous notons que

$$inh^{\mathbb{R}_+}(r) = 1 /_{\mathbb{R}_+ \setminus \{0\}}(1 +^{\mathbb{R}_+} r) > 0$$

et donc $h_{\mathbb{B}}(inh^{\mathbb{R}_+}(r)) = 1 = inh^{\mathbb{B}}(r)$. Et pour les deux constantes $c \in C$, nous avons $h_{\mathbb{B}}(c^{\mathbb{R}_+}) = h_{\mathbb{B}}(c) = c = c^{\mathbb{B}}$. \square

Dans la plupart des Σ -abstractions que nous allons étudier, nous allons abstraire des Σ -algèbres dans des Σ -structures, qui ne sont pas des Σ -algèbres.

2.5 Ajout de constantes

Il est souvent utile d'ajouter les éléments d'une Σ -structure A aux constantes. Pour ceci, nous définissons la signature étendue :

$$\Sigma[\text{dom}(A)] = \Sigma \uplus \text{dom}(A)$$

Les arités de tous les symboles de Σ sont maintenues dans $\Sigma[\text{dom}(A)]$. La Σ -structure A devient aussi une $\Sigma[\text{dom}(A)]$ -structure, si nous interprétons les nouvelles constantes $a \in \text{dom}(A)$ elles-mêmes :

$$a^A = a$$

2.6 Σ -expressions

Soit $\Sigma = \bigcup_{m \in \mathbb{N}} F^{(m)} \cup C$ une signature rangée et \mathcal{V} un ensemble de variables. Les Σ -expressions sont tous les termes que l'on peut construire avec les opérateurs $f \in F^{(m)}$ où $m \in \mathbb{N}$, les constantes $c \in C$, et les variables $x \in \mathcal{V}$:

$$e, e_1, \dots, e_m \in \mathcal{E}_\Sigma ::= x \mid c \mid f(e_1, \dots, e_m)$$

Les $\Sigma_{\text{pos-arith}}$ -expressions contiennent des polynômes positifs avec coefficients naturels, tel que :

$$(1 + 1 + 1) * x * x + (1 + 1) * x * y + 1 + 1 + 1 + 1$$

Les Σ_{arith} -expressions, elles, peuvent contenir en supplément des fractions de polynômes, et ce sans restriction de positivité.

Attention voir cohérence avec les exemples suivants

Elles permettent, par exemple, d'écrire des expressions telles que la cinétique de *Michaelis et Menten* qui décrit la vitesse initiale stationnaire d'une réaction enzymatique :

$$\frac{V_{\max} * x}{K_m + x} \quad \text{où } V_{\max} = \frac{981}{1000} \text{ et } K_m = \frac{45}{10}$$

A savoir que toute constante rationnelle peut être obtenue en $\mathcal{E}_{\Sigma_{\text{arith}}}$ en faisant des fractions de somme n'utilisant que les constantes 1 et 0, qui elles sont définies dans la signature Σ_{arith} . L'ensemble des variables libres $\text{fv}(e) \subseteq \mathcal{V}$ d'une expression est l'ensemble de toutes les variables qui apparaissent en e .

$$\begin{aligned} \llbracket c \rrbracket^{\alpha, S} &= \{c^S\} & \llbracket x \rrbracket^{\alpha, S} &= \{\alpha(x)\} \\ \llbracket f(e_1, \dots, e_n) \rrbracket^{\alpha, S} &= \cup \{f^S(s_1, \dots, s_n) \mid s_i \in \llbracket e_i \rrbracket^{\alpha, S} \text{ for all } 1 \leq i \leq n\} \end{aligned}$$

FIGURE 2.1 – Interprétation ensembliste des Σ -expressions. Étant donné une Σ -structure S et un assignement de variables $\alpha : V \rightarrow \text{dom}(S)$ pour toutes les variables libres nous définissons l'interprétation $\llbracket e \rrbracket^{\alpha, S} \subseteq \text{dom}(S)$.

La sémantique des Σ -expressions $e \in \mathcal{E}_\Sigma$ peut être définie pour toute Σ -structure S . Un assignement de variables dans une Σ -structure S est une fonction partielle $\alpha : V \rightarrow \text{dom}(S)$ pour un sous-ensemble $V \subseteq \mathcal{V}$. Soit S une Σ -structure et α un assignement de variables dans S . Chaque Σ -expression e avec $\text{fv}(e) \subseteq V$ peut être interprétée comme un élément

$$\llbracket e \rrbracket^{\alpha, S} \subseteq \text{dom}(S)$$

suivant FIGURE 2.1. Les opérations et constantes y sont interprétées et appliquées d'une manière ascendante suivant la structure du terme. Comme les opérations des structures peuvent être non déterministes, un ensemble est retourné dans tous les cas.

2.7 Polynômes

Des polynômes avec des coefficients rationnels sont des Σ_{arith} -expressions. Dans la plupart des cas, nous allons seulement considérer des polynômes à coefficients entiers ou positifs. Ces derniers sont des $\Sigma_{pos-arith}$ -expressions, excluant ainsi la soustraction et la division.

Pour pouvoir écrire des polynômes avec des coefficients entiers plus directement, nous introduisons les notations suivantes. Pour tout naturel n et expressions arithmétiques $e, e_1, \dots, e_n \in \mathcal{E}_{\Sigma_{arith}}$, nous définissons l'expression arithmétique $\prod_{i=1}^n e_i = e_1 * \dots * e_n$, qui est égale à 1 si $n = 0$, et $\sum_{i=1}^n e_i = e_1 + \dots + e_n$, qui vaut 0 si $n = 0$. De plus $e^n =_{\text{def}} \prod_{i=1}^n e$ et $ne =_{\text{def}} \sum_{i=1}^n e$. Pour chaque expression $e \in \mathcal{E}_{\Sigma_{arith}}$ nous définissons $-e$ par $0 - e$ et pour chaque entier négatif $z \in \mathbb{Z} \setminus \mathbb{N}$, nous définissons $ze =_{\text{def}} -(ne)$ où $n = -z$.

Définition 10. *Un polynôme à coefficients entiers est une Σ -expression de la forme suivante :*

$$\sum_{j=1}^l z_j \prod_{k=1}^{i_j} x_{j,k}^{m_{j,k}}$$

où l et i_j sont des naturels, z_j des entiers non nuls, $m_{j,k}$ des naturels non nuls, et $x_{1,1}, \dots, x_{l,i_l}$ des variables.

Les z_j sont les *coefficients du polynôme* et ces *monômes* sont les produits $z_j \prod_{k=1}^{i_j} x_{j,k}^{m_{j,k}}$. Le polynôme est dit à *coefficients naturels* si tous ses coefficients sont naturels, autrement dit $z_j \in \mathbb{N}$. Naturellement, tout polynôme à coefficients naturels est en $\mathcal{E}_{\Sigma_{pos-arith}}$. On dit d'un polynôme qu'il n'a pas de *terme constant* si aucun de ses monômes est égal à z_j , c'est-à-dire, $i_j \neq 0$ pour tout $1 \leq j \leq l$.

Un polynôme est dit *linéaire* si tous ses monômes sont des variables, c'est-à-dire, $i_j = 1$ et $m^{j,1} = \dots = m^{j,i_j} = 1$ pour tout $1 \leq j \leq l$. A noter que tout polynôme linéaire a la forme $\sum_{j=1}^l z_j x_{j,1}$, où l et tout $z_j \neq 0$ sont des entiers, et tout $x_{j,1}$, sont des variables. En particulier, les polynômes linéaires n'ont pas de terme constant. De plus, notez que la constante 0 est égale au polynôme linéaire avec $l = 0$.

2.8 Équations

Une Σ -équation est une paire $(e_1, e_2) \in \mathcal{E}_{\Sigma}^2$ que nous écrivons plus simplement $e_1 \doteq e_2$. Pour tout Σ -structure S et assignement de variables $\alpha : V \rightarrow \text{dom}(S)$ avec $V \supseteq \text{fv}(e) \cup \text{fv}(e')$ nous définissons une valeur booléenne $\llbracket e \doteq e' \rrbracket^{\alpha, S} \in \mathbb{B}$ telle que:

$$\llbracket e \doteq e' \rrbracket^{\alpha, S} = \begin{cases} 1 & \text{si } \llbracket e \rrbracket^{\alpha, S} \cap \llbracket e' \rrbracket^{\alpha, S} \neq \emptyset \\ 0 & \text{sinon} \end{cases}$$

La sémantique des Σ -équations $e \doteq e'$ doit prendre en compte le non déterminisme des Σ -structures. Lorsqu'elles sont interprétées sur une Σ -algèbre l'unique valeur de e et de e' dans S doivent être égales. Pour le cas des Σ -structures, chaque expression e dénote un sous-ensemble de la Σ -structure possible, et non juste un élément unique. Dans ce cas, non déterministe, l'égalité est interprétée comme l'intersection non vide, c'est-à-dire que $e \doteq e'$ est vraie pour une Σ -structure S si les interprétations de e et e' contiennent au moins un élément commun.

Définition 11. Une $\Sigma_{pos-arith}$ -équation est appelée positive si elle a la forme $e \doteq 0$ et quasi-positive si elle a la forme $e \doteq ny$, où $n \in \mathbb{N}$, $y \in \mathcal{V}$, et $e \in \mathcal{E}_{\Sigma_{pos-arith}}$.

Cette définition a du sens comme toutes les constantes des $\Sigma_{pos-arith}$ -expressions sont positives, et que tous les opérateurs des $\Sigma_{pos-arith}$ -expressions préservent la positivité quand ils sont interprétés sur \mathbb{R}_+ .

A noter aussi qu'une $\Sigma_{pos-arith}$ -équation positive est quasi-positive, puisque la constante 0 est égale au polynôme $0y$.

Définition 12.

- Une équation polynomiale (à coefficients entiers) est une Σ_{arith} -équation $p \doteq p'$ entre polynômes (à coefficients entiers).
- Une équation linéaire (homogène) est une équation polynomiale avec des polynômes linéaires, et donc sans terme constant.

2.9 Systèmes d'équations

Un système de Σ -équations est une formule conjonctive de la forme $\bigwedge_{i=1}^n e_i \doteq e'_i$. Nous appelons *quasi-positif* respectivement *positif* un système de $\Sigma_{pos-arith}$ -équations, si toutes ses équations le sont.

Une solution d'un système de Σ -équations sur une Σ -structure S est un assignement de variables α tel que $\llbracket e_i \rrbracket^{\alpha, S} = \llbracket e'_i \rrbracket^{\alpha, S}$ pour tout $1 \leq i \leq n$.

Nous rappelons, ci-dessous, que tout système de $\Sigma_{pos-arith}$ -équations linéaires avec coefficients naturels positifs peut être exprimé comme une *équation linéaire matricielle d'entiers homogènes*, et vice versa. Des telles équations ont la forme :

$$\mathbf{A}\mathbf{y} \doteq 0$$

où $\mathbf{A} \in \mathbb{Z}^{n \times m}$ est une matrice d'entiers, m, n des naturels, et $\mathbf{y} \in \mathcal{V}^m$ un vecteur de variables. Soit $\mathbf{A}_{i,j} \in \mathbb{Z}$ la valeur de la matrice se trouvant à la ligne i et la colonne j , et soit $\mathbf{y}_j = \pi_j(\mathbf{y})$ le j -ième élément de \mathbf{y} . Avec ces notations, on peut donner la sémantique d'une équation matricielle en imposant son équivalence au système de Σ_{arith} -équations linéaires suivant :

$$\bigwedge_{i=1}^m \sum_{j=1}^n \mathbf{A}_{i,j} \mathbf{y}_j \doteq 0$$

Et réciproquement, chaque système de Σ_{arith} -équations linéaires peut être réécrit comme une équation linéaire matricielle d'entiers, en regroupant les multiples coefficients pour une même variable du côté gauche de l'équation dans le cas où les coefficients sont entiers et non rationnels. Si une Σ_{arith} -équation linéaire possède des coefficients rationnels, il faudra multiplier celle-ci par un facteur égal au produit des dénominateurs de chaque monôme de l'équation.

Ensuite, nous pouvons réécrire chaque système de Σ_{arith} -équations linéaires comme un système de $\Sigma_{pos-arith}$ -équations. Pour faire cela, soit $\mathbf{A}^+, \mathbf{A}^- \in \mathbb{N}^{n,m}$ les matrices positives, tel que pour tout $1 \leq i \leq n$ et $1 \leq j \leq m$:

$$\begin{aligned} \mathbf{A}_{i,j}^+ &= \max\{0, \mathbf{A}_{i,j}\} \\ \mathbf{A}_{i,j}^- &= -\min\{0, \mathbf{A}_{i,j}\} \end{aligned}$$

Par construction, nous avons $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$ dans la structure $\mathbb{R}^{n,m}$. Ceci montre, que $\mathbf{A}\mathbf{y} \doteq 0$ est \mathbb{R} -équivalent à $\mathbf{A}^+\mathbf{y} \doteq \mathbf{A}^-\mathbf{y}$ et donc \mathbb{R} -équivalent à la $\Sigma_{pos-arith}$ équation :

$$\bigwedge_{i=1}^m \sum_{j=1}^n \mathbf{A}_{i,j}^+ \mathbf{y}_j \doteq \sum_{j=1}^n \mathbf{A}_{i,j}^- \mathbf{y}_j$$

De cette manière, chaque système de Σ_{arith} -équations peut être normalisé en un système de $\Sigma_{pos-arith}$ -équations \mathbb{R} -équivalent.

Exemple 13. Soit un système de $\Sigma_{pos-arith}$ -équations linéaires à coefficients naturels :

$$3x \doteq 0 \wedge 2x \doteq 5y$$

Celui-ci est la normalisation d'un système de Σ_{arith} -équations linéaires à coefficients entiers suivant :

$$3x \doteq 0 \wedge 2x - 5y \doteq 0$$

Et il donnera l'équation linéaire matricielle d'entiers ci-dessous

$$\begin{pmatrix} 3 & 0 \\ 2 & -5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \doteq 0$$

Le système normalisé ci-dessus est quasi-positif, mais pas positif puisque $5y$ apparaît sur le côté droit d'une équation. Plus généralement, le système d'équations linéaires d'une équation linéaire matricielle d'entiers $\mathbf{A}\mathbf{y} \doteq 0$ est positif, si et seulement si, tous les entiers dans \mathbf{A} sont positifs, et quasi-positif, si chaque ligne de \mathbf{A} contient au plus un entier négatif. De plus, ce système d'équations linéaires est triangulaire comme expliqué ci-dessous, mais pas fortement triangulaire.

Définition 14. Nous appelons triangulaire un système de $\Sigma_{pos-arith}$ -équations quasi-positif s'il a la forme $\bigwedge_{i=1}^n e_i \doteq n_i y_i$ telle que les variables y_l sont l -fraîches pour tout $1 \leq l \leq n$, c'est-à-dire:

$$y_l \notin \cup_{i=l}^{l-1} fv(e_i) \cup fv(e'_i) \text{ et si } n_l \neq 0 \text{ alors } y_l \notin fv(e_l).$$

Le système est dit fortement triangulaire si il est triangulaire et satisfait $n_l \neq 0$ pour tout $1 \leq l \leq n$.

Considérons une équation linéaire matricielle d'entiers $\mathbf{A}\mathbf{y} \doteq 0$. Si \mathbf{A} est positive et triangulaire, alors le système d'équations linéaires correspondant est positif et triangulaire aussi. Pour être quasi-positif et fortement triangulaire, les entiers sous la diagonale de \mathbf{A} doivent être négatifs, ceux sur la diagonale doivent être strictement négatifs, et ceux au-dessus de la diagonale doivent être positifs.

2.10 Modes élémentaires

Nous allons introduire le concept bien connu des modes élémentaires d'un réseau de réactions, qui sont l'ensemble des chemins minimaux et uniques réalisables dans le graphe qui garantissent l'état stable. On recherche pour cela toutes les combinaisons linéaires des réactions du réseau conduisant à l'état stable.

Nous verrons ensuite que les modes élémentaires (Gagneur and Klamt, 2004 ; Zanghellini et al., 2013) peuvent être utilisés pour obtenir un système d'équations quasi-positif et fortement triangulaire à partir d'une équation linéaire matricielle d'entiers.

Lorsque l'on suppose qu'un réseau métabolique se trouve dans un état stable, la concentration de chacun de ses métabolites internes est constante. Soit $\mathbf{A} \in \mathbb{Z}^{m \times n}$ sa matrice de stœchiométrie, on obtient le système d'équations :

$$\mathbf{A}\mathbf{y} = 0 \quad (2.1)$$

où $\mathbf{y} \in \mathcal{V}^n$ représente le vecteur de flux. La valeur réelle assignée à chacune de ses variables donne le taux de chaque réaction dans un flux. L'ensemble des solutions pour les flux qui satisfont cette équation matricielle est appelé le *nullspace* de \mathbf{A} . Dans notre cas, toutes les réactions sont irréversibles. Ainsi chaque variable du vecteur \mathbf{y} ont la contrainte d'être positive, on a donc :

$$\mathbf{y} \geq 0 \quad (2.2)$$

Un vecteur de flux répondant à ces contraintes est appelé un mode.

Définition 15. On définit le support du vecteur \mathbf{y} par $\text{supp}(\mathbf{y}) = \{i \mid y_i \neq 0\}$. On dit qu'un vecteur $\mathbf{y} \in \mathcal{V}^n$, non nul, est à support minimal s'il n'existe pas d'autre vecteur $\mathbf{y}' \in \mathcal{V}^n$ non nul tel que $\text{supp}(\mathbf{y}') \subsetneq \text{supp}(\mathbf{y})$.

On dit qu'un mode est un *mode élémentaire* s'il est à support minimal (Schuster and Hilgetag, 1994). Contrairement à l'ensemble des modes, l'ensemble des modes élémentaires d'un réseau est fini modulo une normalisation. N'importe quelle solution du *nullspace* (c'est-à-dire n'importe quel mode) peut se réécrire comme une combinaison linéaire non négative des modes élémentaires (Klamt and Stelling, 2003).

De cette façon les modes élémentaires sont une représentation finie de l'ensemble des solutions.

Il est possible, comme nous allons le voir avec la *double description method* de (Komei and Alain, 1996 ; T.S. et al., 1953), d'exprimer ce problème de façon plus géométrique. L'ensemble des contraintes homogènes 2.1 et 2.2 sur les vecteurs

de flux forme un cône polyédrale convexe CF , appelé un cône de flux dans le domaine de l'analyse de réseau métabolique :

$$CF = \{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{y} = 0 \wedge \mathbf{y} \geq 0\} \quad (2.3)$$

C'est un cas particulier des cônes polyédrales qui eux pour une matrice $N \in \mathbb{R}^{k \times n}$ se définissent plus généralement comme :

$$C = \{\mathbf{x} \in \mathbb{R}^n \mid N\mathbf{x} \geq 0\} \quad (2.4)$$

Les cônes de flux peuvent aussi s'exprimer sous la forme générale en prenant :

$$N = \begin{pmatrix} A \\ -A \\ I \end{pmatrix}$$

Définition 16. Les modes élémentaires d'un réseau de réactions, noté \mathbb{R}_+ -EFMs, ayant $\mathbf{A} \in \mathbb{Z}^{m \times n}$ pour matrice de stœchiométrie, représentent les vecteurs à support minimal non nuls du cône de flux $\{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{y} = 0 \wedge \mathbf{y} \geq 0\}$. Un vecteur $e \in FC$ non nul est un \mathbb{R}_+ -EFM si n'existe pas de vecteur non nul $y \in FC$ tel que :

$$\text{supp}(y) \subsetneq \text{supp}(e)$$

D'après cette définition, il est clair que pour un \mathbb{R}_+ -EFM e donné, n'importe quelle multiplication scalaire $\lambda \cdot e$, avec $\lambda \in \mathbb{N}$ donne aussi un \mathbb{R}_+ -EFM. Nous fixons donc un ensemble unique \mathbb{R}_+ -EFMs sans considérer les doublons engendrés par les scalaires λ . Une grande propriété des modes élémentaires est que tout élément d'un cône de flux FC peut être écrit comme une combinaison linéaire non négative de ces modes élémentaires. On peut ainsi donner la Proposition qui suit.

Proposition 17. Pour tout système d'équations linéaires matricielles à coefficients entiers, noté $\mathbf{A}\mathbf{y} = 0$, il existe une matrice E de naturels, un vecteur de naturels \mathbf{n} , et un vecteur de variables fraîches \mathbf{x} , tel que $\mathbf{A}\mathbf{y} = 0$ est \mathbb{R}_+ -équivalent à $\exists \mathbf{x}. E\mathbf{x} = \mathbf{n}\mathbf{y}$.

Démonstration. Nous allons utiliser dans cette preuve la méthode de la description double de Motzkin (Gagneur and Klamt, 2004 ; Komei and Alain, 1996 ; T.S. et al., 1953) qui décrit un cône de façon équivalente par sa H-représentation, correspondant à l'énumération de semi-espaces, et sa V-représentation qui elle repose sur l'énumération de vecteur générateur et de sommets (dans notre cas, le cône étant pointu, le seul sommet est l'origine). Géométriquement, l'espace de solutions de $\mathbf{A}\mathbf{y} = 0$ sur les réels est un sous-espace linéaire de $\mathbb{R}^{V(y)}$. En se restreignant aux réels positifs, comme nous le faisons, ce sous-espace linéaire doit être intersecté

avec le cône positif $\mathbb{R}_+^{\mathcal{V}(\mathbf{y})}$. Ainsi, $\text{sol}^{\mathbb{R}_+}(\mathbf{A}\mathbf{y} \doteq \mathbf{0})$ est le cône rationnel obtenu par une intersection finie de plusieurs semi-espaces : sa H-représentation est définie par les inéquations $\mathbf{A}\mathbf{y} \leq \mathbf{0} \wedge \mathbf{A}\mathbf{y} \geq \mathbf{0} \wedge \mathbf{y} \geq \mathbf{0}$, on retrouve l'équation 2.3. Les modes élémentaires (\mathbb{R}_+ -EFMs) de $\mathbf{A}\mathbf{y} \doteq \mathbf{0}$ permettent sa V-représentation. Étant donné que nous considérons l'ensemble \mathbb{R}_+ -EFMs normalisé, il y a un nombre fini de vecteur à support minimal. De plus, puisque le cône est rationnel et $\mathbf{A}\mathbf{y} \doteq \mathbf{0}$ est homogène, les modes élémentaires peuvent être normalisés de telle sorte que la V-représentation ne contienne que des coefficients entiers. Les modes élémentaires normalisés seront des vecteurs de naturels dans $\mathbb{N}^{\mathcal{V}(\mathbf{y})}$. Soit e_1, \dots, e_n l'ensemble des modes élémentaires normalisés dans un ordre total arbitraire et E la matrice avec les colonnes e_1, \dots, e_n . Par construction, les variables de \mathbf{y} sont distinctes l'une de l'autre. Selon la V-représentation normalisée du système, chaque point du cône est une combinaison positive linéaire des modes élémentaires $\exists \mathbf{x}. E\mathbf{x} \doteq \mathbf{ny}$, où \mathbf{n} contient les facteurs de normalisation. \square

Théorème 18 (Modes élémentaires). *Pour n'importe quel système d'équations linéaires ϕ à coefficients entiers, il est possible de calculer en temps au plus exponentiel une formule \mathbb{R}_+ -équivalente $\exists \mathbf{x}.\phi'$ telle que ϕ' est un système d'équations linéaires à coefficients naturels quasi-positif et fortement triangulaire et \mathbf{x} est la séquence de variables sur le côté droit des équations.*

Démonstration. Le système linéaire $E\mathbf{x} \doteq \mathbf{ny}$ est quasi-positif, puisque E et \mathbf{n} sont positifs. Les variables de \mathbf{x} peuvent être choisies fraîches, et ainsi distinctes une à une avec \mathbf{y} . Le système linéaire $E\mathbf{x} \doteq \mathbf{ny}$ est fortement triangulaire, puisque chaque variable de \mathbf{y} apparaît exactement dans une équation et jamais sur la gauche. De plus, nous pouvons définir ϕ' comme $E\mathbf{x} \doteq \mathbf{ny}$. Le calcul des modes élémentaires, et donc de E , peut être fait au plus en temps exponentiel dans la taille de ϕ avec la méthode de la double description de Motzkin (Gagneur and Klamt, 2004 ; Komei and Alain, 1996 ; T.S. et al., 1953). \square

2.11 La logique du premier-ordre

Rappelons d'abord la logique du premier-ordre standard. Pour ceci nous prenons \mathcal{V} un ensemble de variables, $\Sigma = \bigcup_{m \in \mathbb{N}} F^{(m)} \cup C$ une signature rangée, et \mathcal{E}_Σ les Σ -expressions avec variables dans \mathcal{V} .

L'ensemble des formules logiques $\phi \in \mathcal{F}_\Sigma$ sont construites selon la syntaxe abstraite de la figure FIGURE 2.3 à partir de Σ -équations $e \doteq e'$ entre des expressions $e, e' \in \mathcal{E}_\Sigma$ dont les variables sont contenues dans \mathcal{V} , et des connectives du premier-ordre.

Pour simplifier, nous définissons la formule $\text{true} =_{\text{def}} 1 \doteq 1$. Étant donné une séquence de formules ϕ_1, \dots, ϕ_n , nous pouvons définir $\bigwedge_{i=1}^n \phi_i$ comme la formule

$$\phi, \phi' \in \mathcal{F}_\Sigma ::= e \doteq e' \mid \exists x.\phi \mid \phi \wedge \phi' \mid \neg\phi \quad \text{où } x \in \mathcal{V} \text{ et } e, e' \in \mathcal{E}_\Sigma$$

FIGURE 2.2 – Formules de la logique du premier-ordre

$$\begin{aligned} \llbracket \phi \wedge \phi' \rrbracket^{\alpha, S} &= \llbracket \phi \rrbracket^{\alpha, S} \wedge^{\mathbb{B}} \llbracket \phi' \rrbracket^{\alpha, S} \\ \llbracket \exists x.\phi \rrbracket^{\alpha, S} &= \begin{cases} 1 & \text{if exists } s \in \text{dom}(S). \\ & \llbracket \phi \rrbracket^{\alpha[x/s], S} = 1 \\ 0 & \text{else} \end{cases} \\ \llbracket \neg\phi \rrbracket^{\alpha, S} &= \neg^{\mathbb{B}}(\llbracket \phi \rrbracket^{\alpha, S}) \end{aligned}$$

FIGURE 2.3 – Interprétation booléenne des formules $\llbracket \phi \rrbracket^{\alpha, S} \in \mathbb{B}$.

$\phi_1 \wedge \dots \wedge \phi_n$ qui est égale à *true* dans le cas où n vaut zéro. De plus, les formules $e \neq 0$ sont définies par $\neg e \doteq 0$. Il est aussi possible de définir des formules $e' \in E$ pour toute expression $e' \in \mathcal{E}_\Sigma$ et sous-ensemble d'expressions $E \subseteq \mathcal{E}_\Sigma$:

$$e' \in E \stackrel{\text{def}}{=} \bigvee_{e \in E} e' \doteq e$$

Cela sera utilisé par la suite pour les formules de $\mathcal{F}_{\Sigma_{\text{pos-arith}}[\text{dom}(\Delta_6)]}$ du type $v \in \{\uparrow, \sim\}$.

La sémantique d'une formule $\phi \in \mathcal{F}_\Sigma$ est une valeur booléenne, qui dépend de la Σ -structure S d'interprétation et d'un assignement de variables $\alpha : \mathcal{V} \rightarrow \text{dom}(S)$.

L'ensemble des variables libres $\text{fv}(\phi) \subseteq \mathcal{V}$ d'une formule ϕ est défini comme suit :

$$\begin{aligned} \text{fv}(e \doteq e') &= \text{fv}(e) \cup \text{fv}(e') & \text{fv}(\exists x.\phi) &= \text{fv}(\phi) \setminus \{x\} \\ \text{fv}(\phi \wedge \phi') &= \text{fv}(\phi) \cup \text{fv}(\phi') & \text{fv}(\neg\phi) &= \text{fv}(\phi) \end{aligned}$$

Soit S une Σ -structure et α un assignement de variables dans S . Nous avons déjà défini une interprétation ensembliste $\llbracket e \rrbracket^{\alpha, S} \subseteq \text{dom}(S)$ pour les Σ -expressions $e \in \mathcal{E}_\Sigma$ avec $\text{fv}(e) \subseteq V$ en FIGURE 2.1. Nous rappelons que l'égalité est interprétée de façon non déterministe lorsque S est une Σ -structure sans être une algèbre. Puis, nous avons assigné une valeur booléenne $\llbracket e \doteq e' \rrbracket^{\alpha, S} \in \mathbb{B}$ à chaque Σ -équation en SECTION 2.9.

En se basant sur cela, nous définissons maintenant dans FIGURE 2.3, pour chaque formule $\phi \in \mathcal{F}_\Sigma$ avec $\text{fv}(\phi) \subseteq V$, une valeur booléenne $\llbracket \phi \rrbracket^{\alpha, S} \in \mathbb{B}$. L'ensemble de solutions d'une formule $\phi \in \mathcal{F}_\Sigma$ sur une Σ -structure S , dans le respect d'un ensemble de variables $V \supseteq \text{fv}(\phi)$, est défini par :

$$\text{sol}_V^S(\phi) = \{\alpha : V \rightarrow \text{dom}(S) \mid \llbracket \phi \rrbracket^{\alpha, S} = 1\}$$

Lorsque $V = fv(\phi)$ l'index V sera omis et donc $sol^S(\phi) = sol_V^S(\phi)$.

Descriptions logiques de différences

Nous montrons comment abstraire les différences concrètes en \mathbb{R}_+^2 en différences abstraites. Pour ceci, nous regardons deux Σ -structures finies de différences abstraites Δ_3 et Δ_6 , où $\Sigma = F_{pos-arith}^{(2)} \cup C_{pos-arith}$ est la signature arithmétique positive. Puis, nous présentons une logique du premier ordre avec des n -uplets permettant de décrire des différences concrètes et de raisonner avec des Σ -abstractions par le Théorème de John dans sa forme généralisée. Finalement, nous présentons des résultats novateurs sur les définitions d'application de fonctions en logique du premier ordre, qui seront une clé du résultat majeur de cette thèse, le calcul exact de l'abstraction de différences des solutions d'un système d'équations linéaires.

Sommaire

3.1	La Σ-algèbre de n-uplets S^n	60
3.2	Abstraction de différences	61
3.2.1	De partition vers abstraction	61
3.2.2	L'abstraction vers Δ_3	62
3.2.3	L'abstraction vers Δ_6	63
3.3	Logique du premier-ordre avec n-uplets	63
3.3.1	Syntaxe et sémantique	64
3.3.2	Liens avec l'algèbre des n -uplets	65
3.3.3	Application aux équations polynomiales	67
3.3.4	Encoder les n -uplets en logique standard	68
3.3.5	Propriété de commutation	71

3.4	Abstraire les différences de solutions	71
3.5	Description logique d'application de fonctions	75

3.1 La Σ -algèbre de n -uplets S^n

Soit Σ une signature de symboles de fonction. Pour chaque Σ -algèbre S et nombre naturel $n \in \mathbb{N}$ nous définissons la Σ -algèbre de n -uplets $S^n = (dom(S)^n, \cdot^{S^n})$ telle que pour tout $s_1, \dots, s_n, s'_1, \dots, s'_n \in dom(S)$ et $f \in F^{(m)}$ où $m \in \mathbb{N}$:

$$f^{S^n}((s_1^1, \dots, s_n^1), \dots, (s_1^m, \dots, s_n^m)) = (f^S(s_1^1, \dots, s_1^m), \dots, f^S(s_n^1, \dots, s_n^m))$$

Les constantes $c \in C$ sont interprétées comme $c^{S^n} = (c^S, \dots, c^S)$.

Dans le cas de $\Sigma = \Sigma_{pos-arith} = F_{pos-arith}^{(2)} \cup F_{pos-arith}^{(1)} \cup C_{pos-arith}$, notons que si 0^S est l'élément neutre de $+^S$, alors 0^{S^n} est aussi l'élément neutre de $+^{S^n}$. De façon similaire, si 1^S est l'élément neutre de $*^S$ alors 1^{S^n} est aussi l'élément neutre de $*^{S^n}$. De plus, l'associativité et la commutativité de $+^{S^n}$ et $*^{S^n}$ découlent de $+^S$ et $*^S$ respectivement.

Etant donné cela, l'algèbre \mathbb{R}_+^2 a l'élément neutre $(0, 0)$ pour $+^{\mathbb{R}_+^2}$ et l'élément neutre $(1, 1)$ pour $*^{\mathbb{R}_+^2}$, et ces opérations sont associatives et commutatives.

Pour chaque fonction $h : A \rightarrow B$ et $n \in \mathbb{N}$ nous pouvons définir la fonction $h^n : A^n \rightarrow B^n$ telle que $h^n(a_1, \dots, a_n) = (h(a_1), \dots, h(a_n))$ pour tout $a_1, \dots, a_n \in A$.

Lemme 19. *Si h est une Σ -abstraction de S à Δ alors h^n est une Σ -abstraction de S^n à Δ^n .*

Démonstration. Soit $f \in F^{(m)}$ où $m \in \mathbb{N}$ et $t^1 = (t_1^1, \dots, t_n^1), \dots, t^m = (t_1^m, \dots, t_n^m) \in dom(S)^n$. Alors nous avons:

$$\begin{aligned} h^n(f^{S^n}(t^1, \dots, t^m)) &= h^n(f^{S^n}((t_1^1, \dots, t_n^1), \dots, (t_1^m, \dots, t_n^m))) \\ &\quad \text{définitions de } h^n \text{ et } S^n \\ &= (h(f^S(t_1^1, \dots, t_1^m)), \dots, h(f^S(t_n^1, \dots, t_n^m))) \\ &\quad \text{puisque } h \text{ est un homomorphisme} \\ &\subseteq ((f^\Delta(h(t_1^1), \dots, h(t_1^m))), \dots, (f^\Delta(h(t_n^1), \dots, h(t_n^m)))) \\ &\quad \text{définition de } \Delta^n \\ &= (f^{\Delta^n}((h(t_1^1), \dots, h(t_n^1)), \dots, (h(t_1^m), \dots, h(t_n^m)))) \\ &\quad \text{définition de } h^n \\ &= f^{\Delta^n}(h^n((t_1^1, \dots, t_n^1), \dots, (t_1^m, \dots, t_n^m))) \\ &= f^{\Delta^n}(h^n(t^1), \dots, h^n(t^m)) \end{aligned}$$

Pour résumer, on a bien $h^n(f^{S^n}(t^1, \dots, t^m)) \subseteq f^{\Delta^n}(h^n(t^1), \dots, h^n(t^m))$. Finalement, pour les constantes $c \in C$ nous avons :

$$\begin{aligned} h^n(c^{S^n}) &= h^n(c^S, \dots, c^S) && \text{définition } S^n \\ &= (h(c^S), \dots, h(c^S)) && \text{définition } h^n \\ &= (c^\Delta, \dots, c^\Delta) && \text{puisque } h \text{ est un homomorphisme} \\ &= c^{\Delta^n} && \text{définition de } \Delta^n \end{aligned}$$

□

3.2 Abstraction de différences

Notre prochain objectif est d'abstraire des différences concrètes en \mathbb{R}_+^2 dans des différences abstraites comme “augmentation”, “réduction” et “sans changement”.

3.2.1 De partition vers abstraction

Une façon générique d'abstraire les différences concrètes de \mathbb{R}_+^2 est de commencer avec un ensemble fini $\Delta \subseteq \mathbb{R}_+^2$ de *différences abstraites*, et une fonction

$$h : \mathbb{R}_+^2 \rightarrow \Delta$$

qui établit comment abstraire n'importe quelle différence concrète en différence abstraite. La fonction h définit une partition de \mathbb{R}_+^2 en classes d'équivalence des différences concrètes qui sont regroupées par leur même différence abstraite.

Étant donné une telle fonction h , il y a une manière unique de définir une interprétation $.\Delta$ telle que $(\Delta, .\Delta)$ devienne une Σ -structure et h une Σ -abstraction. Pour chaque constante $c \in C$ nous devons définir

$$c^\Delta = h(c^{\mathbb{R}_+^2})$$

et pour chaque symbole de fonction $f \in F^{(m)}$, où $m \in \mathbb{N}$, nous devons définir une relation ternaire f^Δ , qui est vue comme une fonction multivaluée, $f^\Delta : \Delta^m \rightarrow 2^\Delta$ doit satisfaire pour toutes valeurs abstraites $\delta_1, \dots, \delta_m \in \Delta$:

$$f^\Delta(\delta_1, \dots, \delta_m) = \left\{ h(f^{\mathbb{R}_+^2}(p_1, \dots, p_m)) \mid \begin{array}{l} p_1, \dots, p_m \in \mathbb{R}_+^2, \\ h(p_1) = \delta_1, \dots, h(p_m) = \delta_m \end{array} \right\}$$

Lemme 20. $h : \mathbb{R}_+^2 \rightarrow \Delta$ est une $\Sigma_{\text{pos-arith}}$ -abstraction.

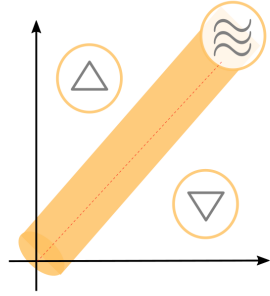


FIGURE 3.1 – La Σ -abstraction
 $h_{\Delta_3} : \mathbb{R}_+^2 \rightarrow \Delta_3$.

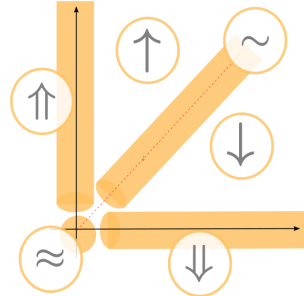


FIGURE 3.2 – La Σ -abstraction
 $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_3$.

Démonstration. Pour tout opérateur $f \in F_{pos-arith}^{(m)}$, où $m \in \{1, 2\}$, et toutes m paires de réels positifs $p^1 = (r_1^1, r_2^1), \dots, p^m = (r_1^m, r_2^m) \in \mathbb{R}_+^2$, la deuxième condition des homomorphismes s'applique comme suit :

$$h(f_{\mathbb{R}_+^2}(p^1, \dots, p^m)) = h((f_{\mathbb{R}_+}(r_1^1, \dots, r_1^m), f_{\mathbb{R}_+}(r_2^1, \dots, r_2^m))) \in f^\Delta(h(p^1), \dots, h(p^m))$$

La première condition sur les constantes $c \in C_{pos-arith}$ s'applique également, comme nous avons par définition $h(c_{\mathbb{R}_+^2}) = c^\Delta$. \square

3.2.2 L'abstraction vers Δ_3

Nous continuons avec la signature arithmétique $\Sigma_{pos-arith}$. Notre objectif ici est de présenter l'abstraction des différences concrètes de la $\Sigma_{pos-arith}$ -algèbre \mathbb{R}_+^2 vers la $\Sigma_{pos-arith}$ -structure à domaine fini $\Delta_3 = \{\Delta, \nabla, \approx\}$, qui fournit les différences abstraites pour toute "augmentation", "réduction", et "sans changement", bien connues du raisonnement qualitatif (voir [Forbus \(1997\)](#)).

Pour définir une $\Sigma_{pos-arith}$ -abstraction de \mathbb{R}_+^2 vers Δ_3 , nous regardons la partition $h_{\Delta_3}(r, r') \in \Delta_3$ telle que pour chaque $r, r' \in \mathbb{R}_+$:

$$h_{\Delta_3}(r, r') = \begin{cases} \Delta = (0, 1) & \text{si } r < r' & \text{augmentation} \\ \nabla = (1, 0) & \text{si } r > r' & \text{réduction} \\ \approx = (0, 0) & \text{si } r = r' & \text{sans changement} \end{cases}$$

Comme expliqué en sous-section 3.2.1, chaque partition d'une Σ -structure tourne l'ensemble des parties dans une Σ -structure, tel que la partition devienne une Σ -abstraction par le Lemme 20.

Les interprétations des opérateurs de la $\Sigma_{pos-arith}$ -structure $\Delta_3 = \{\Delta, \nabla, \approx\}$ sont les relations dans les tables de la FIGURE 3.3, les opérateurs binaires doivent être clos par symétrie. Puis, la partition $h_{\Delta_3} : \mathbb{R}_+^2 \rightarrow \Delta_3$ est une $\Sigma_{pos-arith}$ -abstraction.

δ	δ'	$\delta +^{\Delta_3} \delta'$	$\delta *^{\Delta_3} \delta'$
\triangle	\triangle	$\{\triangle\}$	$\{\triangle\}$
\triangle	∇	$\{\triangle, \approx, \nabla\}$	$\{\triangle, \approx, \nabla\}$
\triangle	\approx	$\{\triangle\}$	$\{\triangle, \approx\}$
\approx	\approx	$\{\approx\}$	$\{\approx\}$
∇	∇	$\{\nabla\}$	$\{\nabla\}$
∇	\approx	$\{\nabla\}$	$\{\nabla, \approx\}$

δ	$inh^{\Delta_3}(\delta)$
\triangle	∇
∇	\triangle
\approx	\approx

c	c^{Δ_3}
0	\approx
1	\approx

FIGURE 3.3 – Interprétation de la Σ -structure Δ_3 .

3.2.3 L'abstraction vers Δ_6

Tout d'abord, rappelons l'abstraction des différences concrètes dans la Σ -structure finie avec domaine $\Delta_6 = \{\uparrow, \downarrow, \sim, \uparrow, \downarrow, \approx\}$ qui était introduite pour la prédiction d'extinction de gènes dans Niehren et al. (2016). Pour définir cette Σ -structure, nous commençons avec la fonction $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_6$ telle que pour chaque nombre $r, r' \in \mathbb{R}_+$, nous avons:

$$h_{\Delta_6}(r, r') = \begin{cases} \uparrow = (1, 2) & \text{si } 0 \neq r < r' & \text{augmentation, mais pas de zéro} \\ \downarrow = (2, 1) & \text{si } r > r' \neq 0 & \text{réduction, mais pas à zéro} \\ \sim = (1, 1) & \text{si } r = r' \neq 0 & \text{sans changement, mais pas à zéro} \\ \uparrow = (0, 2) & \text{si } 0 = r < r' & \text{augmentation de zéro} \\ \downarrow = (2, 0) & \text{si } r > r' = 0 & \text{réduction à zéro} \\ \approx = (0, 0) & \text{si } r = r' = 0 & \text{sans changement à zéro} \end{cases}$$

L'interprétation des opérateurs de la $\Sigma_{pos-arith}$ -structure Δ_6 est faite par les relations dans les tables de la FIGURE 3.4, les opérateurs binaires doivent être clos par symétrie. Par le Lemme 20, la partition $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_6$ est une $\Sigma_{pos-arith}$ -abstraction.

3.3 Logique du premier-ordre avec n -uplets

Nous proposons maintenant d'étendre la logique du premier-ordre avec des n -uplets pour un paramètre $n \in \mathbb{N}$ fixé sans étendre son expressivité. Le cas des paires où $n = 2$ sera utilisé, dans les applications, pour parler des différences.

Nous montrons à présent comment compiler la logique du premier ordre avec n -uplets en logique du premier-ordre standard, et donc comment décrire des n -uplets, plus précisément des paires, d'une manière systématique en logique du premier-ordre standard.

δ	δ'	$\delta +^{\Delta_6} \delta'$	$\delta *^{\Delta_6} \delta'$
\uparrow	\uparrow	$\{\uparrow\}$	$\{\uparrow\}$
\uparrow	\downarrow	$\{\uparrow, \sim, \downarrow\}$	$\{\uparrow, \sim, \downarrow\}$
\uparrow	\sim	$\{\uparrow\}$	$\{\uparrow\}$
\uparrow	\uparrow	$\{\uparrow\}$	$\{\uparrow\}$
\uparrow	\downarrow	$\{\uparrow, \downarrow, \sim\}$	$\{\downarrow\}$
\uparrow	\approx	$\{\uparrow\}$	$\{\approx\}$
\uparrow	\downarrow	$\{\uparrow, \sim, \downarrow\}$	$\{\uparrow\}$

δ	δ'	$\delta +^{\Delta_6} \delta'$	$\delta *^{\Delta_6} \delta'$
\uparrow	\sim	$\{\uparrow\}$	$\{\uparrow\}$
\uparrow	\uparrow	$\{\uparrow\}$	$\{\uparrow\}$
\uparrow	\downarrow	$\{\uparrow, \sim, \downarrow\}$	$\{\approx\}$
\uparrow	\approx	$\{\uparrow\}$	$\{\approx\}$
\sim	\sim	$\{\sim\}$	$\{\sim\}$
\sim	\approx	$\{\sim\}$	$\{\approx\}$
\sim	\downarrow	$\{\downarrow\}$	$\{\downarrow\}$

δ	δ'	$\delta +^{\Delta_6} \delta'$	$\delta *^{\Delta_6} \delta'$
\sim	\downarrow	$\{\downarrow\}$	$\{\downarrow\}$
\approx	\approx	$\{\approx\}$	$\{\approx\}$
\approx	\downarrow	$\{\downarrow\}$	$\{\downarrow\}$
\approx	\downarrow	$\{\downarrow\}$	$\{\downarrow\}$
\downarrow	\downarrow	$\{\downarrow\}$	$\{\downarrow\}$
\downarrow	\downarrow	$\{\downarrow\}$	$\{\downarrow\}$
\downarrow	\downarrow	$\{\downarrow\}$	$\{\downarrow\}$

c	c^{Δ_6}
0	\approx
1	\sim

δ	$inh^{\Delta_6}(\delta)$
\uparrow	\downarrow
\uparrow	\downarrow
\sim	\sim

δ	$inh^{\Delta_6}(\delta)$
\approx	\sim
\downarrow	\uparrow
\downarrow	\uparrow

FIGURE 3.4 – Interprétation de la Σ -structure Δ_6 .

3.3.1 Syntaxe et sémantique

Nous fixons un naturel $n \in \mathbb{N}$ comme paramètre de la logique et une signature relationnelle $\Sigma = \bigcup_{n \in \mathbb{N}} F^{(n)} \cup C$. Pour augmenter la lisibilité, nous supposons que $F^n = \emptyset$ pour tout $n > 2$. Chaque variable dénotera un n -uplet d'éléments du domaine de la Σ -structure d'interprétation, alors que l'interprétation des constantes et des symboles de fonction en Σ restera inchangée.

La syntaxe de la logique du premier-ordre avec n -uplets est donnée par la FIGURE 3.5. Ses expressions $o \in \mathcal{O}_\Sigma^n$ sont comme l'expression de la logique standard $e \in \mathcal{E}_\Sigma$ excepté le fait que les variables x sont maintenant remplacées par les expressions de projection $\pi_i(x)$ où $1 \leq i \leq n$. Cela prend son explication dans le fait que dans cette logique chaque variable correspond à un n -uplet de valeurs, plutôt qu'à une unique valeur. Le seul changement dans la sémantique est que les affectations de variables β utilisent des n -uplets de valeurs du domaine, et donc que:

$$\llbracket \pi_i(x) \rrbracket^{\beta, S} = \{\pi_i(\beta(x))\}$$

L'ensemble des solutions d'une formule $\psi \in \mathcal{F}_\Sigma^n$ sur une Σ -structure S est défini comme suit:

$$n\text{-sol}^S(\psi) = \{\beta : fv(\psi) \rightarrow dom(S)^n \mid \llbracket \psi \rrbracket^{\beta, S} = 1\}$$

$$\begin{aligned}
 o \in \mathcal{O}_\Sigma^n &::= \pi_i(x) \mid c \mid o \odot o \mid f(o) && \text{où } 1 \leq i \leq n, c \in C, f \in F^{(1)} \text{ et } \odot \in F^{(2)}. \\
 \psi \in \mathcal{F}_\Sigma^n &::= o \doteq o' \mid \exists x.\psi \mid \psi \wedge \psi \mid \neg\psi && \text{où } x \in \mathcal{V}
 \end{aligned}$$

 FIGURE 3.5 – Les expressions et formules de la logique du premier-ordre avec n -uplets.

Interprétation des expressions $o \in \mathcal{O}_\Sigma^n$ comme les ensembles $\llbracket o \rrbracket^{\beta,S} \subseteq S$:

$$\llbracket o \odot o' \rrbracket^{\beta,S} = \cup\{(s \odot^S s') \mid s \in \llbracket o \rrbracket^{\beta,S}, s' \in \llbracket o' \rrbracket^{\beta,S}\}$$

$$\llbracket f(o) \rrbracket^{\beta,S} = \cup\{f^S(s) \mid s \in \llbracket o \rrbracket^{\beta,S}\}$$

$$\llbracket c \rrbracket^{\beta,S} = \{c^S\}, \quad \llbracket \pi_i(x) \rrbracket^{\beta,S} = \{\pi_i(\beta(x))\}$$

Interprétation des formules $\psi \in \mathcal{F}_\Sigma^n$ comme des valeurs vraies $\llbracket \psi \rrbracket^{\beta,S} \in \mathbb{B}$:

$$\llbracket o \doteq o' \rrbracket^{\beta,S} = \begin{cases} 1 & \text{if } \llbracket o \rrbracket^{\beta,S} \cap \llbracket o' \rrbracket^{\beta,S} \neq \emptyset \\ 0 & \text{else} \end{cases}$$

$$\llbracket \exists x.\psi \rrbracket^{\beta,S} = \begin{cases} 1 & \text{if exists } s \in \text{dom}(S)^n. \llbracket \psi \rrbracket^{\beta[x/s],S} = 1 \\ 0 & \text{else} \end{cases}$$

$$\llbracket \psi \wedge \psi' \rrbracket^{\beta,S} = \llbracket \psi \rrbracket^{\beta,S} \wedge^{\mathbb{B}} \llbracket \psi' \rrbracket^{\beta,S} \quad \llbracket \neg\psi \rrbracket^{\beta,S} = \neg^{\mathbb{B}}(\llbracket \psi \rrbracket^{\beta,S})$$

 FIGURE 3.6 – Sémantique des Σ -formules du premier-ordre avec n -uplets, où S est une Σ -structure et $\beta : \mathcal{V} \rightarrow \text{dom}(S)^n$ un affectation de variables.

3.3.2 Liens avec l'algèbre des n -uplets

Voyons à présent comment exprimer toute formule du premier-ordre de \mathcal{F}_Σ , interprétée sur une algèbre de n -uplet S^n , par une formule de \mathcal{F}_Σ^n interprétée sur S . Premièrement, une expression du premier ordre $e \in \mathcal{E}_\Sigma$ est convertie – celle qui est interprétée sur la Σ -algèbre S^n – en n expressions projetées $\Pi_i(e) \in \mathcal{O}_\Sigma^n$ où $1 \leq i \leq n$. Pour tout opérateur $f \in F^{(1)}$, $\odot \in F^{(2)}$ et constante $c \in C$ il est défini :

$$\begin{aligned}
 \Pi_i(f(e)) &=_{\text{def}} f(\Pi_i(e)) & \Pi_i(e \odot e') &=_{\text{def}} \Pi_i(e) \odot \Pi_i(e') \\
 \Pi_i(x) &=_{\text{def}} \pi_i(x) & \Pi_i(c) &=_{\text{def}} c
 \end{aligned}$$

Deuxièmement, toute formule $\phi \in \mathcal{F}_\Sigma$ sans n -uplets est convertie – celle qui est interprétée sur l'algèbre des n -uplets S^n – en une formule $\langle \phi \rangle^n \in \mathcal{F}_\Sigma^n$ avec des

n -uplets.

$$\langle e \doteq e' \rangle^n =_{\text{def}} \bigwedge_{i=1}^n \Pi_i(e) \doteq \Pi_i(e') \quad \langle \phi \wedge \phi' \rangle^n =_{\text{def}} \langle \phi \rangle^n \wedge \langle \phi' \rangle^n$$

$$\langle \neg \phi \rangle^n =_{\text{def}} \neg \langle \phi \rangle^n \quad \langle \exists x. \phi \rangle^n =_{\text{def}} \exists x. \langle \phi \rangle^n$$

Lemme 21. Pour chaque $e \in \mathcal{E}_\Sigma$ et Σ -algèbre S , $n \geq 1$, et $\beta : V \rightarrow \text{dom}(S)^n$ avec $\mathcal{V}_\phi \subseteq V \subseteq \mathcal{V}$:

$$\llbracket e \rrbracket^{\beta, S^n} = \llbracket (\Pi_1(e), \dots, \Pi_n(e)) \rrbracket^{\beta, S}$$

Démonstration. Par induction sur la structure des expressions de \mathcal{E}_Σ .

Cas constantes $c \in C$.

$$\llbracket c \rrbracket^{\beta, S^n} = c^{S^n} = (c^S, \dots, c^S) = \llbracket (\Pi_1(c), \dots, \Pi_n(c)) \rrbracket^{\beta, S}$$

Cas variables $x \in \mathcal{V}$.

$$\llbracket x \rrbracket^{\beta, S^n} = \beta(x) = \beta((\pi_1(x), \dots, \pi_n(x))) = \llbracket (\Pi_1(x), \dots, \Pi_n(x)) \rrbracket^{\beta, S}$$

Cas expressions $f(e)$ où $e \in \mathcal{E}_\Sigma$ et $f \in F^{(1)}$.

$$\begin{aligned} \llbracket f(e) \rrbracket^{\beta, S^n} &= \cup \{ (f^{S^n}(s)) \mid s \in \llbracket e \rrbracket^{\beta, S^n} \} \\ \text{ind.hyp.} &= \cup \{ (f^{S^n}(s)) \mid s \in \llbracket (\Pi_1(e), \dots, \Pi_n(e)) \rrbracket^{\beta, S} \} \\ &= \llbracket (f^S(\Pi_1(e)), \dots, f^S(\Pi_n(e))) \rrbracket^{\beta, S} \end{aligned}$$

Cas expressions $e_1 \odot e_2$ où $e_1, e_2 \in \mathcal{E}_\Sigma$ et $\odot \in F^{(2)}$.

$$\begin{aligned} \llbracket e_1 \odot e_2 \rrbracket^{\beta, S^n} &= \cup \{ (s_1 \odot^{S^n} s_2) \mid s_j \in \llbracket e_j \rrbracket^{\beta, S^n} \} \\ \text{ind.hyp.} &= \cup \{ (s_1 \odot^{S^n} s_2) \mid s_j \in \llbracket (\Pi_1(e_j), \dots, \Pi_n(e_j)) \rrbracket^{\beta, S} \} \\ &= \llbracket (\Pi_1(e_1) \odot^S \Pi_1(e_2), \dots, \Pi_n(e_1) \odot^S \Pi_n(e_2)) \rrbracket^{\beta, S} \end{aligned}$$

□

Proposition 22. Pour chaque $n \geq 1$, formule $\phi \in \mathcal{F}_\Sigma$ et Σ -algèbre S :

$$\text{sol}^{S^n}(\phi) = n\text{-sol}^S(\langle \phi \rangle^n)$$

Démonstration. Par induction sur la structure des formules de \mathcal{F}_Σ . Le cas de base des Σ -équations vient essentiellement du Lemme 21. Soit ϕ une Σ -équation de la forme $e \doteq e'$ où $e, e' \in \mathcal{E}_\Sigma$ et β un affectation de variables $\beta : V \rightarrow \text{dom}(S)^n$ tel que $\mathcal{V}(\phi) \subseteq V \subseteq \mathcal{V}$. Ainsi :

Cas $e \doteq e'$ où $e, e' \in \mathcal{E}_\Sigma$.

$$\begin{aligned} \text{sol}^{S^n}(e \doteq e') &= \{ \beta \mid \llbracket e \doteq e' \rrbracket^{\beta, S^n} = 1 \} \\ \text{par Prop. préc.} &= \{ \beta \mid \llbracket \bigwedge_{i=1}^n \Pi_i(e) \doteq \Pi_i(e') \rrbracket^{\beta, S} = 1 \} \\ &= \{ \beta \mid \llbracket \langle e \doteq e' \rangle^n \rrbracket^{\beta, S} = 1 \} \\ &= n\text{-sol}^S(\langle e \doteq e' \rangle^n) \end{aligned}$$

Cas $\phi \wedge \phi'$ où $\phi, \phi' \in \mathcal{F}_\Sigma$.

$$\begin{aligned}
 \text{ind.hyp. } \text{sol}^{S^n}(\phi \wedge \phi') &= \{\beta \mid \llbracket \phi \wedge \phi' \rrbracket^{\beta, S^n} = 1\} \\
 &= \{\beta \mid \llbracket \phi \rrbracket^{\beta, S^n} \wedge \llbracket \phi' \rrbracket^{\beta, S^n} = 1\} \\
 &= \{\beta \mid \llbracket \langle \phi \rangle^n \rrbracket^{\beta, S} \wedge \llbracket \langle \phi' \rangle^n \rrbracket^{\beta, S} = 1\} \\
 &= \{\beta \mid \llbracket \langle \phi \wedge \phi' \rangle^n \rrbracket^{\beta, S} = 1\} \\
 &= n\text{-sol}^S(\langle \phi \wedge \phi' \rangle^n)
 \end{aligned}$$

Cas $\neg\phi$ où $\phi \in \mathcal{F}_\Sigma$.

$$\begin{aligned}
 \text{ind.hyp. } \text{sol}^{S^n}(\neg\phi) &= \{\beta \mid \llbracket \neg\phi \rrbracket^{\beta, S^n} = 1\} \\
 &= \{\beta \mid \neg \llbracket \phi \rrbracket^{\beta, S^n} = 1\} \\
 &= \{\beta \mid \neg \llbracket \langle \phi \rangle^n \rrbracket^{\beta, S} = 1\} \\
 &= \{\beta \mid \llbracket \neg \langle \phi \rangle^n \rrbracket^{\beta, S} = 1\} \\
 &= \{\beta \mid \llbracket \langle \neg\phi \rangle^n \rrbracket^{\beta, S} = 1\} \\
 &= n\text{-sol}^S(\langle \neg\phi \rangle^n)
 \end{aligned}$$

Cas $\exists x.\phi$ où $\phi \in \mathcal{F}_\Sigma$.

$$\begin{aligned}
 \text{ind.hyp. } \text{sol}^{S^n}(\exists x.\phi) &= \{\beta \mid \llbracket \exists x.\phi \rrbracket^{\beta, S^n} = 1\} \\
 &= \{\beta \mid \text{exists } s \in \text{dom}(S)^n. \llbracket \phi \rrbracket^{\beta[x/s], S^n} = 1\} \\
 &= \{\beta \mid \text{exists } s \in \text{dom}(S)^n. \llbracket \langle \phi \rangle^n \rrbracket^{\beta[x/s], S} = 1\} \\
 &= \{\beta \mid \llbracket \exists x.\langle \phi \rangle^n \rrbracket^{\beta, S} = 1\} \\
 &= \{\beta \mid \llbracket \langle \exists x.\phi \rangle^n \rrbracket^{\beta, S} = 1\} \\
 &= n\text{-sol}^S(\langle \exists x.\phi \rangle^n)
 \end{aligned}$$

□

3.3.3 Application aux équations polynomiales

Les différences concrètes en \mathbb{R}_+^2 peuvent être décrites par des systèmes d'équations polynomiales $\phi \in \mathcal{E}_{\Sigma_{\text{pos-arith}}}$ de la logique du premier-ordre interprétés sur \mathbb{R}_+^2 . Comme le montre la Proposition 22, de tels systèmes peuvent être décrits par des systèmes d'équations polynomiales $\langle \phi \rangle^2 \in \mathcal{F}_{\Sigma_{\text{pos-arith}}}^2$ dans la logique du premier-ordre avec des paires, mais être interprétés sur \mathbb{R}_+ . Ceci est fait en dupliquant chaque équation sur \mathbb{R}_+^2 en deux équations sur \mathbb{R}_+ , comme l'illustre l'exemple ci-dessous.

Exemple 23. Soit $\phi \in \mathcal{F}_{\Sigma_{\text{pos-arith}}}$ l'équation polynomiale en logique du premier-ordre standard suivante :

$$3x + 4y^5 \doteq 0$$

La Proposition 22 montre que ϕ a les mêmes solutions sur \mathbb{R}_+^2 que la formule $\langle \phi \rangle^2 \in \mathcal{F}_{\Sigma^{\text{pos-arith}}}^2$ de la logique du premier-ordre avec des n -uplets sur \mathbb{R}_+ . Cette dernière correspond au système d'équations polynomiales avec n -uplets :

$$3\dot{\pi}_1(x) + 4\dot{\pi}_1(y)^5 \doteq 0 \wedge 3\dot{\pi}_2(x) + 4\dot{\pi}_2(y)^5 \doteq 0$$

Ce système est une formule de la logique du premier-ordre avec n -uplets, la présence des projections en témoigne. Chaque équation du système original est doublée de façon à avoir deux projections des différences concrètes en \mathbb{R}_+^2 .

3.3.4 Encoder les n -uplets en logique standard

Le prochain objectif est de réduire la logique avec n -uplet à la logique standard. L'idée est d'introduire des variables fraîches pour les projections.

Exemple 24. Étant donné deux générateurs de variables fraîches v_1 et v_2 , le système d'équations polynomiales

$$3\dot{\pi}_1(x) + 4\dot{\pi}_1(y)^5 \doteq 0 \wedge 3\dot{\pi}_2(x) + 4\dot{\pi}_2(y)^5 \doteq 0$$

dans la logique du premier-ordre avec des paires peut être traduit en système d'équations polynomiales :

$$3v_1(x) + 4v_1(y)^5 \doteq 0 \wedge 3v_2(x) + 4v_2(y)^5 \doteq 0$$

dans la logique du premier-ordre standard. Les 2 variables fraîches $v_i(x)$ et $v_i(y)$ correspondent aux projections $\dot{\pi}_i(x)$ et $\dot{\pi}_i(y)$ respectives.

Plus généralement, nous souhaitons réécrire n'importe quelle formule de la logique avec des n -uplets $\psi \in \mathcal{F}_{\Sigma}^n$ en formule de la logique du premier-ordre standard $\tilde{v}(\psi) \in \mathcal{F}_{\Sigma}$ en introduisant des variables fraîches pour les projections. Pour ce faire, nous fixons n générateurs de variables fraîches $v_1, \dots, v_n : \mathcal{V} \rightarrow \mathcal{V}$. Et nous associons chaque expression $o \in \mathcal{O}_{\Sigma}^n$ avec des projections à une expression $\tilde{v}(o) \in \mathcal{E}_{\Sigma}$ sans nouvelle variable:

$$\begin{aligned} \tilde{v}(\dot{\pi}_i(x)) &=_{\text{def}} v_i(x), & \tilde{v}(f(o)) &=_{\text{def}} f(\tilde{v}(o)), \\ \tilde{v}(c) &=_{\text{def}} c, & \tilde{v}(o \odot o') &=_{\text{def}} \tilde{v}(o) \odot \tilde{v}(o'). \end{aligned}$$

Pour finir, chaque formule avec projections $\psi \in \mathcal{F}_{\Sigma}^n$ est associée à une formule $\tilde{v}(\psi) \in \mathcal{F}_{\Sigma}$ avec des variables fraîches :

$$\begin{aligned} \tilde{v}(o = o') &=_{\text{def}} \tilde{v}(o) = \tilde{v}(o') & \tilde{v}(\neg\psi) &=_{\text{def}} \neg\tilde{v}(\psi) \\ \tilde{v}(\psi \wedge \psi') &=_{\text{def}} \tilde{v}(\psi) \wedge \tilde{v}(\psi') & \tilde{v}(\exists x.\psi) &=_{\text{def}} \exists v_1(x) \dots \exists v_n(x). \tilde{v}(\psi) \end{aligned}$$

Étant donné un affectation de variables $\beta : V \rightarrow \text{dom}(S)^n$ avec $V \subseteq \mathcal{V}$, nous définissons $\nu(\beta) : \uplus_{i=1}^n \nu_i(V) \rightarrow \text{dom}(S)$ tel que pour tout $x \in V$:

$$\nu(\beta)(\nu_i(x)) = \pi_i(\beta(x))$$

La fonction ν est une bijection avec rang $\{\alpha \mid \alpha : \uplus_{i=1}^n \nu_i(V) \rightarrow \text{dom}(S)\}$. L'inverse de cette fonction satisfait $\nu^{-1}(\alpha)(x) = (\alpha(\nu_1(x)), \dots, \alpha(\nu_n(x)))$ pour tout α du rang et tout $x \in V$.

Lemme 25. *Pour chaque expression $o \in \mathcal{O}_\Sigma^n$ et affectation de variables $\beta : V \rightarrow \text{dom}(S)^n$ avec $\mathcal{V}(o) \subseteq V \subseteq \mathcal{V}$ nous avons $\llbracket \tilde{\nu}(o) \rrbracket^{\nu(\beta), S} = \llbracket o \rrbracket^{\beta, S}$.*

Démonstration. Par induction sur la structure des Σ -expressions $o \in \mathcal{O}_\Sigma^n$.

Cas constantes $c \in C$.

$$\llbracket \tilde{\nu}(c) \rrbracket^{\nu(\beta), S} = \llbracket c \rrbracket^{\beta, S}$$

Cas $\pi_i(x)$ où $x \in \mathcal{V}$ et $1 \leq i \leq n$.

$$\begin{aligned} \llbracket \tilde{\nu}(\pi_i(x)) \rrbracket^{\nu(\beta), S} &= \{\nu(\beta)(\nu_i(x))\} \\ &= \{\pi_i(\beta(x))\} \\ &= \llbracket o \rrbracket^{\beta, S} \end{aligned}$$

Cas $f(o)$ où $o \in \mathcal{O}_\Sigma^n$ et $f \in F^{(1)}$.

$$\begin{aligned} \llbracket \tilde{\nu}(f(o)) \rrbracket^{\nu(\beta), S} &= \llbracket f(\tilde{\nu}(o)) \rrbracket^{\nu(\beta), S} \\ &= \cup\{(f^S(s) \mid s \in \llbracket \tilde{\nu}(o) \rrbracket^{\nu(\beta), S})\} \\ \text{ind.hyp.} &= \cup\{(f^S(s) \mid s \in \llbracket o \rrbracket^{\beta, S})\} \\ &= \llbracket f(o) \rrbracket^{\beta, S} \end{aligned}$$

Cas $o_1 \odot o_2$ où $o_1, o_2 \in \mathcal{O}_\Sigma^n$ et $\odot \in F^{(2)}$.

$$\begin{aligned} \llbracket \tilde{\nu}(o_1 \odot o_2) \rrbracket^{\nu(\beta), S} &= \llbracket \tilde{\nu}(o_1) \odot \tilde{\nu}(o_2) \rrbracket^{\nu(\beta), S} \\ &= \cup\{(s_1 \odot^S s_2 \mid s_i \in \llbracket \tilde{\nu}(o_i) \rrbracket^{\nu(\beta), S})\} \\ \text{ind.hyp.} &= \cup\{(s_1 \odot^S s_2 \mid s_i \in \llbracket o_i \rrbracket^{\beta, S})\} \\ &= \llbracket o_1 \odot o_2 \rrbracket^{\beta, S} \end{aligned}$$

Cas $\pi_i(o_1, \dots, o_n)$ où $o_1, \dots, o_n \in \mathcal{O}_\Sigma^n$.

$$\begin{aligned} \llbracket \tilde{\nu}(\pi_i(o_1, \dots, o_n)) \rrbracket^{\nu(\beta), S} &= \llbracket \tilde{\nu}(o_i) \rrbracket^{\nu(\beta), S} \\ \text{ind.hyp.} &= \llbracket o_i \rrbracket^{\beta, S} \\ &= \llbracket \pi_i(o_1, \dots, o_n) \rrbracket^{\beta, S} \end{aligned}$$

□

Proposition 26. *Pour chaque $\psi \in \mathcal{F}_\Sigma^n$, Σ -structure S , et $n \geq 1$: $n\text{-sol}^S(\psi) = \nu^{-1}(\text{sol}^S(\tilde{\nu}(\psi)))$.*

Démonstration. Tout d'abord prouvons la Proposition qui suit par induction sur la structure des Σ -formules de \mathcal{F}_Σ^n , où le cas de base découle du Lemme 25.

Proposition 27. *Pour chaque affectation de variables $\beta : V \rightarrow \text{dom}(S)^n$ avec $V \subseteq \mathcal{V}$ et formule $\psi \in \mathcal{F}_\Sigma^n$ nous avons $\llbracket \tilde{\nu}(\psi) \rrbracket^{\nu(\beta), S} = \llbracket \psi \rrbracket^{\beta, S}$.*

Démonstration. La preuve de la Proposition est par induction sur la structure des Σ -formules de \mathcal{F}_Σ^n .

Cas $o \doteq o'$ où $o, o' \in \mathcal{O}_\Sigma^n$.

$$\begin{aligned} \llbracket o \doteq o' \rrbracket^{\beta, S} = 1 &\Leftrightarrow \llbracket o \rrbracket^{\beta, S} \cap \llbracket o' \rrbracket^{\beta, S} \neq \emptyset \\ \text{Lemme 25} &\Leftrightarrow \llbracket \tilde{\nu}(o) \rrbracket^{\nu(\beta), S} \cap \llbracket \tilde{\nu}(o') \rrbracket^{\nu(\beta), S} \neq \emptyset \\ &\Leftrightarrow \llbracket \tilde{\nu}(o) \doteq \tilde{\nu}(o') \rrbracket^{\nu(\beta), S} = 1 \\ &\Leftrightarrow \llbracket \tilde{\nu}(o \doteq o') \rrbracket^{\nu(\beta), S} = 1 \end{aligned}$$

Cas $\psi \wedge \psi'$ où $\psi, \psi' \in \mathcal{F}_\Sigma^n$.

$$\begin{aligned} \llbracket \psi \wedge \psi' \rrbracket^{\beta, S} = 1 &\Leftrightarrow \llbracket \psi \rrbracket^{\beta, S} \wedge \llbracket \psi' \rrbracket^{\beta, S} = 1 \\ \text{ind.hyp} &\Leftrightarrow \llbracket \tilde{\nu}(\psi) \rrbracket^{\nu(\beta), S} \wedge \llbracket \tilde{\nu}(\psi') \rrbracket^{S, \nu(\beta)} = 1 \\ &\Leftrightarrow \llbracket \tilde{\nu}(\psi) \wedge \tilde{\nu}(\psi') \rrbracket^{\nu(\beta), S} = 1 \\ &\Leftrightarrow \llbracket \tilde{\nu}(\psi \wedge \psi') \rrbracket^{\nu(\beta), S} = 1 \end{aligned}$$

Cas $\neg\psi$ où $\psi \in \mathcal{F}_\Sigma^n$.

$$\begin{aligned} \llbracket \neg\psi \rrbracket^{\beta, S} = 1 &\Leftrightarrow \neg \llbracket \psi \rrbracket^{\beta, S} = 1 \\ \text{ind.hyp.} &\Leftrightarrow \neg \llbracket \tilde{\nu}(\psi) \rrbracket^{\nu(\beta), S} = 1 \\ &\Leftrightarrow \llbracket \neg \tilde{\nu}(\psi) \rrbracket^{\nu(\beta), S} = 1 \\ &\Leftrightarrow \llbracket \tilde{\nu}(\neg\psi) \rrbracket^{\nu(\beta), S} = 1 \end{aligned}$$

Cas $\exists x.\psi$ où $\psi \in \mathcal{F}_\Sigma^n$.

$$\begin{aligned} \llbracket \exists x.\psi \rrbracket^{\beta, S} = 1 &\Leftrightarrow \text{exist } s \in \text{dom}(S)^n. \llbracket \psi \rrbracket^{\beta[x/s], S} = 1 \\ \text{ind.hyp.} &\Leftrightarrow \text{exist } s \in \text{dom}(S)^n. \llbracket \tilde{\nu}(\psi) \rrbracket^{\nu(\beta[x/s]), S} = 1 \\ &\Leftrightarrow \text{exist } s_1 \in \text{dom}(S) \dots s_n \in \text{dom}(S). \llbracket \psi \rrbracket^{\nu(\beta[\nu_i(x)/s_i]), S} = 1 \\ &\Leftrightarrow \llbracket \exists \nu_1(x) \dots \exists \nu_n(x). \tilde{\nu}(\psi) \rrbracket^{\nu(\beta), S} = 1 \\ &\Leftrightarrow \llbracket \exists x. \tilde{\nu}(\psi) \rrbracket^{\nu(\beta), S} = 1 \end{aligned}$$

□

La preuve de la Proposition est directe par induction sur la structure des Σ -formules de \mathcal{F}_Σ^n . Finalement, la Proposition implique la Proposition par :

$$\begin{aligned} \beta \in n\text{-sol}^S(\psi) &\Leftrightarrow \nu(\beta) \in n\text{-sol}^S(\tilde{\nu}(\psi)) \quad \text{prop. préc.} \\ &\Leftrightarrow \nu^{-1}(\nu(\beta)) \in \nu^{-1}(n\text{-sol}^S(\tilde{\nu}(\psi))) \\ &\Leftrightarrow \beta \in \nu^{-1}(n\text{-sol}^S(\tilde{\nu}(\psi))) \end{aligned}$$

□

3.3.5 Propriété de commutation

Comme auparavant, considérons n générateurs de variables fraîches v_1, \dots, v_n et l'opérateur ν^{-1} qui associe les affectations des variables fraîchement générées aux affectations de n -uplets. Nous allons montrer la propriété de commutation de l'opérateur ν^{-1} avec les Σ -abstractions.

Lemme 28. *Pour chaque Σ -abstraction $h : S \rightarrow \Delta$ et affectation de variables fraîches $\alpha : \uplus_{i=1}^n v_i(V) \rightarrow \text{dom}(S)$:*

$$\nu^{-1}(h \circ \alpha) = h^n \circ \nu^{-1}(\alpha)$$

Démonstration. Pour chaque variable $x \in V$ nous avons :

$$\begin{aligned} \nu^{-1}(h \circ \alpha)(x) &= (h(\alpha(v_1(x))), \dots, h(\alpha(v_n(x)))) \\ &= h^n((\alpha(v_1(x))), \dots, \alpha(v_n(x))) \\ &= h^n(\nu^{-1}(\alpha)(x)) \\ &= (h^n \circ \nu^{-1}(\alpha))(x) \end{aligned}$$

□

Proposition 29. *Pour chaque ensemble fini $V \subseteq \mathcal{V}$, sous-ensemble R d'affectations de variables du type $\uplus_{i=1}^n v_i(V) \rightarrow \text{dom}(S)$, et de Σ -abstraction $h : S \rightarrow \Delta$:*

$$\nu^{-1}(h \circ R) = h^n \circ \nu^{-1}(R)$$

Démonstration. Par le lemme 28: $\nu^{-1}(h \circ R) = \{\nu^{-1}(h(\alpha)) \mid \alpha \in R\} = \{h^n(\nu^{-1}(\alpha)) \mid \alpha \in R\} = h^n \circ \nu^{-1}(R)$ □

3.4 Abstraire les différences de solutions

Revoyons la notion d'abstraction des différences de [Coutte et al. \(2015\)](#) ; [John et al. \(2013b\)](#) ; [Niehren et al. \(2016\)](#) en appliquant notre notion de Σ -abstraction aux différences concrètes dans la Σ -algèbre \mathbb{R}_+^2 où $\Sigma_{\text{pos-arith}} = F_{\text{pos-arith}}^{(2)} \cup F_{\text{pos-arith}}^{(1)} \cup C_{\text{pos-arith}}$.

Plus généralement, soit S une Σ -algèbre, telle que l'algèbre \mathbb{R}_+^2 des différences concrètes, et $V \subseteq \mathcal{V}$ est un sous-ensemble de variables. Pour chaque affectation de variables $\alpha, \alpha' : V \rightarrow \text{dom}(S)$, on définit un affectation de variables à des paires d'éléments du domaine de la structure

$$\text{diff}(\alpha, \alpha') : V \rightarrow \text{dom}(S)^2$$

que nous appelons les différences de α et α' , telle que pour chaque variable $x \in V$, $\text{diff}(\alpha, \alpha')(x) = (\alpha(x), \alpha'(x))$. Pour chaque sous-ensemble R d'affectations de variables du type $V \rightarrow \text{dom}(S)$ nous définissons l'ensemble des affectations de différence de R par:

$$\text{diff}(R) = \{\text{diff}(\alpha, \alpha') \mid \alpha, \alpha' \in R\}$$

De plus, pour chaque Σ -abstraction $h : S^2 \rightarrow \Delta$ et sous-ensemble R' d'affectations de différences concrètes du type $V \rightarrow \text{dom}(S)^2$, on définit l'application de l'abstraction:

$$h \circ R' =_{\text{def}} \{h \circ \beta \mid \beta \in R'\}$$

Définition 30. Pour chaque Σ -abstraction $h : S^2 \rightarrow \Delta$ et formule $\phi \in \mathcal{F}_\Sigma$ on définit l'abstraction des différences de l'ensemble de S -solutions de ϕ par:

$$\text{sol}^S(\phi)^\Delta = h \circ \text{diff}(\text{sol}^S(\phi))$$

La définition originale de $\text{sol}(\phi)^\Delta$ dans Niehren et al. (2016) était similaire, mais ne rendait pas explicite le rôle de diff et $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_6$. En faisant ainsi, il est à présent possible de dire que l'abstraction des différences de l'ensemble de \mathbb{R}_+ -solutions d'une formule est l'ensemble de \mathbb{R}_+^2 -solutions de la même formule.

Lemme 31. Pour chaque formule $\phi \in \mathcal{F}_\Sigma$ et Σ -algèbre S : $\text{diff}(\text{sol}^S(\phi)) = \text{sol}^{S^2}(\phi)$.

Démonstration. Pour tout $\alpha : \mathcal{V} \rightarrow \text{dom}(S)$, $\alpha' : \mathcal{V} \rightarrow \text{dom}(S)$, $\alpha, \alpha' \in \text{sol}^S(\phi)$ on peut construire l'affectation de variables $\alpha'' : \mathcal{V} \rightarrow \text{dom}(S)^2$ avec $\text{diff}(\alpha, \alpha') = (\alpha(x), \alpha'(x)) = \alpha''(x)$. Pour avoir ainsi $\text{diff}(\text{sol}^S(\phi)) \subseteq \text{sol}^{S^2}(\phi)$.

Dans l'autre sens, pour tout affectation de variables $\alpha'' : \mathcal{V} \rightarrow \text{dom}(S)^2$, $\alpha'' \in \text{sol}^{S^2}(\phi)$ on peut générer deux affectations de variables $\alpha : \mathcal{V} \rightarrow \text{dom}(S)$, $\alpha' : \mathcal{V} \rightarrow \text{dom}(S) \in \text{sol}^S(\phi)$ avec $\forall x \in \mathcal{V}, \alpha(x) = \pi_1(\alpha''(x)) \wedge \alpha'(x) = \pi_2(\alpha''(x))$. Donc $\text{sol}^{S^2}(\phi) \subseteq \text{diff}(\text{sol}^S(\phi))$, et finalement $\text{diff}(\text{sol}^S(\phi)) = \text{sol}^{S^2}(\phi)$

□

Comme conséquence immédiate on a pour chaque Σ -abstraction $h : S^2 \rightarrow \Delta$ la propriété $\text{sol}(\phi)^\Delta = h_\Delta \circ \text{sol}^{S^2}(\phi)$. Notre prochain objectif est de montrer que nous pouvons surapproximer l'ensemble $\text{sol}(\phi)^\Delta$ par $\text{sol}^\Delta(\phi)$ (Corollaire 35).

Lemme 32. Soit $h' : S' \rightarrow \Delta$ une Σ -abstraction, et $\alpha : V \rightarrow \text{dom}(S')$ un affectation de variables. Pour chaque expression $e \in \mathcal{E}_\Sigma$ avec $V(e) \subseteq V$: $h'(\llbracket e \rrbracket^{\alpha, S'}) \subseteq \llbracket e \rrbracket^{h' \circ \alpha, \Delta}$.

Démonstration. La preuve est par induction sur la structure des expressions $e \in \mathcal{E}_\Sigma$. Soit α un affectation de variables de $\text{dom}(S')$. Pour chaque expression $e = f(e_1)$ où $f \in F^{(1)}$ nous avons:

$$\begin{aligned} h'(\llbracket f(e_1) \rrbracket^{\alpha, S'}) &= h'(f^{S'}(\llbracket e_1 \rrbracket^{\alpha, S'})) \\ &\subseteq f^\Delta(h'(\llbracket e_1 \rrbracket^{\alpha, S'})) \quad \text{homomorph.} \\ &\subseteq f^\Delta \llbracket e_1 \rrbracket^{h' \circ \alpha, \Delta} \quad \text{hyp. ind.} \\ &= \llbracket f(e_1) \rrbracket^{h' \circ \alpha, \Delta} \end{aligned}$$

Pour chaque expression $e = e_1 \odot e_2$ où $\odot \in F^{(2)}$ nous avons:

$$\begin{aligned} h'(\llbracket e_1 \odot e_2 \rrbracket^{\alpha, S'}) &= h'(\llbracket e_1 \rrbracket^{\alpha, S'} \odot^{S'} \llbracket e_2 \rrbracket^{\alpha, S'}) \\ &\subseteq h'(\llbracket e_1 \rrbracket^{\alpha, S'}) \odot^\Delta h'(\llbracket e_2 \rrbracket^{\alpha, S'}) \quad \text{homomorph.} \\ &\subseteq \llbracket e_1 \rrbracket^{h' \circ \alpha, \Delta} \odot^\Delta \llbracket e_2 \rrbracket^{h' \circ \alpha, \Delta} \quad \text{hyp. ind.} \\ &= \llbracket e_1 \odot e_2 \rrbracket^{h' \circ \alpha, \Delta} \end{aligned}$$

Pour chaque expression $e = x \in V$ nous avons:

$$h'(\llbracket x \rrbracket^{\alpha, S'}) = h'(\{\alpha(x)\}) = \llbracket x \rrbracket^{h' \circ \alpha, \Delta}$$

Pour chaque constante $e = c \in C$ nous avons:

$$h'(\llbracket c \rrbracket^{\alpha, S'}) = h'(c^{S'}) = c^\Delta = \llbracket c \rrbracket^{h \circ \alpha, \Delta} \quad \text{homomorph.}$$

□

Proposition 33. Soit $h : S' \rightarrow \Delta$ une Σ -abstraction et $\alpha : V \rightarrow \text{dom}(S')$ un affectation de variables. Pour chaque formule positive $\phi \in \mathcal{F}_\Sigma$ avec $V(\phi) \subseteq V$: $\llbracket \phi \rrbracket^{\alpha, S'} \leq \llbracket \phi \rrbracket^{h \circ \alpha, \Delta}$.

Démonstration. La preuve est par induction sur la structure de la Σ -formule positive ϕ . Si ϕ est une équation $e \doteq e'$ alors il suit par le Lemme 32 que: $h(\llbracket e \rrbracket^{\alpha, S'}) \subseteq \llbracket e \rrbracket^{h \circ \alpha, \Delta}$ et $h(\llbracket e' \rrbracket^{\alpha, S'}) \subseteq \llbracket e' \rrbracket^{h \circ \alpha, \Delta}$. D'où:

$$\begin{aligned} \llbracket e \doteq e' \rrbracket^{\alpha, S'} = 1 &\Leftrightarrow \llbracket e \rrbracket^{\alpha, S'} \cap \llbracket e' \rrbracket^{\alpha, S'} \neq \emptyset \\ &\Leftrightarrow h(\llbracket e \rrbracket^{\alpha, S'}) \cap h(\llbracket e' \rrbracket^{\alpha, S'}) \neq \emptyset \\ &\Rightarrow \llbracket e \rrbracket^{h \circ \alpha, \Delta} \cap \llbracket e' \rrbracket^{h \circ \alpha, \Delta} \neq \emptyset \quad \text{Lemme 32} \\ &\Leftrightarrow \llbracket e \doteq e' \rrbracket^{h \circ \alpha, \Delta} = 1 \end{aligned}$$

Cela montre que $\llbracket e \doteq e' \rrbracket^{\alpha, S'} \leq \llbracket e \doteq e' \rrbracket^{h \circ \alpha, \Delta}$ comme requis. On considère ensuite le cas où ϕ est une conjonction de la forme $\phi' \wedge \phi''$.

$$\begin{aligned} \llbracket \phi' \wedge \phi'' \rrbracket^{\alpha, S'} &= \llbracket \phi' \rrbracket^{\alpha, S'} \wedge^{\mathbb{B}} \llbracket \phi'' \rrbracket^{\alpha, S'} \\ &\leq \llbracket \phi' \rrbracket^{h \circ \alpha, \Delta} \wedge^{\mathbb{B}} \llbracket \phi'' \rrbracket^{h \circ \alpha, \Delta} \quad \text{hyp. ind.} \\ &= \llbracket \phi' \wedge \phi'' \rrbracket^{h \circ \alpha, \Delta} \end{aligned}$$

Pour le dernier cas où ϕ est une formule avec quantificateur existentiel de la forme $\exists x.\phi'$.

$$\begin{aligned} \llbracket \exists x.\phi' \rrbracket^{\alpha, S'} = 1 &\Leftrightarrow (\text{existe } s \in \text{dom}(S'). \llbracket \phi' \rrbracket^{\alpha[x/s], S'}) = 1 \\ &\Rightarrow (\text{existe } s \in \text{dom}(S'). \llbracket \phi' \rrbracket^{h\circ\alpha[x/s], \Delta}) = 1 \quad \text{hyp. ind.} \\ &\Leftrightarrow \llbracket \exists x.\phi' \rrbracket^{h\circ\alpha, \Delta} = 1 \end{aligned}$$

Cela montre que $\llbracket \exists x.\phi' \rrbracket^{\alpha, S'} \leq \llbracket \exists x.\phi' \rrbracket^{h\circ\alpha, \Delta}$ comme requis. \square

Théorème 34 ((Théorème de John généralisé)). *Soit $h : S \rightarrow \Delta$ une Σ -abstraction et $\alpha : V \rightarrow \text{dom}(S)$ un affectation de variables. Pour chaque formule positive $\phi \in \mathcal{F}_\Sigma$ avec $\mathcal{V}(\phi) \subseteq V$:*

$$h \circ \text{sol}^S(\phi) \subseteq \text{sol}^\Delta(\phi)$$

Démonstration. Soit h une Σ -abstraction de S à Δ et $\phi \in \mathcal{F}_\Sigma$ une formule positive. Pour chaque affectation de variables α de $\text{dom}(S)$, nous savons que $\llbracket \phi \rrbracket^{\alpha, S} \leq \llbracket \phi \rrbracket^{h\circ\alpha, \Delta}$ par la Proposition 33 puisque ϕ est positive. Cela équivaut à $\{h \circ \alpha \mid \alpha \in \text{sol}^S(\phi)\} \subseteq \text{sol}^\Delta(\phi)$ et ainsi $h \circ \text{sol}^S(\phi) \subseteq \text{sol}^\Delta(\phi)$ comme requis \square

Corollaire 35 ((John et al., 2013b ; Niehren et al., 2016)). *Pour $\Delta \in \{\Delta_3, \Delta_6\}$ et chaque formule positive du premier-ordre $\phi \in \mathcal{F}_\Sigma$:*

$$\text{sol}^{\mathbb{R}_+}(\phi)^\Delta \subseteq \text{sol}^\Delta(\phi)$$

Démonstration. Avec la Σ -structure $S = \mathbb{R}_+^2$ et $h : \mathbb{R}_+^2 \rightarrow \Delta$ equal to h_{Δ_3} ou h_{Δ_6} , cela découle de la Définition 30, du Lemme 31 et du Théorème 34 :

$$\text{sol}^{\mathbb{R}_+}(\phi)^\Delta = h \circ \text{diff}(\text{sol}^{\mathbb{R}_+}(\phi)) = h \circ \text{sol}^{\mathbb{R}_+^2}(\phi) \subseteq \text{sol}^\Delta(\phi)$$

\square

Si Δ est fini alors l'ensemble $\text{sol}^\Delta(\phi)$ est fini, alors que $\text{sol}^{\mathbb{R}_+}(\phi)$ est infini. De plus, si ϕ est une formule conjonctive, on peut calculer l'ensemble $\text{sol}^\Delta(\phi)$ par un solveur de contrainte à domaine fini à partir de ϕ et des tables de Δ (tel que Mini-zinc Rendl et al. (2015)). Par contre, le calcul de l'ensemble fini $h \circ \text{diff}(\text{sol}^S(\phi))$, pour des structures infinies S , reste flou. Ce problème est ouvert, même pour le cas où ϕ est un système d'équations linéaires homogènes et $S = \mathbb{R}_+$, tel que l'ensemble infini $\text{sol}^S(\phi)$ peut être représenté de façon fini par une matrice triangulaire.

Ceci est le cœur de la question que nous allons résoudre dans ce chapitre. L'approche que nous allons utiliser se base sur une réécriture de formule ϕ en une formule \mathbb{R}_+ -équivalente qui est h -exacte sous le sens suivant :

Définition 36. Soit $h : S \rightarrow \Delta$ une Σ -abstraction. Une Σ -formule ϕ est dite h -exacte si :

$$h(sol^S(\phi)) = sol^\Delta(\phi).$$

Pour chaque formule h -exacte ϕ , $h(sol_V^S(\phi))$ peut être calculé exactement en calculant $sol_V^\Delta(\phi)$ comme décrit au-dessus.

3.5 Description logique d'application de fonctions

Nous rappelons ce que cela signifie pour une fonction ou une relation d'être définie par une formule de logique des tuples du premier-ordre et comment définir en logique l'application d'une fonction définie en logique du premier-ordre. Puis nous prouvons des propriétés pour de telles définitions d'application de fonction.

Ces propriétés novatrices serviront plus tard pour calculer des abstractions des différences, par décomposition en abstraction booléenne et des fonctions définissables en logique du premier-ordre.

Définition 37. Une \mathcal{F}_Σ^n -définition d'arité m est une fonction $F : \mathcal{V}^m \rightarrow \mathcal{F}_\Sigma^n$ pour laquelle il existe une formule $\psi \in \mathcal{F}_\Sigma^n$ et une séquence de variables distinctes $\mathbf{x} \in \mathcal{V}^m$ telle que $\mathcal{V}(\psi) = \{\mathbf{x}\}$ et $F(\mathbf{y}) = \psi[\mathbf{x}/\mathbf{y}]$ pour tout $\mathbf{y} \in \mathcal{V}^m$. Pour chaque Σ -structure S , F définit la relation m -aire suivante F^{S^n} sur $dom(S)^n$:

$$F^{S^n} = \{(\alpha(\pi_1(\mathbf{x})), \dots, \alpha(\pi_m(\mathbf{x}))) \mid \alpha \in n\text{-sol}^S(\psi)\}$$

La formule $F(\mathbf{y})$ représente le fait que les valeurs de \mathbf{y} appartiennent à la relation définie par la formule ψ . Savoir précisément quelle séquence de variables distinctes \mathbf{y} est choisie n'a pas d'importance car $F(\mathbf{y}) = F(\mathbf{x})[\mathbf{x}/\mathbf{y}]$, étant donné que les solutions de $F(\mathbf{y})$ et $F(\mathbf{x})$ sur la structure S sont identiques exception faite du renommage de variables.

Lemme 38. Pour chaque définition du premier-ordre $F : \mathcal{V}^m \rightarrow \mathcal{F}_\Sigma^n$ et séquence $\mathbf{y} = y_1 \dots y_m$ de variables distinctes:

$$n\text{-sol}^S(F(\mathbf{y})) = \{[y_1/s_1, \dots, y_m/s_m] \mid (s_1, \dots, s_m) \in F^{S^n}\}.$$

Démonstration. Cela suit directement le Lemme. □

L'objectif maintenant est de généraliser la définition de l'application de fonctions de logique du premier-ordre aux fonctions d'arités supérieures et de prouver les propriétés formelles de telles définitions.

La notation vectorielle sera utilisée tout au long de ce chapitre. Fixons $\ell, k, n \in \mathbb{N}$ et considérons les définitions du premier-ordre $F : \mathcal{V}^\ell \times \mathcal{V}^k \rightarrow \mathcal{F}_\Sigma^n$ qui définissent une fonction partielle $F^{S^n} \subseteq dom(S^n)^\ell \times dom(S^n)^k$ pour la Σ -structure S

considérée. Pour chaque m , nous pouvons adapter la définition du premier-ordre F à une définition du premier-ordre $F^m : \mathcal{V}^{m\ell} \times \mathcal{V}^{mk} \rightarrow \mathcal{F}_\Sigma^n$ où F est appliquée m -fois, de telle sorte que pour toutes séquences $\mathbf{x}^1, \dots, \mathbf{x}^\ell, \mathbf{y}^1, \dots, \mathbf{y}^k \in \mathcal{V}^m$:

$$F^m(\mathbf{x}^1 \dots \mathbf{x}^\ell \mathbf{y}^1 \dots \mathbf{y}^k) =_{\text{def}} \bigwedge_{i=1}^m F(\mathbf{x}_i^1 \dots \mathbf{x}_i^\ell \mathbf{y}_i^1 \dots \mathbf{y}_i^k)$$

Pour chaque définition du premier-ordre $G : \mathcal{V}^{m\ell} \rightarrow \mathcal{F}_\Sigma^n$, nous introduisons une définition du premier-ordre $F^m(G) : \mathcal{V}^{mk} \rightarrow \mathcal{F}_\Sigma^n$ telle que pour tout $\bar{\mathbf{y}} \in \mathcal{V}^{mk}$:

$$F^m(G)(\bar{\mathbf{y}}) =_{\text{def}} \exists \bar{\mathbf{x}}. G(\bar{\mathbf{x}}) \wedge F^m(\bar{\mathbf{x}}, \bar{\mathbf{y}})$$

où $\bar{\mathbf{x}} = (\mathbf{x}^1 \dots \mathbf{x}^\ell) \in \mathcal{V}^{m\ell}$ est une séquence de variables fraîches. A noter que $\text{fv}(F^m(G)(\bar{\mathbf{y}})) = \{\bar{\mathbf{y}}\}$ et donc le choix de \mathbf{x} n'a pas d'importance.

Lemme 39. Soit $F : \mathcal{V}^{\ell+k} \rightarrow \mathcal{F}_\Sigma^n$ et $G : \mathcal{V}^{m\ell} \rightarrow \mathcal{F}_\Sigma^n$ des définitions du premier-ordre et S une Σ -structure S . Si la relation $F^{S^n} \subseteq \text{dom}(S^n)^{\ell+k}$ est une fonction partielle du type $\text{dom}(S^n)^\ell \times \text{dom}(S^n)^k$ alors la relation $(F^m)^{S^n}$ est une fonction partielle du type $\text{dom}(S^n)^{m\ell} \times \text{dom}(S^n)^{mk}$ telle que :

$$(F^m)^{S^n}(G^{S^n}) = F^m(G)^{S^n}$$

Démonstration. Soit $\bar{\mathbf{x}} = \mathbf{x}^1 \dots \mathbf{x}^\ell \in \mathcal{V}^{m\ell}$ et $\bar{\mathbf{y}} = \mathbf{y}^1 \dots \mathbf{y}^k \in \mathcal{V}^{mk}$ des séquences de variables telles qu'aucune variable n'apparaît deux fois dans $\bar{\mathbf{x}}\bar{\mathbf{y}}$. Alors:

$$\begin{aligned} (F^m)^{S^n}(G^{S^n}) &= \{(F^m)^{S^n}(\alpha(\bar{\mathbf{x}})) \mid \alpha \in n\text{-sol}^S(G(\bar{\mathbf{x}}))\} \\ &= \{\alpha'(\bar{\mathbf{y}}) \mid \alpha' \in n\text{-sol}^S(F^m(\bar{\mathbf{x}}\bar{\mathbf{y}})), \alpha'_{|\bar{\mathbf{x}}} \in n\text{-sol}^S(G(\bar{\mathbf{x}}))\} \\ &= \{\alpha'(\bar{\mathbf{y}}) \mid \alpha' \in n\text{-sol}^S(G(\bar{\mathbf{x}}) \wedge F^m(\bar{\mathbf{x}}\bar{\mathbf{y}}))\} \\ &= \{\alpha''(\bar{\mathbf{y}}) \mid \alpha'' \in n\text{-sol}^S(\exists \bar{\mathbf{x}}. G(\bar{\mathbf{x}}) \wedge F^m(\bar{\mathbf{x}}\bar{\mathbf{y}}))\} \\ &= F^m(G)^{S^n} \end{aligned}$$

Notons, pour la dernière étape précédente, que pour chaque $\alpha' \in n\text{-sol}^S(G(\bar{\mathbf{x}}) \wedge F^m(\bar{\mathbf{x}}\bar{\mathbf{y}}))$ nous pouvons choisir α'' comme la restriction $\alpha'_{|\mathcal{V} \setminus \{\bar{\mathbf{x}}\}}$. Inversement, pour chaque $\alpha'' \in n\text{-sol}^S(\exists \bar{\mathbf{x}}. G(\bar{\mathbf{x}}) \wedge F^m(\bar{\mathbf{x}}\bar{\mathbf{y}}))$ il doit exister une solution $\alpha' \in n\text{-sol}^S(G(\bar{\mathbf{x}}) \wedge F^m(\bar{\mathbf{x}}\bar{\mathbf{y}}))$ telle que α'' est la restriction $\alpha'_{|\mathcal{V} \setminus \{\bar{\mathbf{x}}\}}$. □

Pour le cas $k = 1$ et $\ell = 1$ le Lemme 39 mène à la conséquence suivante.

Proposition 40 ($(\ell = 1$ et $k = 1)$). Pour chaque définition du premier-ordre $G : \mathcal{V}^m \rightarrow \mathcal{F}_\Sigma^n$ et $F : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{F}_\Sigma^n$, séquence de variables $\mathbf{y} \in \mathcal{V}^m$ et Σ -structure S pour laquelle la relation F^{S^n} est une fonction partielle du type $\text{dom}(S^n) \times \text{dom}(S^n)$:

$$F^{S^n} \circ n\text{-sol}^S(G(\mathbf{y})) = n\text{-sol}^S(F^m(G)(\mathbf{y}))$$

Démonstration. A partir des Lemmes 38 et 39 :

$$\begin{array}{l|l}
F^{S^n} \circ n\text{-sol}^S(G(\mathbf{y})) & \\
\text{Lemme 38} & = F^{S^n} \circ \{[\mathbf{y}/\mathbf{s}] \mid \mathbf{s} \in G^{S^n}\} \\
& = \{[\mathbf{y}/\mathbf{s}] \mid \mathbf{s} \in (F^m)^{S^n}(G^{S^n})\} \\
\text{Lemme 39} & = \{[\mathbf{y}/\mathbf{s}] \mid \mathbf{s} \in F^m(G)^{S^n}\} \\
\text{Lemme 38} & = n\text{-sol}^S(F^m(G)(\mathbf{y}))
\end{array}$$

□

Pour le cas où ℓ est général et $k = 1$, la Proposition 40 peut être généralisée comme suit :

Proposition 41 ($(\ell \geq 1$ et $k = 1)$). Soit $G : \mathcal{V}^{m\ell} \rightarrow \mathcal{F}_\Sigma^n$ et $F : \mathcal{V}^\ell \times \mathcal{V} \rightarrow \mathcal{F}_\Sigma^n$ des définitions du premier-ordre, et S une Σ -structure telle que la relation F^{S^n} est une fonction partielle du type $\text{dom}(S^n)^\ell \times \text{dom}(S^n)$. Alors pour chaque $\mathbf{y} \in \mathcal{V}^m$ et générateur de variables fraîches v_1, \dots, v_ℓ , la séquence de variables $v(\mathbf{y}) = v_1(\mathbf{y}) \dots v_\ell(\mathbf{y})$ satisfait :

$$F^{S^n} \circ v^{-1}(n\text{-sol}^S(G(v(\mathbf{y})))) = n\text{-sol}^S(F^m(G)(\mathbf{y}))$$

où $v^{-1}(\alpha)(\mathbf{y}) = (\alpha(v_1(\mathbf{y})), \dots, \alpha(v_\ell(\mathbf{y})))$ pour tout $\mathbf{y} \in \mathcal{V}(\mathbf{y})$ et $\alpha : \cup_{i=1}^\ell \mathcal{V}(v_i(\mathbf{y})) \rightarrow \text{dom}(S^n)$.

Démonstration. De nouveau à partir des Lemmes 38 et 39, par généralisation et adaptation de la preuve de la Proposition 40 :

$$\begin{array}{l|l}
F^{S^n} \circ v^{-1}(n\text{-sol}^S(G(v(\mathbf{y})))) & \\
\text{Lemme 38} & = F^{S^n} \circ \{v^{-1}[\mathbf{y}/\mathbf{s}] \mid \mathbf{s} \in G^{S^n}\} \\
& = \{[\mathbf{y}/\mathbf{s}] \mid \mathbf{s} \in (F^m)^{S^n}(G^{S^n})\} \\
\text{Lemme 39} & = \{[\mathbf{y}/\mathbf{s}] \mid \mathbf{s} \in F^m(G)^{S^n}\} \\
\text{Lemme 38} & = n\text{-sol}^S(F^m(G)(\mathbf{y}))
\end{array}$$

□

Pour le cas général $k \geq 1$ et $l = 1$, le Lemme 39 mène à la généralisation suivante de la Proposition 40:

Proposition 42 ($(\ell = 1$ et $k \geq 1)$). Pour chaque définition du premier-ordre $G : \mathcal{V}^m \rightarrow \mathcal{F}_\Sigma^n$ et $F : \mathcal{V} \times \mathcal{V}^k \rightarrow \mathcal{F}_\Sigma^n$ et chaque structure S telle que la relation F^{S^n} est une fonction partielle du type $\text{dom}(S^n) \times \text{dom}(S^n)^k$, et chaque séquence de variables fraîches $\mathbf{y}, \mathbf{y}^1, \dots, \mathbf{y}^k \in \mathcal{V}^m$:

$$(F^m)^{S^n} \circ n\text{-sol}^S(G(\mathbf{y})) = \{[\mathbf{y}/(\alpha(\mathbf{y}^1), \dots, \alpha(\mathbf{y}^k))] \mid \alpha \in n\text{-sol}^S(F^m(G)(\mathbf{y}^1, \dots, \mathbf{y}^k))\}$$

Démonstration. Cela est une autre généralisation de la preuve de la Proposition 40 :

$$\begin{aligned}
 & F^{S^n} \circ n\text{-sol}^S(G(\mathbf{y})) \\
 \text{Lemme 38} & \quad = F^{S^n} \circ \{[\mathbf{y}/\mathbf{s}] \mid \mathbf{s} \in G^{S^n}\} \\
 & = \{[\mathbf{y}/(\mathbf{s}^1, \dots, \mathbf{s}^k)] \mid (\mathbf{s}^1, \dots, \mathbf{s}^k) \in (F^m)^{S^n}(G^{S^n})\} \\
 \text{Lemme 39} & \quad = \{[\mathbf{y}/(\mathbf{s}^1, \dots, \mathbf{s}^k)] \mid (\mathbf{s}^1, \dots, \mathbf{s}^k) \in F^m(G)^{S^n}\} \\
 \text{Lemme 38} & \quad = \{[\mathbf{y}/(\alpha(\mathbf{y}^1), \dots, \alpha(\mathbf{y}^k))] \mid \alpha \in n\text{-sol}^S(F^m(G)(\mathbf{y}^1, \dots, \mathbf{y}^k))\}
 \end{aligned}$$

□

Réseaux de réactions avec cinétique partielle

Nous introduisons des systèmes de réactions chimiques avec information cinétique partielle, et discutons leur sémantique, qui est définie modulo similarité d'expressions cinétiques. Nous sommes particulièrement intéressés par la sémantique d'équilibre de flux, aussi appelée sémantique à l'état stable. Puis nous verrons un exemple concret de ce formalisme avec la présentation des modèles biologiques sur lesquels nous allons appliquer nos analyses.

Sommaire

4.1	Systèmes de réactions chimiques	80
4.1.1	Sans cinétique	80
4.1.2	Avec cinétique	84
4.1.3	Avec cinétique modulo similarité	88
4.2	Cinétique partielle des modificateurs	89
4.3	Réseaux avec cinétiques partielles: le langage de modélisation	90
4.4	Syntaxe graphique	92
4.5	Exemple du réseau de régulation <i>PIIv – Leu</i>	93
4.5.1	Modélisation biologique	93
4.5.2	Modélisation formelle	94
4.6	Exemple de réseau métabolique	96
4.7	Modes élémentaires	100

4.1 Systèmes de réactions chimiques

4.1.1 Sans cinétique

Soit \mathcal{S} un ensemble fini d'espèces qui peuvent représenter biologiquement une protéine, un enzyme, un gène, de l'ARN messager, ... Une solution chimique avec des espèces en \mathcal{S} est un multi-ensemble d'espèces:

$$s : \mathcal{S} \rightarrow \mathbb{N}$$

La multiplicité $s(A)$ est appelée le coefficient stœchiométrique de l'espèce A dans la solution s . L'ensemble des solutions constructibles avec les espèces \mathcal{S} est noté $Sol_{\mathcal{S}}$. Quand l'ensemble d'espèces est fixé dans le contexte, on dénotera le même ensemble par Sol . Pour des espèces différentes $A_1, \dots, A_n \in \mathcal{S}$ et des naturels n_1, \dots, n_n on utilisera la notation

$$s = n_1 A_1 + \dots + n_n A_n$$

pour désigner la solution s avec $s(A_i) = n_i$ pour tout $1 \leq i \leq n$, et $s(A) = 0$ pour toute autre espèce dans $A \in \mathcal{S} \setminus \{A_1, \dots, A_n\}$. La solution vide sera notée $\mathbf{0}$, où $\mathbf{0} : \mathcal{S} \rightarrow \mathbb{N}$ est tel que $\mathbf{0}(A) = 0$ pour tout $A \in \mathcal{S}$.

Définition 43. Soit $\mathcal{N}s$ un ensemble fini de noms de réactions.

- Une réaction (chimique) sans cinétique est un triplet $r = (n, s_1, s_2)$ composé d'un nom de la réaction $n \in \mathcal{N}s$, et de deux solutions chimiques $s_1, s_2 \in Sol$.
- Un réseau de réactions (chimiques) sans cinétique est un ensemble fini de réactions, tel que les noms des réactions sont tous différents.

Nous dénoterons par $nom_r = n$ le nom de la réaction et par $\mathbf{r}_n = r$ la réaction elle-même, que nous écrivons comme suit :

$$(n) : s_1 \rightarrow s_2$$

Puis, nous appellerons $react_r = s_1$ les réactants et $prod_r = s_2$ les produits de la réaction.

Notons que seules les réactions irréversibles sont considérées, c'est-à-dire celles qui ne peuvent transformer que s_1 en s_2 , et ne peuvent pas faire la transformation inverse. Une réaction réversible sera donc modélisée par deux réactions, une dans chaque sens.

Un réseau de réactions R peut être vu comme un graphe biparti avec des arêtes labellisées par des naturels non nuls. Ce graphe est comme un réseau de Petri, mais sans marquage de place (comme R manque de concentration ou de solution

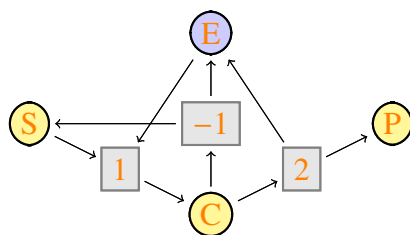


FIGURE 4.1 – Graphe du réseau de réactions de la dégradation enzymatique.



FIGURE 4.2 – Le réseau de la dégradation enzymatique.

FIGURE 4.3 – La matrice de stœchiométrie du réseau de la dégradation enzymatique.

initiale). Le graphe R contient des nœuds de deux types appelés *places* et *transitions*. Pour chaque espèce apparaissant dans R le graphe contient une place, et pour chaque réaction de R une transition. Nous dessinerons le nœud de type place, pour une espèce A , par un cercle contenant A et le nœud de type transition, pour une réaction $r_n \in R$, par une boîte contenant son nom n .

Pour toute réaction $r \in R$ et toute espèce A avec $react_r(A) > 0$ le graphe contient une arête à partir du nœud place de A vers le nœud transition de r . Cette arête est labellisée par $react_r(A)$. Symétriquement, pour toute réaction $r \in R$ et toute espèce A avec $prod_r(A) > 0$, le graphe contient une arête à partir du nœud transition de r vers le nœud place de A , qui est labellisée par $prod_r(A)$. Lorsque nous dessinerons des graphes de réseau de réactions, seuls les labels stœchiométriques différents de 1 seront représentés. Une arête sans label sera donc implicitement labellisée par 1.

Exemple 44. Un exemple classique est celui de la dégradation enzymatique. Ce réseau de réactions avec les espèces $S, E, P, C \in \mathcal{S}$ contient les 3 réactions avec noms $1, -1, 2 \in \mathcal{N}$ comme nous pouvons le voir avec la FIGURE 4.2. Ce réseau de réactions est représenté graphiquement par la FIGURE 4.1. Nous pouvons aussi voir l'ensemble de réactions qui décrit la transformation d'un substrat S en un produit

P , en présence d'une enzyme E , avec la formation d'un complexe intermédiaire C .

Biologiquement, une réaction peut représenter la liaison ou la désunion de deux molécules, le transport d'une molécule d'un compartiment vers un autre compartiment, la transcription d'ADN en ARN messenger, la traduction de l'ARN messenger en protéine, . . .

Nous appelons *la stœchiométrie d'une réaction r* la fonction $stoic_r : \mathcal{S} \rightarrow \mathbb{Z}$ définie par:

$$stoic_r(A) = prod_r(A) - react_r(A)$$

Le *coefficient stœchiométrique* d'une espèce A pour une réaction r donnée correspond à la valeur $stoic_r(A)$. Si sa valeur est positive, resp. négative, elle signifie que l'espèce A est produite, resp. consommée, par la réaction r .

Une espèce A avec $stoic_r(A) = 0$ a la même multiplicité dans les réactants et produits de la réaction r . A noter qu'une réaction $A + B \rightarrow A + C$ sera équivalente à $B \rightarrow C$ dans la sémantique stable. Il est donc possible mais pas obligatoire de faire apparaître les enzymes et répresseurs qui modifient la vitesse de la réaction dans les réactants et produits.

Soit R un ensemble de réactions. La collection $(stoic_r(A))_{A \in \mathcal{S}, r \in R}$ contient toute l'information de la *matrice de stœchiométrie* de R , qui est un élément $\mathbf{A} \in \mathbb{Z}^{n,m}$, où $n = |\mathcal{S}|$ et $m = |R|$. Les lignes de \mathbf{A} correspondent aux espèces dans \mathcal{S} et les colonnes aux réactions de R , les deux ont un ordre fixé arbitrairement. Si A est la i -ième espèce dans l'ordre, et r la j -ième réaction de R , alors :

$$\mathbf{A}_{ij} = stoic_r(A)$$

Exemple 45. Si l'on cherche à avoir la matrice de stœchiométrie du réseau sans cinétique représentant la dégradation enzymatique en FIGURE 4.1, qui a été vue avec l'exemple FIGURE 44, en fixant l'ordre $\mathbf{r}_1 < \mathbf{r}_{-1} < \mathbf{r}_2$ sur les réactions et $E < S < C < P$ sur les espèces, on obtient le résultat en FIGURE 4.3.

Nous donnons à un réseau de réactions chimiques R sans cinétique une sémantique stable – aussi appelée sémantique de l'équilibre de flux. Pour ceci, le réseau est vu comme une transformation de fonctions qui assigne des concentrations aux espèces.

Intuitivement, nous souhaitons assigner des vitesses aux réactions du réseau, telles que les vitesses de toutes les réactions se stabilisent. Nous nous intéressons à des vitesses positives puisque les réactions considérées sont irréversibles.

A chaque réseau R nous associons le système de Σ_{arith} -équations linéaires L'_R suivant, avec des variables fraîches $v_r \in \mathcal{V}$ pour chaque réaction $r \in R$, qui désignera sa vitesse:

$$L'_R = \bigwedge_{A \in \mathcal{S}} 0 \doteq \sum_{r \in R} stoic_r(A) v_r$$

L'expression à droite désigne la vitesse totale du changement de la concentration de l'espèce A en appliquant en même temps toutes les réactions de $r \in R$. Si cette vitesse totale pour A vaut zéro, la concentration de A restera stable quand on appliquera toutes les réactions de R simultanément.

Définition 46. Soit $\{v_{nom_r} \mid r \in R\} \subseteq V \subseteq \mathcal{V}$. Un assignement positif $\alpha : V \rightarrow \mathbb{R}_+$ est dit stable pour R , si α est une solution de L'_R dans la structure \mathbb{R} , c'est-à-dire, si $\alpha \in \text{sol}^{\mathbb{R}}(L'_R)$.

Le système d'équations L'_R peut être réécrit d'une manière \mathbb{R} -équivalente dans une équation matricielle : $\mathbf{Ax} \doteq 0$ tel que \mathbf{A} est la matrice de stœchiométrie de R et \mathbf{x} le vecteur des variables v_r pour les réactions $r \in R$ dans l'ordre fixé (pour définir la matrice de stœchiométrie).

Les coefficients $\text{stoic}_r(A)$ de la matrice \mathbf{A} et de L'_R peuvent être négatifs. De ce fait, nous devons utiliser la signature arithmétique complète Σ_{arith} pour écrire les équations de L'_R , et ne pouvons donc pas interpréter ces équations sur toute $\Sigma_{pos-arith}$ -structure \mathbb{R}_+ . Pour éviter ceci, nous pouvons reformuler le système d'équations L'_R comme un système L_R de $\Sigma_{pos-arith}$ -équations en évitant les coefficients négatifs comme suit:

$$L_R = \bigwedge_{A \in \mathcal{S}} \sum_{r \in R} \text{react}_r(A)v_{nom_r} \doteq \sum_{r \in R} \text{prod}_r(A)v_{nom_r}$$

Définition 47. La sémantique stable d'un réseau de réactions R sans cinétique est $\text{sol}^{\mathbb{R}_+}(L_R)$, c'est-à-dire, l'ensemble des assignements positifs, qui sont stables pour R .

Exemple 48. Soit R le réseau de réactions représenté graphiquement en FIGURE 4.5. L'ensemble de ses espèces est $\mathcal{S} = \{A, B, C\}$, et l'ensemble de ses réactions $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \mathbf{r}_5\}$. On peut ainsi inférer les systèmes d'équations linéaires équivalentes $L'_R \in \mathcal{F}_{\Sigma_{arith}}$ et $L_R \in \mathcal{F}_{\Sigma_{pos-arith}}$:

$$L'_R = \begin{cases} v_3 - 2v_1 \doteq 0 \\ \wedge v_4 - v_2 \doteq 0 \\ \wedge v_1 + v_2 - v_5 \doteq 0 \end{cases} \quad L_R = \begin{cases} 2v_1 \doteq v_3 \\ \wedge v_2 \doteq v_4 \\ \wedge v_5 \doteq v_1 + v_2 \end{cases}$$

Puis, on peut réécrire L'_R encore d'une manière équivalente comme l'équation matricielle avec des entiers :

$$\begin{pmatrix} -2 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix} \doteq 0$$

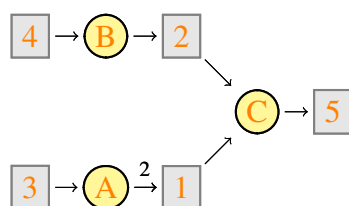
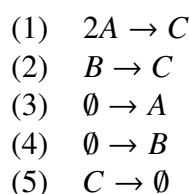


FIGURE 4.4 – Représentation graphique d'un réseau de réactions simple.



$$\begin{array}{c}
 \mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{r}_4 \quad \mathbf{r}_5 \\
 A \quad \left(\begin{array}{ccccc} -2 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & -1 \end{array} \right) \\
 B \\
 C
 \end{array}$$

FIGURE 4.5 – Le réseau de réactions de l'exemple simple.

FIGURE 4.6 – La matrice de stœchiométrie du réseau de l'exemple simple.

Ici, nous multiplions la matrice de stœchiométrie du réseau donnée en FIGURE 4.6 avec le vecteur de variable pour les réactions dans l'ordre utilisé dans la matrice de stœchiométrie.

Alternativement, on peut donner aux réseaux de réactions chimiques sans cinétique une sémantique non déterministe (Madelaine, 2017 ; Madelaine et al., 2014), qui ne sera pas définie formellement dans cette thèse. L'idée est de voir chaque réaction chimique comme une règle de réécriture de solutions chimiques. Étant donné une solution chimique initiale, cette règle de réécriture définit un graphe de solutions chimiques, obtenu par l'application de règles de réécriture d'une manière soit synchrone, soit asynchrone. On peut ensuite s'intéresser aux attracteurs de tels graphes pour savoir s'il est possible ou obligatoire d'y accéder à partir de la solution chimique initiale.

4.1.2 Avec cinétique

Pour pouvoir contraindre les vitesses des réactions dans un réseau R , nous devons ajouter des cinétiques aux réactions. Celles-ci sont des expressions arithmétiques, avec des variables pour les concentrations des espèces. Pour ceci, nous utiliserons les espèces elles mêmes comme variables pour la concentration, et puis

les variables pour les vitesses des réactions de R :

$$\mathcal{V} \supseteq \mathcal{S} \uplus \{v_{nom_r} \mid r \in R\}$$

Une expression cinétique est une expression arithmétique $e \in \mathcal{E}_{\Sigma_{arith}}$, telle que pour tout assignement $\alpha : \mathcal{S} \rightarrow \mathbb{R}_+$:

- la valeur de e est bien définie en évitant toute division par zéro, ainsi $\llbracket e \rrbracket^{\alpha, \mathbb{R}}$ est un singleton, et
- la valeur de e est positive, c'est-à-dire, $\llbracket e \rrbracket^{\alpha, \mathbb{R}} \subseteq \mathbb{R}_+$,

Une *expression cinétique* comme $k * A * B$ où $k = 912/10$ exprime une loi d'action de masse pour une réaction avec réactants $A, B \in \mathcal{S}$ et un paramètre $k \in \mathcal{E}_{\Sigma_{arith}}$.

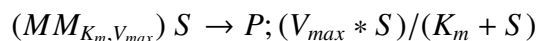
En effet, une expression cinétique possède un *paramètre cinétique*, noté k , qui est un descripteur de réel positif non nul. Tout paramètre est vu comme une expression $\mathcal{E}_{\Sigma_{arith}}$ formée par combinaison des opérateurs de la signature Σ_{arith} et de ses constantes 0 et 1. Il indique la probabilité qu'a une réaction de se produire. En biologie systémique, une des problématiques réside dans la détermination de la valeur de ces paramètres. Une fois interprété sur \mathbb{R}_+ il représente un réel, alors que sur Δ_6 il devient l'élément \approx . Son abstraction nous permet d'ignorer ce paramètre et donc de ne plus avoir besoin de connaître sa valeur. C'est la motivation principale de l'utilisation de ces abstractions.

Définition 49. — Une réaction avec cinétique est un quadruplet $r = (n, s_1, s_2, e)$ constitué d'une réaction sans cinétique (n, s_1, s_2) et d'une expression cinétique e .

- Un réseau de réactions avec cinétique est un ensemble fini R de réactions avec cinétique, tel que toutes les réactions de R ont un nom différent.

La réaction s'écrira $(n) s_1 \rightarrow s_2; e$, et son expression cinétique est $cin_r = e$.

Exemple 50. Considérons la réaction simple qui transforme une espèce S en une espèce P . En appliquant la loi de Michaelis-Menten à la réaction on obtient la réaction avec cinétique $\mathbf{r}_{MM_{K_m, V_{max}}}$:



où $V_{max} \in \mathcal{E}_{\Sigma_{arith}}$ est la vitesse maximale calculée en fonction de la concentration de S et $K_m \in \mathcal{E}_{\Sigma_{arith}}$ est la constante de Michaelis de l'enzyme qui catalyse la réaction, elles sont toutes les deux vues comme des paramètres.

Le graphe d'un réseau de réactions avec cinétique est défini comme pour un réseau sans cinétique, excepté que les nœuds pour les réactions sont annotés par l'expression cinétique de la réaction. Un exemple pour le graphe du réseau avec cinétique de la dégradation enzymatique est donné en FIGURE 4.7. Les lois cinétiques sont toutes des lois de masse-action.

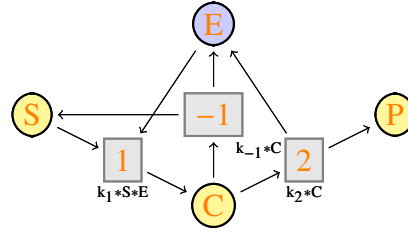
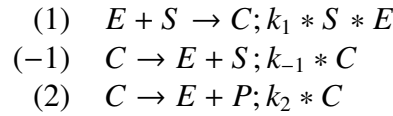


FIGURE 4.7 – Graphe du réseau de réactions avec cinétique de la dégradation enzymatique, avec $k_1, k_2, k_{-1} \in \mathcal{E}_{\Sigma_{arith}}$ les paramètres cinétiques respectifs des réactions $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_{-1}$.

Exemple 51. Si l'on reprend le réseau de la FIGURE 4.1, il serait possible de l'enrichir en y ajoutant une description des cinétiques de chaque réaction en appliquant la loi d'action de masse. Le réseau avec cinétique de la dégradation enzymatique deviendrait alors:



avec $k_1, k_{-1}, k_2 \in \mathcal{E}_{\Sigma_{arith}}$ des paramètres respectifs des réactions $\mathbf{r}_1, \mathbf{r}_{-1}$ et \mathbf{r}_2 .

La sémantique stable d'un réseau de réactions chimiques avec cinétique R est obtenue en enrichissant le système d'équations linaires L_R . Chaque réaction $r \in R$ donne ainsi l'équation cinétique :

$$v_{nom_r} \doteq cin_r$$

L'ensemble de toutes les équations de réseau avec cinétique R est le système d'équations arithmétiques E_R défini comme suit :

$$E_R = L_R \wedge C_R \quad \text{où} \quad C_R = \bigwedge_{r \in R} v_{nom_r} \doteq cin_r$$

Pour toute expression $e \in \mathcal{E}_{\Sigma_{arith}}$, on définit $e \geq 0$ la formule existentielle $\exists y. e \doteq y * y$ dans $\mathcal{F}_{\Sigma_{arith}}$.

Définition 52. La sémantique stable des réseaux de réactions avec cinétique R est $sol^{\mathbb{R}}(E_R \wedge \bigwedge_{x \in fv(E_R)} x \geq 0)$.

Ceci est l'ensemble des solutions réelles positives du système d'équations arithmétiques E_R . Nous montrons maintenant que des $\Sigma_{pos-arith}$ -équations interprétées sur \mathbb{R}_+ sont suffisantes pour exprimer cet ensemble de solutions. Dans le cas des lois de masse-action, cela provient déjà du résultat suivant:

Lemme 53. Si R est un réseau de réactions avec cinétique tel que $\text{cin}_r \in \mathcal{E}_{\Sigma_{\text{pos-arith}}}$ pour tout $r \in R$, alors :

$$\text{sol}^{\mathbb{R}}(C_R \wedge \bigwedge_{x \in \text{fv}(C_R)} x \geq 0) = \text{sol}^{\mathbb{R}_+}(C_R)$$

Démonstration. Il n'y a pas de différence dans l'interprétation sur les deux structures \mathbb{R} et \mathbb{R}_+ pour des $\Sigma_{\text{pos-arith}}$ -équations et des assignements de variables vers \mathbb{R}_+ . \square

Autrement dit, comme pour les lois du type *Michaelis-Menten*, le système d'équations C_R peut être réécrit dans un système avec signature $\Sigma_{\text{pos-arith}}$.

Lemme 54. Pour chaque réseau de réactions avec cinétique, le système de Σ_{arith} -équations C_R peut être réécrit en temps linéaire dans un système d'équations avec quantificateurs existentiels $C'_R \in \mathcal{F}_{\Sigma_{\text{pos-arith}}}$ tel que :

$$\text{sol}^{\mathbb{R}}(C_R \wedge \bigwedge_{x \in \text{fv}(C_R)} x \geq 0) = \text{sol}^{\mathbb{R}_+}(C'_R)$$

Démonstration. La Σ_{arith} -formule $e - e' \doteq e''$ peut se redéfinir $\exists z. z \doteq e - e' \wedge z \doteq e''$. On peut ainsi se séparer de l'opérateur de soustraction, avec la $\Sigma_{\text{pos-arith}}$ -formule équivalente égale à $\exists z. z + e' \doteq e \wedge z \doteq e''$. Dans la même idée, la Σ_{arith} -formule $e/e' \doteq e''$ est équivalente à la $\Sigma_{\text{pos-arith}}$ -formule $\exists z. z * e' \doteq e \wedge z \doteq e''$. On définit un système de $\Sigma_{\text{pos-arith}}$ -équations C'_R en appliquant ce changement à chaque Σ_{arith} -formule de C_R . On a donc $\text{sol}^{\mathbb{R}}(C_R) = \text{sol}^{\mathbb{R}}(C'_R)$. Pour finir d'après le Lemme 53 on a : $\text{sol}^{\mathbb{R}}(C'_R) \wedge \bigwedge_{x \in \text{fv}(C'_R)} x \geq 0 = \text{sol}^{\mathbb{R}_+}(C'_R)$ \square

Alternativement, on considère souvent la sémantique déterministe d'un réseau de réactions avec cinétique qui décrit l'évolution dynamique des concentrations d'espèce dans le temps. Mais pour nous, les vitesses des réactions et les concentrations sont des fonctions allant du temps vers des réels positifs, donc des fonctions du type $\mathbb{R}_+ \rightarrow \mathbb{R}_+$. Ces fonctions sont contraintes par des *systèmes d'équations différentielles ordinaires (ODEs)*, qui sont des Σ_{deriv} -équations ODE_R . Syntactiquement, ces Σ_{deriv} -équations contiennent les équations arithmétiques linéaires L'_R avec signature Σ_{arith} , mais sémantiquement, l'interprétation change de \mathbb{R} à $\mathbb{R}_+ \rightarrow \mathbb{R}$ pour ajouter l'axe du temps.

$$ODE_R = \bigwedge_{A \in \mathcal{S}} \overset{\circ}{A} \doteq \sum_{r \in R} \text{stoic}_r(A) v_{\text{nom}_r} \wedge \bigwedge_{r \in R} v_{\text{nom}_r} \doteq \text{cin}_r$$

Il est facile de voir que E_R a les mêmes solutions dans la structure $\mathbb{R}_+ \rightarrow \mathbb{R}$ que $ODE_R \wedge \bigwedge_{A \in \mathcal{S}} \overset{\circ}{A} \doteq 0$.

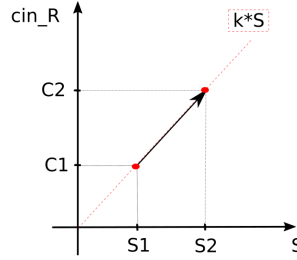
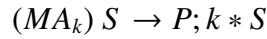


FIGURE 4.8 – Graphique représentant l'évolution de la cinétique de la réaction \mathbf{r}_{MA_k} entre deux états stationnaires différents en fonction de la concentration de l'espèce réactante S

Définition 55. La sémantique déterministe de R avec cinétique est $\text{sol}^{\mathbb{R}_+ \rightarrow \mathbb{R}}(\text{ODE}_R)$, c'est-à-dire, l'ensemble de solutions du système d'équations dérivées ordinaires de R dans la Σ_{deriv} -structure des fonctions réelles du type $\mathbb{R}_+ \rightarrow \mathbb{R}$.

Exemple 56. Si l'on reprend la réaction simple de l'exemple 50 qui transforme une espèce S en une espèce P . En appliquant la loi d'action de masse cette fois-ci, nous obtenons la réaction avec cinétique \mathbf{r}_{MA_k} :



avec $k \in \mathcal{E}_{\Sigma_{\text{arith}}}$ le paramètre cinétique de la réaction.

Le graphe 4.8 nous permet de représenter l'évolution de la vitesse de la réaction \mathbf{r}_{MA_k} entre deux états stables qui fixent la valeur de concentration d'espèce S , que l'on note S_1 et S_2 . Si l'on compare la valeur des vitesses à l'état stable 1 et à l'état stable 2, on voit que sa valeur augmente puisque la différence entre S_1 et S_2 augmente. Ceci est un exemple d'application de l'abstraction h_{Δ_6} de différences des états stables. Ici, si S vaut \uparrow (qui représente une augmentation) alors la valeur abstraite de sa cinétique $k *^{\Delta_6} S$ vaudra \uparrow également. On voit que le paramètre, qui est forcément strictement positif, ne change pas la monotonie de la courbe évolutive, il peut donc être ignoré lors de l'abstraction des différences. Et c'est, en effet, ce qu'il se passe en pratique étant donné que toute abstraction de paramètres vaut \sim , il n'ajoutent aucune contrainte au système.

4.1.3 Avec cinétique modulo similarité

Deux expressions cinétiques de réaction peuvent être interprétées modulo similarité relativement à un domaine abstrait $\Delta \in \{\Delta_3, \Delta_6\}$. Ainsi deux expressions $e, e' \in \mathcal{E}_{\Sigma_{\text{pos-arith}}}$ sont dites similaires modulo Δ ssi :

$$e \sim e' \Leftrightarrow \forall \alpha : f_V(e) \cup f_V(e') \rightarrow \text{dom}(\Delta). \llbracket e \rrbracket^{\alpha, \Delta} = \llbracket e' \rrbracket^{\alpha, \Delta}$$

De même, un réseau R de réactions avec cinétique peut être vu comme un réseau de réactions avec information cinétique partielle, qui décrit tout réseau R' similaire, c'est-à-dire, pour tout n, s_1, s_2, e, e' :

$$\begin{aligned} (n, s_1, s_2, e') \in R' &\Rightarrow \exists e. (n, s_1, s_2, e) \in R \wedge e \sim e' \\ (n, s_1, s_2, e) \in R &\Rightarrow \exists e'. (n, s_1, s_2, e') \in R' \wedge e \sim e' \end{aligned}$$

Deux réactions similaires contiennent donc les mêmes réactions si on enlève les expressions cinétiques. Ces deux expressions cinétiques doivent être similaires modulo Δ , et non nécessairement égales.

Pour des réseaux de réactions R avec cinétique, qui sont à interpréter modulo similarité, nous définissons l'abstraction des différences sous contraintes cinétiques comme suit, où $V = fV(E_R)$ est l'ensemble des variables de R :

$$D_R = \{\beta \mid \beta \in h_\Delta \circ \text{sol}_V^{\mathbb{R}_+^2}(L_R) \cap \text{sol}_V^\Delta(\bigwedge_{r \in R} v_{nom_r} \doteq cin_r)\}$$

Tout réseau R' similaire a nécessairement la même abstraction de différence sous contraintes cinétiques, puisque les équations arithmétiques L_R ne dépendent pas des cinétiques. Et que, $\text{sol}_V^\Delta(v_{nom_r} \doteq cin_r)$ ne change pas si on remplace cin_r par une expression similaire modulo Δ .

4.2 Cinétique partielle des modificateurs

En modélisation biologique, on connaît souvent les enzymes, qui activent et accélèrent une réaction et les inhibiteurs qui ralentissent une réaction. Par contre, on connaît plus rarement sa cinétique précise.

Il est argumenté dans (John et al., 2013a), que les expressions cinétiques des modificateurs peuvent toujours être spécifiées modulo similarité comme ci-dessous. Pour rappeler ceci, nous considérons pour chaque solution chimique s des expressions avec signature $\mathcal{E}_{\Sigma_{pos-arith}}$, qui modélisent les cinétiques possibles dans le cas où s contient des activateurs, des accélérateurs, ou des inhibiteurs respectivement.

$$\begin{aligned} Act(s) &= \prod_{A \in \mathcal{S}} A^{s(A)} \\ Acc(s) &= \prod_{A \in \mathcal{S}} (1 + A)^{s(A)} \\ Inh(s) &= inh(\sum_{A \in \mathcal{S}} s(A)A) \end{aligned}$$

Notons que celles-ci sont des expressions avec signature $\Sigma_{pos-arith}$, comme nous y avons permis le symbole inh . Cela évite tout problème posé par la division par zéro pour l'opérateur de division (qui n'est pas admis en $\Sigma_{pos-arith}$). Nous définissons la cinétique d'expression avec activateurs s_1 , accélérateurs s_2 et inhibiteurs

s_3 :

$$Mod(s_1, s_2, s_3) = Acc(s_1) * Act(s_2) * Inh(s_3)$$

Notons que les valeurs des expressions $Act(\mathbf{0})$, $Act(\mathbf{0})$ et $Inh(\mathbf{0})$, interprétées sur \mathbb{R} ou \mathbb{R}_+ , valent toujours 1, il en va donc de même pour $Mod(\mathbf{0}, \mathbf{0}, \mathbf{0})$.

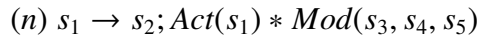
4.3 Réseaux avec cinétiques partielles: le langage de modélisation

Un langage de modélisation pour des réseaux de réactions avec information cinétique partielle a été proposé dans (John et al., 2013a), et nous allons l'évoquer ci-dessous.

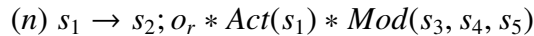
L'objectif est de spécifier pour chaque réaction d'un réseau, ses réactants, produits, activateurs, accélérateurs et inhibiteurs, ainsi qu'une expression cinétique pour capter la dépendance des réactions.

Définition 57. Une réaction avec cinétique partielle est un sixuplet $r = (n, s_1, s_2, s_3, s_4, s_5)$ tel que (n, s_1, s_2) est une réaction sans cinétique et $s_3, s_4, s_5 \in Sol$ des solutions chimiques.

Nous appellerons $act_r = s_3$ la solution des activateurs, $acc_r = s_4$ la solution des accélérateurs, $inhi_r = s_5$ la solution des inhibiteurs. Une réaction avec cinétique partielle, qui ne sera pas candidate de *knock-out*, décrit la réaction chimique avec cinétique modulo similarité suivante :



Pour modéliser le fait qu'une réaction r soit candidate pour un *knock-out*, nous considérons des nouvelles variables booléennes $o_r \in \mathcal{V}$ pour chaque réaction $r \in R$. Cette variable sera utilisée dans la cinétique de la réaction comme suit :

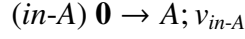


Si $o_r \doteq 0$ alors la réaction r ne sera pas active, elle aura ainsi une vitesse $v_{nom_r} \doteq 0$. Si par contre $o_r \doteq 1$, la réaction r sera activée avec la même vitesse qu'une réaction qui n'est pas candidate pour un *knock-out* $v_{nom_r} \doteq Act(s_1) * Mod(s_3, s_4, s_5)$.

Nous nous intéressons aux réseaux de réactions avec des flux d'entrée et de sortie. Ceux-ci seront utiles, en modélisation, pour connecter un sous-réseau du métabolisme avec d'autres, ou encore pour décrire la dépendance du réseau à son contexte, autrement dit son milieu d'expérimentation en laboratoire.

Pour cette raison, nous pouvons considérer pour chaque espèce $A \in \mathcal{S}$ une nouvelle variable x_A représentant la vitesse du flux d'entrée de A et une variable

y_A pour la vitesse du flux de sortie de A . Le flux d'entrée sera contrôlé par l'environnement. Pour représenter cela, sa vitesse ne sera pas contrainte et on aura :



L'équation pour la vitesse de chaque flux d'entrée est donc $v_{in-A} \doteq v_{in-A}$, ce qui n'impose aucune contrainte. Les flux de sortie seront quant à eux contrôlés par le réseau lui-même, en suivant une cinétique similaire à la loi de masse d'action, c'est-à-dire :



La vitesse du flux de sortie satisfera donc $y_A \doteq A$ modulo similarité. En résumé, nous considérons l'ensemble de variables suivant :

$$\mathcal{V} = \{v_{nom_r}, o_{nom_r} \mid r \in R\} \cup \{A, v_{in-A}, v_{out-A} \mid A \in \mathcal{S}\}$$

Définition 58. *Un réseau de réactions avec cinétique partielle est un quadruple $P = (R, I, O, K)$, tel que R est un ensemble fini de réactions avec cinétique partielle, chacune avec un nom différent, $I, O \subseteq \mathcal{S}$ sont des sous-ensembles finis d'espèces, et les éléments de $K \subseteq R$ sont des réactions candidates pour un knock-out.*

Les éléments de $K \subseteq R$ sont les réactions qui sont des candidates de *knock-out*. Le réseau avec cinétique partielle P définit un réseau avec cinétique modulo similarité unique :

$$\begin{aligned} sim(P) = & \{(nom_r) react_r \rightarrow prod_r; Act(react_r) * Mod(act_r, acc_r, inhi_r) \mid r \in R \setminus K\} \\ & \cup \{(nom_r) react_r \rightarrow prod_r; o_r * Act(react_r) * Mod(act_r, acc_r, inhi_r) \mid r \in K\} \\ & \cup \{(in-A) \mathbf{0} \rightarrow A; v_{in-A} \mid A \in I\} \\ & \cup \{(out-A) A \rightarrow \mathbf{0}; A \mid A \in O\} \end{aligned}$$

L'ensemble de toutes les variables d'un réseau P de réactions avec cinétique partielle est :

$$fv(P) = fv(L_{sim(P)}) \cup fv(C_{sim(P)})$$



Les abstractions de différences pour le réseau de réactions avec information cinétique partielle P doivent donc appartenir à l'ensemble d'assignements suivant où $V = fv(P)$:


$$D_P = h_\Delta \circ sol_V^{\mathbb{R}_+^2}(L_{sim(P)}) \cap sol_V^{\Delta_6}(C_{sim(P)} \wedge \bigwedge_{r \in K} o_r \in \{\Downarrow, \sim\})$$

Ici, la formule $o_r \in \{\Downarrow, \sim\}$ est un raccourci pour la disjonction $o_r \doteq \Downarrow \vee o_r \doteq \sim$. Cela exprime le fait que la réaction r est candidate pour un *knock-out*, par conséquent la variable o_r peut soit être réduite à 0 en cas de *knock-out*, soit rester à 1 lorsque la réaction n'est pas altérée. La signature de cette formule est $\Sigma_{pos-arith}[dom(\Delta_6)]$.

4.4 Syntaxe graphique


Afin de représenter l'ensemble des variables d'un réseau de réactions avec cinétique partielle, des règles de syntaxe graphique ont été mises en place. Ainsi on utilisera les notations suivantes:

Une espèce $A \in \mathcal{S}$ est représentée par un rond . Une réaction r_n est représentée par une boîte grise . Dans le cas où la réaction est candidate pour un *knock-out*,

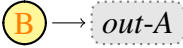
elle est représentée par une boîte orange claire . Comme il a été expliqué auparavant, concernant une réaction r_n , on va pour chaque $A \in \mathcal{S}$ réactant de cette réaction dessiner une flèche orientée de A vers r_n et labellisée par $react_r(S)$,; et pour chaque espèce $P \in \mathcal{S}$ produit de la réaction, une flèche labellisée par $prod_r(P)$ allant de r_n vers P sera représentée. Pour rappel, le label n 'est présent que dans le cas où $react_r(S) > 1$ et respectivement $prod_r(P) > 1$. La réaction transformant le

réactant A en produit B sera donc représentée comme suit .

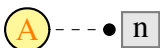
Le flux d'entrée $in-A$ pour une espèce $A \in \mathcal{S}$ est représenté par un rond gris en-

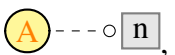

touré de pointillés et est relié à A par une flèche continue . Le flux

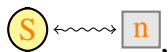
de sortie nommé $out-A$ pour une espèce $A \in \mathcal{S}$ est représenté par un rond gris

entouré de pointillés et est relié à v_{in-A} par une flèche continue .

Concernant les modificateurs, l'activation, l'accélération et l'inhibition de $A \in \mathcal{S}$

sur $r \in \mathcal{N}$ seront représentés respectivement par ,

, et . Dans la section suivante, nous pouvons voir avec la FIGURE 4.9 un exemple de représentation graphique d'un réseau de réactions

avec cinétique partielle. Une simplification sur les réactions notées , est la simplification de deux réactions celle qui produit S et celle qui la dégrade.

Nous utiliserons cette notation dorénavant, elle permet d'alléger le réseau en ne rendant pas explicite la réaction de dégradation qui sera nécessaire pour respecter l'équilibre du système notamment en ce qui concerne la régulation. En effet, la présence des gènes régulateurs reste en interne et ils ne peuvent pas être des entrées du système. Ils sont considérés présents en continu et seul un *knock-out* peut stopper leur activité. De ce fait, ils doivent tous avoir une réaction de production et une de dégradation pour être représentés correctement. C'est le cas par exemple de **CcpA** qui est produit et dégradé par la réaction r_{14} et respectivement r_{-14} dans le réseau de réactions représenté dans la FIGURE 4.9.

4.5 Exemple du réseau de régulation *Pilv – Leu*

Avant de montrer l'application de la formalisation que nous venons de mettre en place, nous allons voir le sens biologique du réseau de régulation *Pilv – Leu*.

4.5.1 Modélisation biologique

Reprenons l'explication faite pour la partie du métabolisme de la biosynthèse de la leucine chez la bactérie *B. subtilis*. Intéressons nous plus particulièrement à l'opéron *ilv – leu* qui encode des enzymes responsables de la production de leucine, à savoir que sa surproduction sera notre cible pour la prédiction. Cet opéron est régulé par le promoteur $P_{Ilv-Leu}$. Toute cette information est contenue dans la réaction r_8 .

La réaction r_2 , sous l'influence de BS_{CodY} , BS_{TnrA} et BS_{CcpA} qui représentent l'activité des sites de fixation de respectivement **CodY**, **TnrA** et **CcpA**, permet l'activation du promoteur $P_{Ilv-Leu}$ qui est responsable de la production de leucine. Les protéines **CodY** et **TnrA** sont les espèces d'entrée du réseau, leur ajout à partir du contexte est représenté par *in-CodY* et *in-TnrA*, et leur dégradation par respectivement r_{51} et r_{52} . Ces deux protéines jouent un rôle dans la régulation de la dégradation de la leucine par l'inhibition de la réaction r_2 . La protéine **CcpA** est exprimée et dégradée par la réaction r_{14} et respectivement r_{-14} .

Il est connu de la littérature, que la transcription au niveau du promoteur $P_{Ilv-Leu}$ est inhibée par **CodY** lors de sa fixation sur celui-ci. C'est pourquoi la réaction r_1 est introduite, elle active le site de liaison du promoteur pour **CodY**. Cela déclenche le ralentissement de la réaction r_2 et, de ce fait, ralentit l'activité du promoteur.

Cette liaison de **CodY** au site de liaison du promoteur peut être bloquée lorsque **CcpA** est lié au promoteur. C'est pourquoi il est modélisé que **CcpA** ralentit la réaction r_1 .

L'action du promoteur $P_{Ilv-Leu}$ est aussi réprimé par la leucine **Leu** au niveau de la boîte *TATA*. Cela est capturé par le contrôle négatif de **Leu** sur la réaction r_2 .

La protéine **TnrA**, tout comme **CodY**, réprime l'activité du promoteur en se fixant au site de liaison BS_{TnrA} avec la réaction r_7 .

Indépendamment, la protéine **CcpA** accentue la transcription de l'opéron *ilv – leu* en se fixant dessus, ce qui accélère la réaction r_2 . La fixation de **CcpA** est représentée par la réaction r_9 . De plus, l'accélération de la réaction r_2 par l'activation du site de liaison BS_{CcpA} est inhibée lorsque BS_{TnrA} est actif. Ces deux sites de liaison peuvent se lier l'un à l'autre en formant une boucle d'ADN. Ce phénomène est capturé par l'inhibition de la réaction r_9 par BS_{TnrA} .

4.5.2 Modélisation formelle

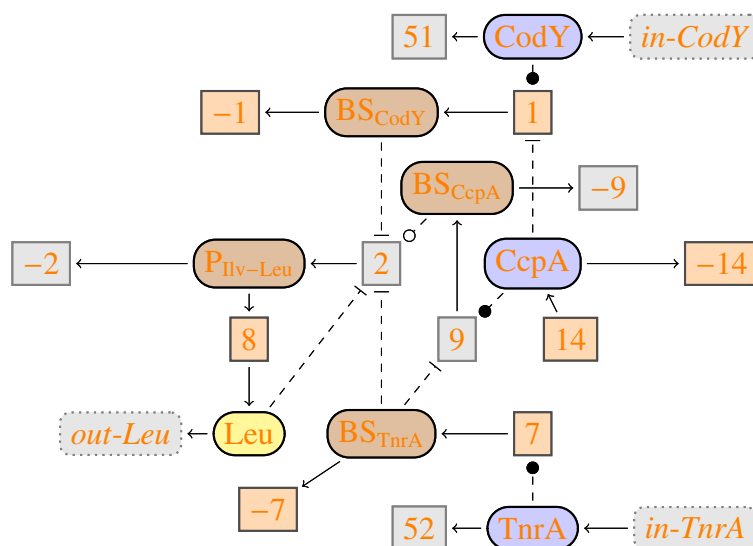


FIGURE 4.9 – Représentation graphique du réseau de réaction avec cinétique partielle $P_{Ilv-Leu}$ pour la régulation du promoteur $P_{Ilv-Leu}$.

- (1) $\mathbf{0} \rightarrow BS_{CodY}; o_1 * CodY * inh(CcpA)$
- (-1) $BS_{CodY} \rightarrow \mathbf{0}; BS_{CodY}$
- (2) $\mathbf{0} \rightarrow P_{Ilv-Leu}; inh(BS_{CodY} + Leu + BS_{TnrA}) * (1 + BS_{CcpA})$
- (-2) $P_{Ilv-Leu} \rightarrow \mathbf{0}; P_{Ilv-Leu}$
- (7) $\mathbf{0} \rightarrow BS_{TnrA}; o_7 * TnrA$
- (-7) $BS_{TnrA} \rightarrow \mathbf{0}; BS_{TnrA}$
- (9) $\mathbf{0} \rightarrow BS_{CcpA}; CcpA * inh(BS_{TnrA})$
- (-9) $BS_{CcpA} \rightarrow \mathbf{0}; BS_{CcpA}$
- (51) $CodY \rightarrow \mathbf{0}; CodY$
- (52) $TnrA \rightarrow \mathbf{0}; TnrA$
- (14) $\mathbf{0} \rightarrow CcpA; o_{14}$
- (-14) $CcpA \rightarrow \mathbf{0}; CcpA$
- (8) $P_{Ilv-Leu} \rightarrow Leu; P_{Ilv-Leu}$
- (*in-CodY*) $\mathbf{0} \rightarrow CodY; v_{in-CodY}$
- (*in-TnrA*) $\mathbf{0} \rightarrow TnrA; v_{in-TnrA}$
- (*out-Leu*) $Leu \rightarrow \mathbf{0}; Leu$

FIGURE 4.10 – Réseau de réactions $sim(P_{Ilv-Leu})$ avec cinétique modulo similarité. Dans cet exemple, pour modéliser la régulation du promoteur de *Ilv - Leu*

$$\begin{array}{ll}
v_1 \doteq o_1 * \text{CodY} * \text{inh}(\text{CcpA}) & v_{-1} \doteq \text{BS}_{\text{CodY}} \\
v_2 \doteq \text{inh}(\text{BS}_{\text{CodY}} + \text{Leu} + \text{BS}_{\text{TnrA}}) & v_{-2} \doteq \text{P}_{\text{Ilv-Leu}} \\
& *(1 + \text{BS}_{\text{CcpA}}) \\
v_{-7} \doteq \text{BS}_{\text{TnrA}} & v_7 \doteq o_7 * \text{TnrA} \\
v_{-9} \doteq \text{BS}_{\text{CcpA}} & v_9 \doteq \text{CcpA} * \text{inh}(\text{BS}_{\text{TnrA}}) \\
v_{52} \doteq \text{TnrA} & v_{51} \doteq \text{CodY} \\
v_{-14} \doteq \text{CcpA} & v_{14} \doteq o_{14} \\
v_{\text{in-CodY}} \doteq v_{\text{in-CodY}} & v_8 \doteq o_8 * \text{P}_{\text{Ilv-Leu}} \\
v_{\text{out-Leu}} \doteq \text{Leu} & v_{\text{in-TnrA}} \doteq v_{\text{in-TnrA}}
\end{array}$$

FIGURE 4.11 – Système de contraintes non linéaires de similarité cinétique $C_{P_{Ilv-Leu}}$: des $\Sigma_{\text{pos-arith}}[\text{dom}(\Delta_6)]$ -équations à interpréter sur Δ_6 .

$$\begin{array}{ll}
\text{Leu} : & v_{\text{out-Leu}} \doteq v_8 \\
\text{CodY} : & v_{51} \doteq v_{\text{in-CodY}} \\
\text{BS}_{\text{CodY}} : & v_{-1} \doteq v_1 \\
\text{BS}_{\text{TnrA}} : & v_{-7} \doteq v_7 \\
\text{CcpA} : & v_{-14} \doteq v_{14} \\
\text{TnrA} : & v_{52} \doteq v_{\text{in-TnrA}} \\
\text{P}_{\text{Ilv-Leu}} : & v_{-2} + v_8 \doteq v_2 \\
\text{BS}_{\text{CcpA}} : & v_{-9} \doteq v_9
\end{array}$$

FIGURE 4.12 – Système d'équations linéaires $L_{P_{Ilv-Leu}}$ de $\Sigma_{\text{pos-arith}}$ -équations à l'état stable, à interpréter sur \mathbb{R}_+ .

dans la bactérie *B. subtilis*, nous définissons le réseau de réactions avec cinétique partielle $P_{Ilv-Leu} = (R_{P_{Ilv-Leu}}, I_{P_{Ilv-Leu}}, O_{P_{Ilv-Leu}}, K_{P_{Ilv-Leu}})$ où :

$\mathcal{S} = \{\text{Leu}, \text{BS}_{\text{CodY}}, \text{P}_{\text{Ilv-Leu}}, \text{BS}_{\text{TnrA}}, \text{BS}_{\text{CcpA}}, \text{CodY}, \text{TnrA}, \text{CcpA}\}$ l'ensemble des espèces.

$R_{P_{Ilv-Leu}} = \{\mathbf{r}_1, \mathbf{r}_{-1}, \mathbf{r}_2, \mathbf{r}_{-2}, \mathbf{r}_7, \mathbf{r}_{-7}, \mathbf{r}_9, \mathbf{r}_{-9}, \mathbf{r}_{51}, \mathbf{r}_{52}, \mathbf{r}_{14}, \mathbf{r}_{-14}, \mathbf{r}_8\}$ est l'ensemble des réactions avec cinétique partielle données graphiquement dans la FIGURE 4.9.








$I_{P_{Ilv-Leu}} = \{\text{CodY}, \text{TnrA}\}$ est l'ensemble des espèces en entrée.

$O_{P_{Ilv-Leu}} = \{\text{Leu}\}$ est l'ensemble des espèces en sortie.

$K_{P_{Ilv-Leu}} = \{\mathbf{r}_1, \mathbf{r}_{14}, \mathbf{r}_7, \mathbf{r}_8\}$ est l'ensemble de réactions candidates pour le *knock-out*.

Nous pouvons voir dans FIGURE 4.10 la définition du modèle sous la forme d'un réseau de réactions $\text{sim}(P_{Ilv-Leu})$ avec cinétique modulo similarité. Et sa représentation graphique est donnée par la FIGURE 4.9. En appliquant le formalisme mis en place nous obtenons donc le système de contraintes non linéaires de similarité cinétique $C_{P_{Ilv-Leu}}$, avec la FIGURE 4.11, et le système d'équations linéaires $L_{P_{Ilv-Leu}}$ de $\Sigma_{\text{pos-arith}}$ -équations à l'état stable avec la FIGURE 4.12.

4.6 Exemple de réseau métabolique

La syntaxe graphique a été enrichie de quelques notations permettant de garder les modèles lisibles et d'ajouter des informations. Tout d'abord, nous pouvons voir que les espèces possèdent une couleur. Cette couleur dépend du type de représentation que l'on souhaite faire. En jaune on retrouve les métabolites, en bleu les protéines et en marron les espèces dites artificielles qui représentent des complexes, comme par exemple le promoteur d'un opéron ou un site de liaison. On aura donc les notations respectives suivantes: , , et . De plus, la production de certaines espèces dans un réseau est essentielle, leur absence entraînant un dysfonctionnement dans le système biologique représenté. On peut représenter cette contrainte de façon graphique en entourant l'espèce concernée de rouge comme suit . Il est aussi possible d'intégrer des espèces artificielles (des *clusters de protéines*) qui représentent un groupe de protéines participant à une même réaction. Par exemple, avec , on représente toutes les protéines générées par l'opéron **BkL**. Et pour garder les graphes lisibles même avec les réseaux de grande taille, nous utilisons des *split-points* qui relient différentes copies de la même espèce. Par exemple, nous voyons que l'espèce  est relié à sa copie . Il n'y a pas de différence entre l'espèce et sa copie, cela allège juste la lecture en évitant d'entrecroiser trop d'arcs. Il est également possible de créer des *edgeclusters*, qui sont des points permettant de relier les arcs partageant le même effet. A titre d'exemple, les arcs d'activation de **Bcd** et **YwaA** sur les réactions r_{29} , r_{32} et r_{35} sont partagés à l'aide d'un edgecluster. Avec l'utilisation de ces différents sucres syntactiques, la lecture s'en trouve facilitée.

Ce formalisme a été mis en place pour permettre l'étude et la modélisation de réseaux bien plus complexes et plus grands. La figure suivante nous donne la représentation graphique du réseau de réactions représentant une partie du métabolisme de la bactérie *B. subtilis* avec sa régulation. Il s'agit de la voie de synthèse des acides aminés branchés : valine, leucine et isoleucine.

Sa description a été faite dans les travaux précédents, nous allons en reprendre la traduction ici.

Production de Leucine, Valine et Isoleucine.

Si nous regardons uniquement les espèces intermédiaires qui produisent les sorties, à savoir **Ket_a** pour **Ile** par la réaction r_{26} , **Ket_b** pour **Val** par la réaction r_{36} , et **Ket_c** pour **Leu** par la réaction r_{45} . Ces trois réactions doivent être activées par **YwaA+YbgE**, ce qui revient à être activées soit par **YwaA** avec r_6 , soit **YbgE** avec r_5 . Il y a également les réactions inverses r_{29} , r_{32} et r_{35} avec des activateurs différents

: les protéines **Bcd** et **YwaA**. Les espèces intermédiaires **Ket_a**, **Ket_b**, **Ket_c** peuvent aussi être transformées en **Acyl-Coa** par les réactions **r₂₈**, **r₃₁** et **r₃₄**. **Acyl-Coa** peut ensuite sortir dans le contexte pour servir à la biosynthèse d'acides gras. Cela est régulé par **Bkl**. La création de **Leu**, **Val** et **Ile** à partir de **Ket_a**, **Ket_b**, **Ket_c** est toujours associée à une transformation parallèle de **Glu** en **OxoGlu**, et les réactions inverses sont couplées à la transformation inverse de **OxoGlu** en **Glu**. Le **Glu** entre par le cycle de biosynthèse du glutamate du contexte et **OxoGlu** peut lui sortir dans le contexte (il sera utile par exemple dans le cycle TCA). Les espèces intermédiaires, quant à elles, sont produites comme suit: **Ket_a** est produit à partir de **Akb** par **r₂₇**, **Ket_b** est produit à partir de **Pyr** par **r₃₀** et **Ket_c** à partir de **Ket_b** par **r₃₃**. **Akb** est produit à partir de **Thr** par **r₄₁**. Et **Thr** entre dans le contexte par la voie de biosynthèse de la thréonine (qui est liée à la voie glycolytique par la voie de biosynthèse de l'aspartate). **Pyr** peut lui entrer via la voie glycolytique dans le contexte. **IlvA** active la conversion de **Thr** en **Akb**. **IlvBH** (un cluster de **IlvB** et **IlvH**) **IlvC** et **IlvD** activent la transformation de **Akb** en **Ket_a**. Les protéines **LeuA** et **LeuBCD** (un cluster de **LeuB**, **LeuC** et **LeuD**) forment une succession de réactions justifiant la transformation de **Ket_b** en **Ket_c**. Les gènes des protéines **IlvBH**, **IlvC**, **LeuA** et **LeuBCD** sont localisés dans le même opéron, donc ils sont sous la dépendance du même promoteur **PilvLeu**.

Régulation.

Les expressions des différentes enzymes impliquées dans la production des acides aminés branchés, présentées juste au-dessus, sont caractérisées par les réactions suivantes : **r₂₂** pour **IlvA**; **r₂₅** pour **IlvD**; **r₂₃** pour **IlvBH**; **r₂₄** pour **IlvC**; **r₃₈** pour **LeuBCD**; **r₃₇** pour **LeuA**; **r₁₁** pour **Bkl**; **r₁₀** pour **BCD**; **r₄₇** pour **YwaA** et **r₄₆** pour **YbgE**. Toutes ces réactions d'expressions sont contrôlées par un complexe de régulation. En effet, nous considérons les protéines **CcpA**, **CodY**, **TnrA** et **BkdR** qui ont des fonctions de régulation et sont exprimées respectivement par les réactions **r₁₄**, **r₁₅**, **r₁₆**, **r₁₂** et **r₁₃**. Les réactions d'expression de ces régulateurs peuvent être liées à différents facteurs du contexte que nous ne modélisons pas ici. **Gtp** vient du contexte et peut être dégradé par la réaction **r₁₇**. Ce métabolite augmente l'affinité de **CodY** à son site de liaison, de même pour **Val** et **Ile**. L'expression de **IlvA**, **IlvD** et **YbgE** sont dépendants de leur propre promoteur et leur expression est sous-réglée par **CodY** au niveau de la transcription. Cela est illustré par les réactions **r₂₂**, **r₂₅** et **r₄₆** respectivement. **IlvA** est désactivé par **Ile** et **Val** avec **r₂₀** et **r₂₁**. La transcription des protéines **IlvC**, **IlvBH**, **LeuA** et **LeuBCD** commence avec l'activation de leur promoteur **PilvLeu** qui contrôle les réactions **r₂₃**, **r₂₄**, **r₃₇** et **r₃₈** respectivement. Sa régulation est représentée par la réaction **r₂** et est dépendante de plusieurs mécanismes décrits plus tard. De plus, il y a une sous-régulation de **PilvLeu** par **Leu**, et **Leu** désactive également **LeuA** par la réaction **r₃₉**. Les protéines **Bkl** et **Bcd** reposent sur l'action de deux promoteurs dont l'activité est

représentée par l'acteur **OBkLBcd**. Une partie n'est pas régulée via la réaction r_4 , et l'autre est impactée positivement par **BkdR** et sous-réglée par **TnrA** et **CodY** comme l'illustre la réaction r_3 . Bien que non régulée au niveau transcriptionnelle, la protéine **BkdR** est activée par **Ile** et **Val**, ce qui est modélisé par l'introduction de deux **BkdR** produisant les réactions r_{12} et r_{13} . La protéine **YbgE**, exprimée par r_{46} , est sous-réglée par **CodY**. Il n'y a pas de régulation connue pour **YwaA** qui est exprimée par r_{47} .

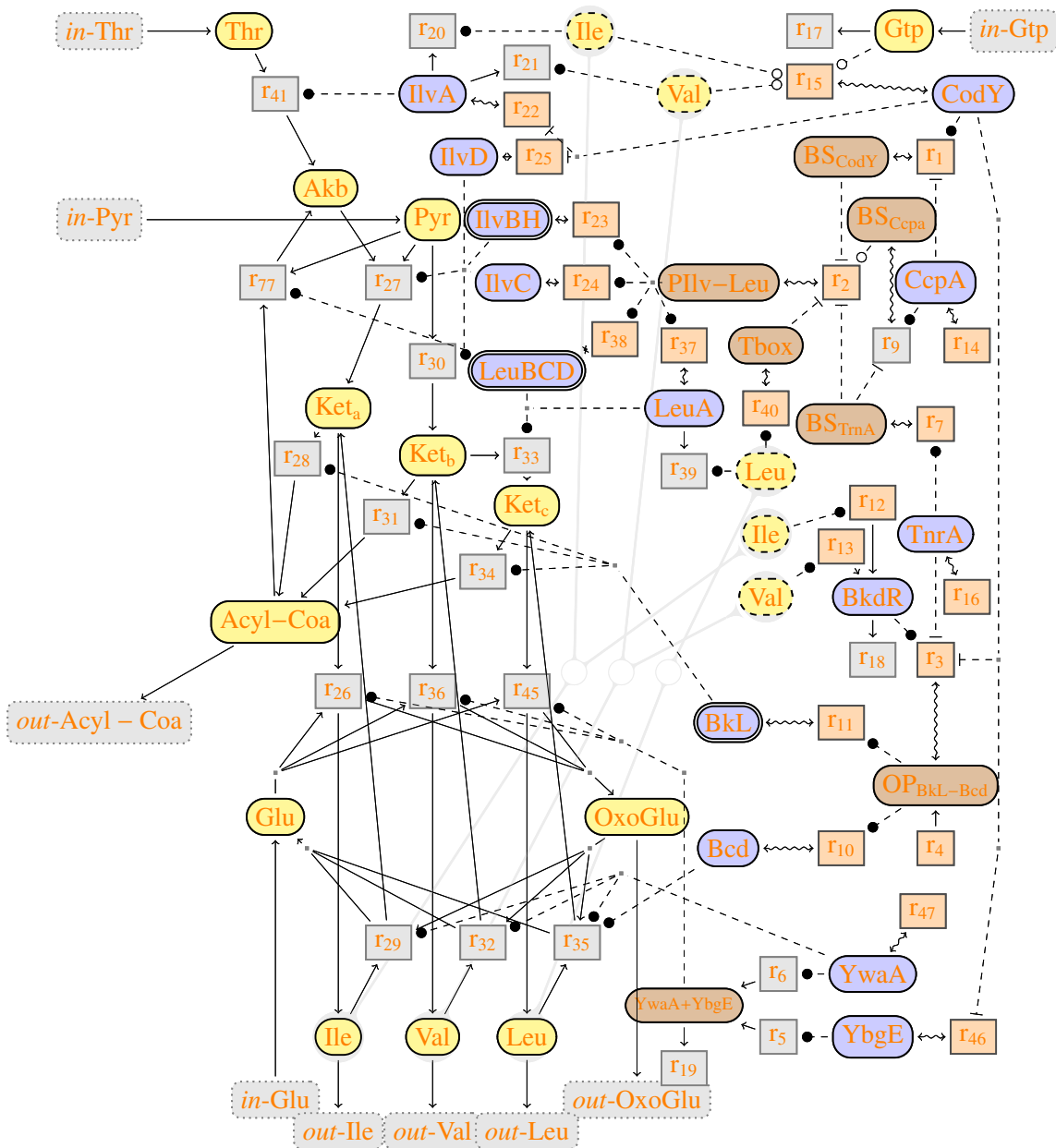


FIGURE 4.13 – Représentation graphique d’une partie du métabolisme de la bactérie *B. subtilis* avec sa régulation.

4.7 Modes élémentaires

A l'aide du calcul des modes élémentaires d'un réseau de réactions décrit formellement en section 2.10, on peut obtenir la description des flux métaboliques d'un réseau.

Comme nous avons pu le voir, en étudiant un réseau métabolique à l'état stationnaire on obtient une équation linéaire matricielle $\mathbf{A}\mathbf{y} \doteq 0$, où \mathbf{A} représente sa matrice de stœchiométrie et \mathbf{y} un vecteur de variables distinctes représentant ses espèces. Notre ensemble de réactions étant irréversibles nous cherchons à trouver l'ensemble de valeur des variables de \mathbf{y} satisfaisants le cône de flux :

$$CF = \{\mathbf{y} \in \mathbb{R}_+^n \mid \mathbf{A}\mathbf{y} = 0\}$$

Soit e_1, \dots, e_n l'ensemble des modes élémentaires normalisés dans un ordre total arbitraire du cône CF et E la matrice ayant pour colonne e_1, \dots, e_n . Par la Proposition 17, nous avons vu que cette équation linéaire matricielle inférée $\mathbf{A}\mathbf{y} \doteq 0$ est équivalente sur \mathbb{R}_+ au système $\exists \mathbf{x}. E\mathbf{x} \doteq \mathbf{ny}$.

En effet, chaque point dans le cône CF peut s'exprimer comme une combinaison linéaire non négative de ses \mathbb{R}_+ -EFMs.

Un mode élémentaire est le plus petit sous-ensemble de réactions qui se trouve à l'état stable ainsi toute espèce produite dans ce flux, sera consommée dans les mêmes proportions. Au cours de cette thèse, nous avons développé un module permettant la visualisation de ces flux dans la représentation graphique de nos réseaux de réactions faite par l'outil graphique existant. On peut, par exemple, voir avec la FIGURE 4.14 l'ensemble des flux métaboliques du réseau vu précédemment qui sont chacun représentés par une couleur différente.

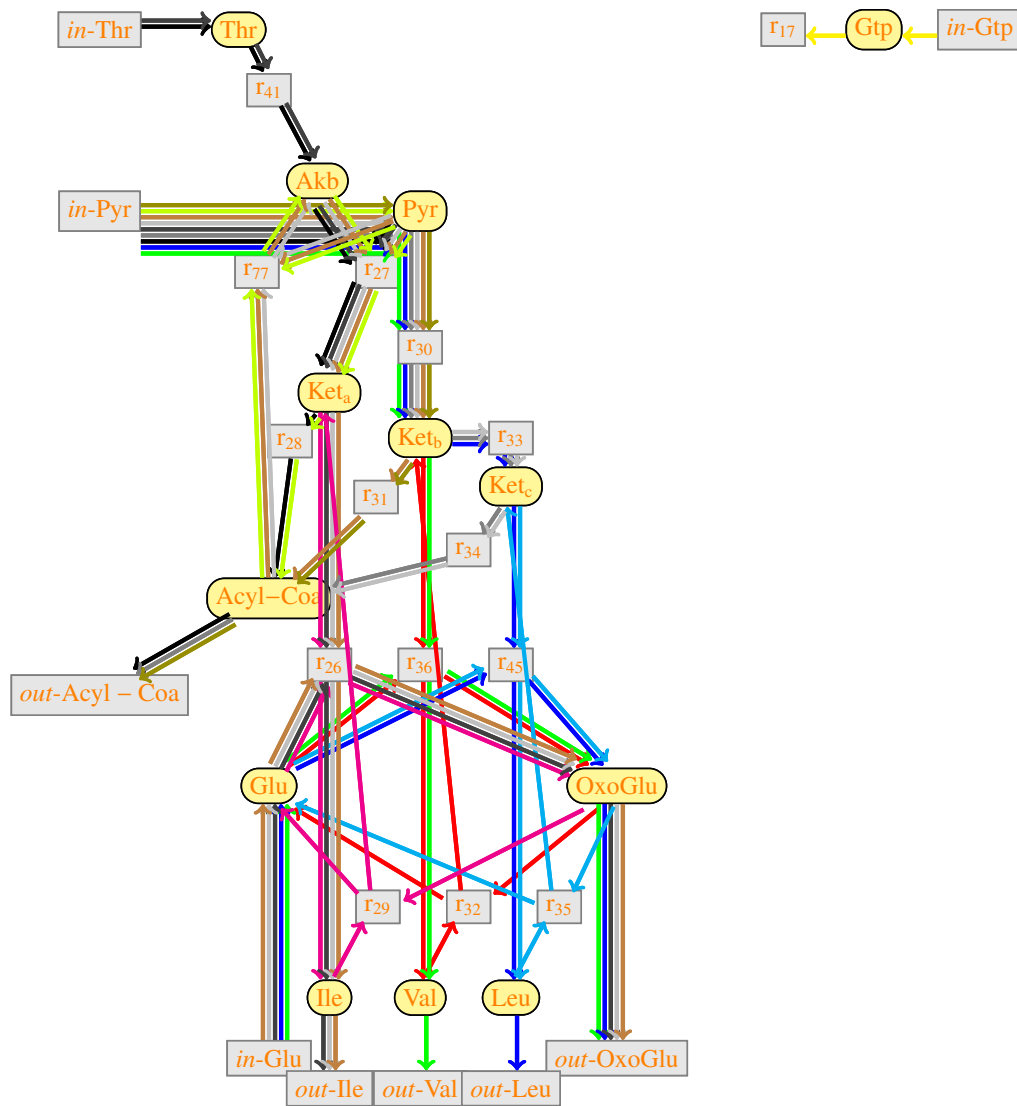


FIGURE 4.14 – Modèle du métabolisme de la bactérie *B. subtilis* Leu.

Prédiction de changements de réseau

*Dans cette section, nous présenterons le problème étudié dans cette thèse: le calcul d'une manière exacte de l'abstraction des différences de réseaux de réactions avec information cinétique partielle. Pour ceci, nous reverrons en premier lieu les travaux mis en place par l'équipe BioComputing pour la prédiction de changements de réseaux (Coutte et al., 2015 ; John et al., 2013a ; Niehren et al., 2015), qui motivent cet objectif. Ils posent les fondements des réseaux de réactions qui nous intéressent ainsi que leur algorithme de prédiction qualitative via l'abstraction approximative de John et la programmation par contraintes sur des domaines finis. L'approximation de John peut être améliorée par des heuristiques dont nous allons aborder le principe. Elle sera appliquée à d'autres heuristiques au cours de cette thèse. Nous démontrerons l'intérêt de ces algorithmes avec un exemple de prédiction sur un réseau biologique, la surproduction d'un métabolite la leucine qui est un acide aminé branché utilisé lors de la production de la surfactine (un peptide non ribosomique) par la bactérie *B. Subtilis*. Nous discuterons finalement l'outil développé par l'équipe BioComputing dont le but principal est de faire la prédiction de changements de réseau.*

Sommaire

5.1	Raisonnement intuitif	104
5.2	Abstraction de différences exacte	106
5.3	Approximation de John	108
5.4	Programmation par contraintes à domaine fini	108
5.5	Résoudre des contraintes à différences sur Δ_6	109

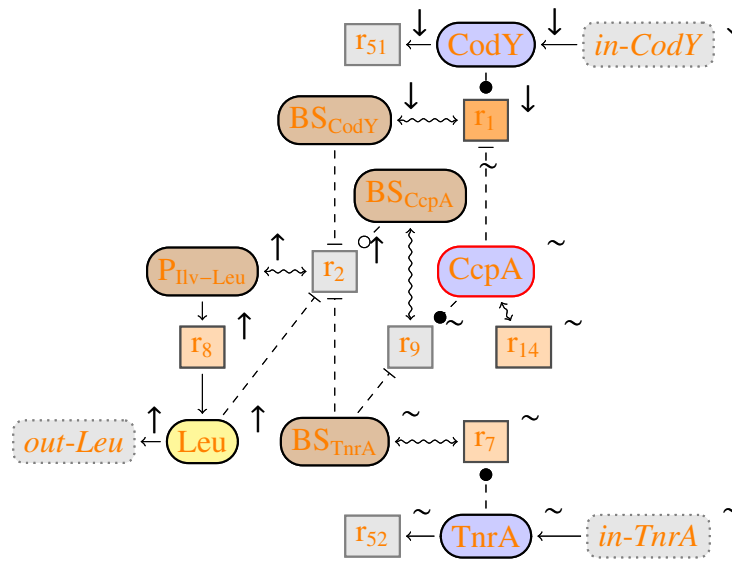


FIGURE 5.1 – Explication de la prédiction du changement d’influx $v_{in-CodY} = \downarrow$ pour obtenir la cible $v_{out-Leu} = \uparrow$.

5.6 Des solutions de contraintes à la prédiction	110
5.6.1 Algorithme	110
5.6.2 Exemple d’application	111
5.7 Heuristiques	112
5.7.1 Réécriture	112
5.7.2 Conséquences immédiates des modes élémentaires	113
5.8 Prédiction pour la surproduction de surfactine	113
5.9 L’outil BioComputing’s Reaction-networks	114

5.1 Raisonnement intuitif

Commençons par voir l’explication intuitive de l’approche pour prédire comment surproduire la leucine Leu dans le modèle $P_{Ilv-Leu}$ représenté graphiquement par la FIGURE 4.9. Les changements de modèles que nous admettons sont une augmentation ou une réduction des vitesses des flux d’entrées de $CodY$ et $TnrA$.

Sachant que la leucine est produite uniquement à partir de $P_{Ilv-Leu}$ par la réaction r_8 , pour atteindre notre cible, à savoir augmenter la concentration de leucine, il faut augmenter le taux de cette réaction. Cela est fait en augmentant l’activité de

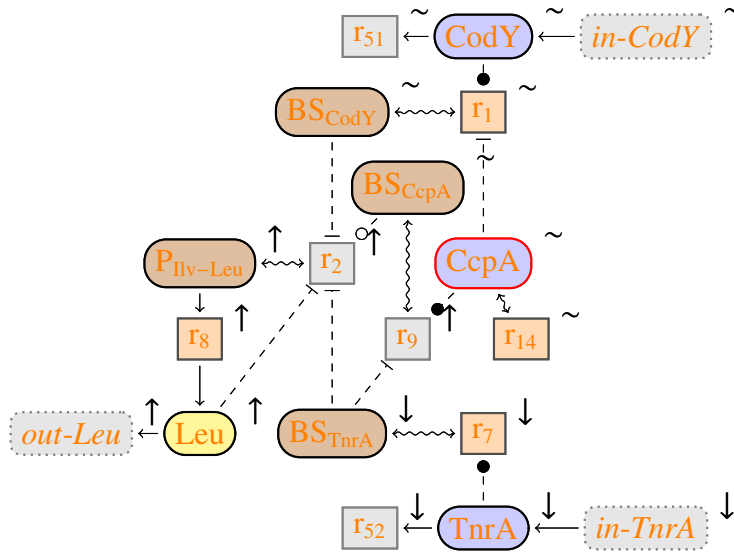


FIGURE 5.2 – Explication de la prédiction du changement d’influx $v_{in-TnrA} = \downarrow$ pour obtenir la cible $v_{out-Leu} = \uparrow$.

$P_{Ilv-Leu}$. Pour ce faire, il y a 3 possibilités qui consistent à diminuer l’effet d’inhibition d’un des trois inhibiteurs Leu , BS_{CodY} ou BS_{TnrA} , ou une autre en augmentant l’effet d’accélération de BS_{CcpA} . Sachant que l’on veut augmenter la production de Leu , il n’est pas dans notre intérêt de diminuer sa concentration, cette solution n’est donc pas à prendre en compte. Pour cette augmentation de l’activation du site de liaison BS_{CcpA} , on peut soit réduire l’activité de son inhibiteur BS_{TnrA} , ou augmenter la concentration de son activateur $CcpA$. Etant donné que $CcpA$, n’a pas de flux d’entrée, nous n’avons pas la main sur ce changement il peut donc être enlevé des solutions.

Avec ces informations on en arrive à vouloir réduire l’activité de BS_{CodY} ou celle de BS_{TnrA} . Si l’on ne considère que les changements de flux, les prédictions d’un changement seront :

- une augmentation du flux d’entrée de $CodY$
- une augmentation du flux d’entrée de $TnrA$

Pour illustrer cette explication intuitive, nous pouvons voir la représentation graphique de deux solutions avec la FIGURE 5.1, et la FIGURE 5.2, correspondant à une augmentation du flux d’entrée de $CodY$ ou respectivement de $TnrA$.

5.2 Abstraction de différences exacte

L'abstraction de différences h_{Δ_6} nous sera utile pour prédire des changements d'un réseau de réactions avec information cinétique partielle P . La cible du changement sera spécifiée par une formule $T \in \mathcal{F}_{\Sigma_{\text{pos-arith}}[\text{dom}(\Delta_6)]}$, qui sera à interpréter en Δ_6 . En particulier, la cible peut contenir des égalités comme :

$$v_{\text{out-Leu}} \doteq \uparrow$$

pour dire que la vitesse du flux de sortie de *Leu* doit être augmentée. La cible permet aussi de restreindre les valeurs abstraites que peuvent prendre les variables de flux d'entrée en incluant des contraintes de la forme :

$$\bigwedge_{A \in I} v_{\text{in-}A} \in \{\sim, \downarrow, \uparrow\}$$

On peut aussi exprimer la contrainte sur les knock-outs qui a été présentée auparavant, elle s'écrira donc :

$$\bigwedge_{r \in K} o_{\text{nom}_r} \in \{\sim, \Downarrow\}$$

Les abstractions de différences pour le réseau de réactions avec information cinétique partielle P avec ces contraintes cibles T doivent donc appartenir à l'ensemble d'assignements suivant, où $V = \text{fv}(P) \cup \text{fv}(T)$:

$$D_P(T) = h_{\Delta} \circ \text{sol}_{\mathbb{R}_+^2}^V(L_{\text{sim}(P)}) \cap \text{sol}_{\mathbb{R}_+^2}^{\Delta_6}(C_{\text{sim}(P)} \wedge \bigwedge_{r \in K} o_{\text{nom}_r} \in \{\sim, \Downarrow\} \wedge T)$$

Pour la prédiction de changement, nous ne nous intéresserons qu'à la valeur des *variables de contrôle* définies :

$$\text{controle}(P) = \{o_{\text{nom}_r} \mid r \in K\} \cup \{v_{\text{in-}A} \mid A \in I\}$$

Les autres variables de $D_P(T)$ peuvent être projetées, elles sont donc quantifiées existentiellement. Les valeurs des variables de contrôle de P sont contrôlées à l'extérieur du réseau P . Par exemple, par un knock-out de gène qui éteint la réaction représentant l'expression du gène, ou par des changements du milieu.

Ainsi, les projections de ces abstractions de différences sur les variables de contrôle, dont on veut fixer les valeurs avec les contraintes T , sont décrites ainsi :

$$D_P^{\text{controle}}(T) = \{\beta_{\text{controle}(P)} \mid \beta \in D_P(T)\}$$

L'intérêt de cette projection est qu'elle réduit considérablement le nombre d'assignements de différences abstraites, ce qui rend le calcul de la prédiction réalisable en temps de calcul.

Ainsi, pour tout sous-ensemble fini d'expressions $X, S \subseteq \mathcal{E}_{\Sigma_{pos-arith}[dom(\Delta_6)]}$ et $n \in \mathbb{N}$ nous définissons une formule de $\mathcal{F}_{\Sigma_{pos-arith}[dom(\Delta_6)]}$:

$$\#(X, S) \leq n =_{\text{def}} \bigwedge_{\substack{X' \subseteq X \\ |X'|=n+1}} \neg \bigwedge_{x \in X'} x \in S$$

disant qu'il n'y a pas plus de n expressions incluses dans X qui sont égales à des expressions de S . Si l'on souhaite, par exemple, que le nombre maximum de changements de variable de contrôle soit égal à 2, alors pour les variables de contrôle $o_1, \dots, o_l, v_{in-A_1}, \dots, v_{in-A_k}$, la contrainte s'écrira :

$$\#(\{o_1, \dots, o_l, v_{in-A_1}, \dots, v_{in-A_k}\}, \{\Downarrow, \downarrow, \uparrow\}) \leq 2$$

où $\{A_1, \dots, A_n\} = I$ sont les espèces ayant un flux d'entrée et $K = \{1, \dots, l\}$ les noms des candidats au knock-out.

Les variables candidates pour un knock-out sont colorées en orange clair sur la représentation graphique du modèle. Si lorsque l'on affiche une solution une réaction est sélectionnée pour un knock-out, elle sera colorée en orange foncé.

Exemple 59. Si l'on reprend l'exemple du réseau $P_{Ilv-Leu}$, les contraintes additionnelles nécessaires pour la prédiction de 2 changements à appliquer pour surproduire l'espèce *Leu* sont :

$$\begin{aligned} T &= v_{in-CodY} \in \{\sim, \downarrow, \uparrow\} && \% \text{ inflows } \{CodY, TnrA\} \in I \\ &\wedge v_{in-TnrA} \in \{\sim, \downarrow, \uparrow\} \\ &\wedge o_1 \in \{\sim, \Downarrow\} && \% \text{ knockout candidates } \{1, 7, 14, 8\} \in K_{P_{Ilv-Leu}} \\ &\wedge o_7 \in \{\sim, \Downarrow\} \\ &\wedge o_{14} \in \{\sim, \Downarrow\} \\ &\wedge o_8 \in \{\sim, \Downarrow\} \\ &\wedge \#(\{o_1, o_7, o_{14}, o_8, v_{in-CodY}, v_{in-TnrA}\}, \{\Downarrow, \downarrow, \uparrow\}) \leq 2 && \% \text{ at most 2 changes} \\ &\wedge v_{out-Leu} \doteq \uparrow && \% \text{ overproduction target} \end{aligned}$$

Pour faire une prédiction exacte, nous aimerions résoudre le problème algorithmique ABSTRACTION EXACTE définie en FIGURE 5.3. Ce problème demande de calculer $D_p^{\text{contrôle}}(T)$ pour un réseau P et une cible T donnés.

La résolution de ce problème exact n'est pas trivial, étant donné que l'ensemble des solutions réelles d'un système linéaire est infinie, et donc il en va de même pour l'ensemble de solution $sol_V^{\mathbb{R}_+^2}(L_{sim}(P))$.

En effet, il n'est pas possible de générer toutes ces solutions en premier lieu, puis de les abstraire ensuite. Pour éviter ce problème algorithmique, il a été proposé de surapproximer l'abstraction exacte.

entrées:

- un réseau de réactions avec cinétique partielle P , et
- une cible $T \in \mathcal{F}_{\Sigma_{pos-arith}[dom(\Delta_6)]}$.

sortie:

$$D_P^{contrôle}(T)$$

FIGURE 5.3 – Le problème algorithmique ABSTRACTION EXACTE.

entrées:

- un réseau de réactions avec cinétique partielle P , et
- une cible $T \in \mathcal{F}_{\Sigma_{pos-arith}[dom(\Delta_6)]}$.

sortie:

$$sol^{\Delta_6}(\exists \mathcal{V} \setminus \text{contrôle}(P). L_{sim(P)} \wedge C_{sim(P)} \wedge \bigwedge_{r \in K} o_{nom_r} \in \{\sim, \Downarrow\} \wedge T)$$

FIGURE 5.4 – Le problème algorithmique APPROXIMATION DE JOHN.

5.3 Approximation de John

Nous rappelons par la section 3.2.3, que Δ_6 est bien une $\Sigma_{pos-arith}[dom(\Delta_6)]$ -structure. Une première surapproximation de $h_{\Delta_6} \circ sol_V^{\mathbb{R}_+^2}(L_{sim(P)})$ est donc donnée par le Théorème 34 de John. Ainsi on a :

$$h_{\Delta_6} \circ sol_V^{\mathbb{R}_+^2}(L_{sim(P)}) \subseteq sol_V^{\Delta_6}(L_{sim(P)})$$

En se basant sur cette surapproximation, nous considérons le problème algorithmique APPROXIMATION DE JOHN en FIGURE 5.4. Étant donné un réseau P et une cible $T \in \mathcal{F}_{\Sigma_{pos-arith}[dom(\Delta_6)]}$, il s'agit de calculer dans la structure finie Δ_6 l'ensemble de solutions de la formule :

$$\exists \mathcal{V} \setminus \text{contrôle}(P). L_{sim(P)} \wedge C_{sim(P)} \wedge \bigwedge_{r \in K} o_{nom_r} \in \{\sim, \Downarrow\} \wedge T$$

L'ensemble de ces solutions est une surapproximation de $D_P^{contrôle}(T)$ qui sera calculé par l'ABSTRACTION EXACTE.

5.4 Programmation par contraintes à domaine fini

La programmation par contraintes à domaine fini (Schulte and Carlsson, 2006) permet d'énumérer l'ensemble des solutions d'une Σ -formule de la logique du premier-ordre sur une Σ -structure à domaine fini. Pour voir cela, il faut noter que les relations de la Σ -structure peuvent être spécifiées par des tables. De plus, comme le domaine de la Σ -structure est fini, des quantificateurs existentiels peuvent

être réécrits en disjonctions et les quantificateurs universels en conjonctions. Nous utiliserons le langage MiniZinc (Rendl et al., 2015) pour écrire les contraintes, et le solveur Gecode (Schulte et al., 2010) pour leur résolution. La syntaxe de MiniZinc est proche du raisonnement mathématiques et rend son utilisation plus facile.

Par conséquent, toute $\Sigma_{pos-arith}[dom(\Delta_6)]$ -formule du premier-ordre peut être résolue dans la $\Sigma_{pos-arith}[dom(\Delta_6)]$ -structure finie Δ_6 avec l'utilisation d'un solveur de contrainte à domaine fini.

Les problèmes de satisfaction de contraintes, ou de formules de logique de premier-ordre, sont des problèmes durs d'un point de vue de complexité. Le problème de satisfiabilité de contraintes conjonctives booléennes est NP-complet, et le problème de satisfiabilité de formule booléenne de logique de premier ordre est PSPACE-complet. La complexité devrait être similaire pour des contraintes ou formules sur la $\Sigma_{pos-arith}[dom(\Delta_6)]$ -structure Δ_6 .

Même si elle est dure dans le pire des cas, la programmation par contraintes nous donne de bons résultats pour les cas qui nous intéressent, mais ce n'est pas le cas en général. C'est pourquoi il est important de réduire le nombre de valeurs possibles pour les variables, comme par exemple, par l'ajout de la contrainte du type $\#(X, S) \leq n$, qui pose une limite maximale sur le nombre n de variables de X autorisées à prendre des valeurs dans S . Il est également important de réduire le nombre de variables libres, comme le fait la restriction de variables de contrôle avec la quantification existentielle dans l'approximation de John en FIGURE 5.4.

5.5 Résoudre des contraintes à différences sur Δ_6

Pour résoudre des contraintes à différences, il faut donc d'abord définir dans le langage MiniZinc notre structure Δ_6 avec le domaine abstrait $dom(\Delta_6) = \{\sim, \uparrow, \downarrow, \approx, \uparrow\uparrow, \downarrow\downarrow\}$. L'interprétation de chaque symbole de $\Sigma_{pos-arith}$ doit être spécifiée, on voit que le prédicat *abstractSumTable* permet d'identifier chaque élément de la table *sumTable* comme un élément satisfaisant la relation de somme abstraite. Le prédicat *abstractSum* quant à lui permet de mettre en relation les 3 variables qui lui sont passées en paramètre. Nous ne les verrons pas ici mais nous donnons aussi la définition en terme de contraintes de l'accélération (*abstractAccelerator*), l'inhibition (*abstractInhibitor*), le produit (*abstractProduct*) et les constantes.

A titre d'exemple, l'équation $v_7 \doteq o_7 + \Delta_6 \text{TnrA}$ devient en MiniZinc la contrainte relationnelle :

```
constraint abstractSum(o7, TnrA, v7);
```

```

predicate abstractSum(var int: s1, var int: s2, var int: s3) =
  abstractSumTable(s1, s2, s3);

predicate abstractSumTable(var Delta6: s1, var Delta6: s2, var Delta6: s3) =
  let {
    array [1..3] of var int: tuple;
    constraint tuple[1] = s1;
    constraint tuple[2] = s2;
    constraint tuple[3] = s3;
  } in table(tuple, sumTable);

array [1..52, 1..3] of Delta6: sumTable =
[[| No, No, No | No, Up, Up | No, Do, Do | No, NoN, No | No, UpN, Up | No, DoN, Do
| Up, No, Up | Up, Up, Up | Up, Do, Up | Up, Do, No | Up, Do, Do | Up, NoN, Up
| Up, UpN, Up | Up, DoN, Up | Up, DoN, No | Up, DoN, Do | Do, No, Do | Do, Up, Up
| Do, Up, No | Do, Up, Do | Do, Do, Do | Do, NoN, Do | Do, UpN, Up | Do, UpN, No
| Do, UpN, Do | Do, DoN, Do | NoN, No, No | NoN, Up, Up | NoN, Do, Do | NoN, NoN,
  NoN
| NoN, UpN, UpN | NoN, DoN, DoN | UpN, No, Up | UpN, Up, Up | UpN, Do, Up | UpN, Do, No
| UpN, Do, Do | UpN, NoN, UpN | UpN, UpN, UpN | UpN, DoN, Up | UpN, DoN, No | UpN, DoN, Do
| DoN, No, Do | DoN, Up, Up | DoN, Up, No | DoN, Up, Do | DoN, Do, Do | DoN, NoN,
  DoN
| DoN, UpN, Up | DoN, UpN, No | DoN, UpN, Do | DoN, DoN, DoN
|];

```

FIGURE 5.5 – Définition de l’opérateur abstrait de somme en Δ_6 dans le langage MiniZinc.

5.6 Des solutions de contraintes à la prédiction

Nous allons voir comment sont utilisés les ensembles de solutions des contraintes à différence pour faire un raisonnement qualitatif sur les réseaux de réactions avec information cinétique partielle.

5.6.1 Algorithme

Actuellement pour faire nos prédictions, nous ne nous intéressons qu’aux solutions qui ont un seul changement (non nul). Pour faire de la prédiction de single knock-out de gènes, nous nous intéressons uniquement aux solutions qui appliquent comme unique changement un knock-out.

L’algorithme de prédiction a la forme :

entrée :

- Un réseau de réactions P
- Un ensemble de contraintes cibles T

sortie:

- Un ensemble de prédictions

Avec l’approximation de John, nous résolvons le système à différences $sol^{\Delta_6}(\exists \mathcal{V} \setminus controle(P). L_{sim(P)} \wedge C_{sim(P)} \wedge \bigwedge_{r \in K} o_r \in \{\sim, \Downarrow\} \wedge T)$ avec un solveur de contraintes à domaine fini.

A titre d’exemple, pour savoir quels changements sur les flux d’entrée du réseau $P_{Ilv-Leu}$ permettent d’engendrer une augmentation du flux de sortie de **Leu**,

nous devons interpréter les contraintes à différences inférées à partir du réseau sur Δ_6 , que nous retrouvons avec la FIGURE 4.12 et la FIGURE 4.11, et les contraintes cibles décrites dans l'exemple 59. Nous ajoutons la contrainte:

$$\#(\{o_1, o_7, o_{14}, o_8\}, \{\downarrow\}) \leq 0$$

à T pour indiquer qu'il n'y a pas de knock-out de réaction autorisée. Puisqu'on ne s'intéresse qu'aux changements de flux d'entrées, les variables de contrôle conservées après projection sur les solutions abstraites seront donc ici les variables $v_{in-CodY}$ et $v_{in-TnrA}$.

$v_{in-CodY}$	$v_{in-TnrA}$
↓	~
~	↓

FIGURE 5.6 – Ensemble de solutions abstraites calculé à partir de l'ensemble de contraintes avec cible inféré du réseau $P_{Ilv-Leu}$ sur Δ_6 .

La FIGURE 5.6 nous donne l'ensemble de solutions générées.

Une ligne représente une solution, et pour chaque colonne on a une valeur qui est associée à la variable concernée, ici $v_{in-CodY}$, et $v_{in-TnrA}$. Chaque ligne sera ensuite interprétée en prédiction. Ici il y a 2 solutions donnant 2 prédictions, à savoir une diminution de **CodY**, ou une diminution de **TnrA**.

Une fois l'analyse faite, on peut représenter une solution générée directement sur le graphe en annexant chaque nœud (espèces et réactions) par la valeur de Δ_6 qui lui est assignée.

Nous pouvons voir la représentation graphique d'une solution avec la FIGURE 5.1 pour laquelle les deux entrées diminuent.

En prenant les solutions minimales, notre algorithme nous dit que pour avoir une augmentation de *Leu* les deux solutions existantes sont une diminution de $v_{in-TnrA}$ ou une diminution de $v_{in-CodY}$.

5.6.2 Exemple d'application

Si l'on applique une fois de plus l'heuristique approximative de John, pour 2 changements maximum, la FIGURE 5.7 donne la table de solutions pour les 11 meilleures prédictions. On y retrouve bien les changements expliqués précédemment, seul ou combiné, à savoir le knock-out de la réaction r_1 et r_7 et les diminutions des flux $v_{in-CodY}$ et $v_{in-TnrA}$.

Si l'on prend les changements minimaux, la prédiction se restreint 4 premières solutions, ou aux 2 premières si l'on ne s'intéresse qu'au single knock-out.

O_1	O_7	O_8	O_{14}	$V_{in-CodY}$	$V_{in-TrnA}$
↓	~	~	~	~	~
~	↓	~	~	~	~
~	~	~	~	~	↓
~	~	~	~	↓	~
~	~	~	~	↓	↓
↓	~	~	~	↓	~
↓	~	~	~	~	↓
↓	~	~	~	↑	~
~	↓	~	~	↓	~
~	↓	~	~	~	↓
~	↓	~	~	~	↑

FIGURE 5.7 – Table des 11 meilleures solutions de l’heuristique approximative de John.

5.7 Heuristiques

Afin d’améliorer la précision de l’approximation, des heuristiques ont été mise en place. Elles partagent la même idée qui est l’ajout de contraintes linéaires au système de départ avant de faire son interprétation abstraite.

5.7.1 Réécriture

Si l’on considère un réseau de réactions avec cinétique partielle P , pour obtenir ces contraintes G , il suffit d’appliquer un algorithme de Gauss sur le système linéaire $L_{sim}(P)$, on a $G = Gauss(L_{sim}(P))$. On obtient donc l’algorithme suivant :

entrée

— un réseau de réactions avec cinétique partielle P , et

— une cible $T \in \mathcal{F}_{\Sigma_{pos-arith}[dom(\Delta_6)]}$.

sortie:

$$sol^{\Delta_6}(\exists \mathcal{V} \setminus controle(P). L_{sim}(P) \wedge C_{sim}(P) \wedge \bigwedge_{r \in K} o_r \in \{\sim, \downarrow\} \wedge T \wedge G)$$

Cependant, cette heuristique reste faible comme l’algorithme de Gauss n’est appliqué qu’une fois sur un seul ordre du système linéaire. Des contraintes additionnelles seraient obtenues si l’on appliquait le Gauss sur le système avec un ordre différent sur les équations. Sur les réseaux simples vus jusqu’à présent cette heuristique n’améliore pas les résultats. Si l’on regarde à présent le réseau de réactions représentant une partie plus large du métabolisme de la bactérie B. Subtilis, on voit que le nombre de solutions pour une prédiction de 2 changements maximum dans le but d’augmenter la leucine passe de 292 solutions dont 16 doubles

knock-outs avec la méthode approximative de John à 249 dont 14 doubles knockouts. Cette approche permet donc de réduire un peu le nombre de fausses solutions sans rajouter de différence en terme de temps de calcul.

5.7.2 Conséquences immédiates des modes élémentaires

Basée sur le même principe d'ajout de contraintes, une autre heuristique a été mise en place au cours de cette thèse. Ces contraintes additionnelles reposent sur l'étude des flux dans la partie métabolique du réseau de réactions, et met en lien les variables de flux d'entrée et de flux de sortie selon les flux qui les traversent.

Si on définit une variable e_i , où $0 \leq i \leq n$ avec n le nombre de modes élémentaire, pour chaque mode élémentaire du réseau, nous pouvons décrire chaque réaction du réseau comme la somme des modes qui la composent, et créer de nouvelles équations mettant en lien les flux d'entrée et de sortie par leur décomposition. En analysant la décomposition en flux métabolique, on voit que l'on a l'égalité $v_{in-Glu} = v_{out-OxoGlu}$ car ces variables ont toutes les deux les 5 même flux qui les décrivent. On note aussi que $v_{in-Thr} = \mathbf{r}_{41} = \mathbf{r}_{27}$. Ces 3 égalités forment l'ensemble de contraintes additionnelles, noté F , on obtient ainsi l'algorithme suivant:

entrée

- un réseau de réactions avec cinétique partielle P , et
- une cible $T \in \mathcal{F}_{\Sigma_{pos-arith}[dom(\Delta_6)]}$.

sortie:

$$sol^{\Delta_6}(\exists \mathcal{V} \setminus \text{controle}(P). L_{sim(P)} \wedge C_{sim(P)} \wedge \bigwedge_{r \in K} o_r \in \{\sim, \Downarrow\} \wedge T \wedge F)$$

Cependant, le nombre de conséquences engendrées même si elles sont essentielles ne sont pas suffisantes pour donner de réelles améliorations. En effet, si l'on regarde les solutions qui sont générées pour la prédiction qui a été faite avec l'heuristique précédente, on obtient 250 solutions dont 16 knockouts. Cela améliore l'approximation de John, mais n'est toujours pas satisfaisant. Nous verrons par la suite qu'une contribution majeure de cette thèse repose sur l'étude des modes élémentaires utilisés d'une autre façon.

5.8 Prédiction pour la surproduction de surfactine

Cette analyse qualitative est mise en application sur un vrai modèle biologique, l'objectif étant de prédire quels changements effectuer sur la bactérie B. Subtilis pour qu'elle surproduise la surfactine, un peptide non ribosomique, en grande quantité. Dans un article de [Coutte et al. \(2015\)](#), il a été prouvé qu'une augmentation de la production de leucine, qui est un précurseur de la surfactine, amenée à une augmentation de la production en surfactine. L'augmentation de la production de surfactine est donc notre cible. Le réseau de réactions qui a été

modélisé pour l'étude de l'approximation de John représente une partie du métabolisme de la bactérie avec sa régulation. Il s'agit de la chaîne des acides aminés : isoleucine, leucine et valine. Nous retrouvons dans sa représentation graphique sur la partie en haut à droite le sous-réseau $P_{Ilv-Leu}$, que nous avons expliqué en détail juste au dessus.

Nous ajoutons la possibilité de définir une contrainte abstraite lorsque qu'une espèce S est essentielle pour imposer que sa valeur ne tombe jamais à zéro. Autrement dit une formule du type $S \notin \{\downarrow, \uparrow, \approx\}$.

D'après la table de solutions retournées lors de l'approximation de John, pour la prédiction de single knock-out du réseau de réactions représentant le cycle carbone des acides aminés branchés, nous avons la prédiction qu'une des variables parmi l'ensemble qui suit doit être stoppée par un knock-out :

$$\{o_1, o_3, o_4, o_7, o_{10}, o_{11}, o_{12}, o_{13}, o_{14}, o_{15}, o_{16}, o_{37}, o_{38}, o_{40}, o_{46}, o_{47}\}$$

On retrouve un résumé du comparatif avec les heuristiques dans la table 5.10, il représente le nombre de solutions obtenues pour l'analyse de ce réseau lors de la prédiction de 1 changement avec l'approximation de John et les 2 autres heuristiques présentées précédemment. On voit que ces heuristiques permettent d'améliorer un peu les prédictions.

5.9 L'outil BioComputing's Reaction-networks

Les travaux effectués par l'équipe BioComputing pour la prédiction de changements de réseaux (Niehren et al., 2015), (John et al., 2013a), (Coutte et al., 2015) ont abouti à la mise en place d'un outil : BioComputing's Reaction-networks. Bien que cet outil n'ait pas encore d'interface adapté pour une utilisation externe, il est complet.

Cet outil permet de modéliser des réseaux de réactions avec le langage que nous avons présenté en section 4.4 et d'effectuer des prédictions de changements via la méthode approximative de John.

Il prend en entrée un fichier XML décrivant un réseau de réactions, et un autre contenant les contraintes cibles. A l'aide de la sémantique stable, l'ensemble des contraintes à différences est traduit en MiniZinc puis résolu par le solveur GeCode.

De plus par une transformation XSLT, l'outil produit la représentation graphique des réseaux sous le format LaTeX par l'utilisation de la librairie Tikz. Les syntaxes graphiques et XML, étant équivalentes il est également possible d'inférer la syntaxe XML à partir d'une représentation graphique. L'outil offre donc une sortie LATEX complète contenant le détail concernant le réseau de réactions : sa représentation graphique, l'ensemble de contraintes à différences issues de la

sémantique stable et les contraintes cibles additionnelles; et les résultats de la prédiction sous forme de table comme nous l'avons vu et quelques prédictions sont représentées graphiquement également.

Lors de cette thèse nous verrons que nous avons étendu cet outil en y ajoutant et intégrant un module permettant la représentation graphique de flux, ainsi qu'un module pour chaque nouvel algorithme développé, à savoir, l'algorithme exact et l'heuristique EFM-conséquences.

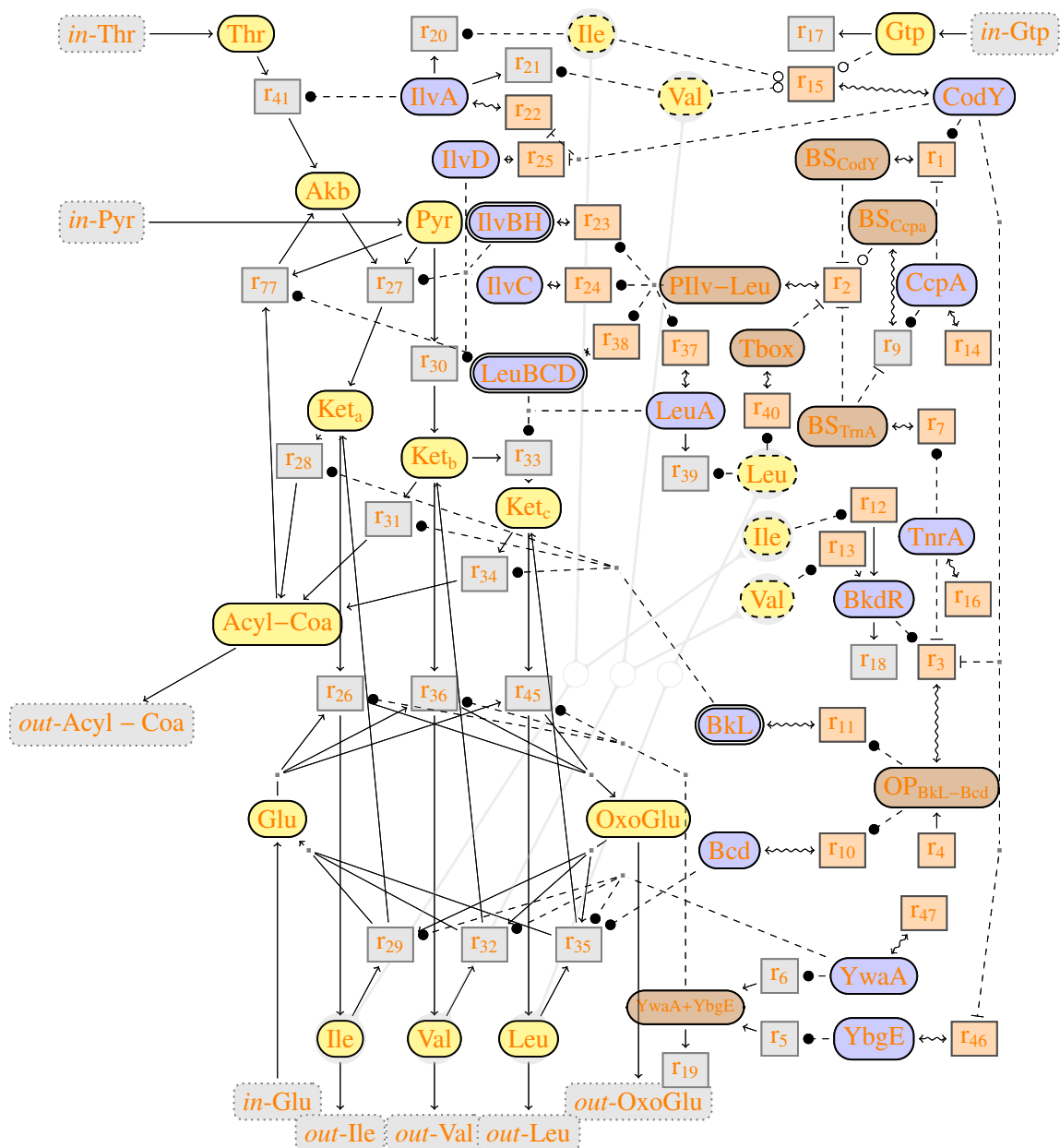


FIGURE 5.8 – Modèle du métabolisme avec régulation de la bactérie *B. Subtilis*
Leu

O_1	O_3	O_4	O_7	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}	O_{16}	O_{37}	O_{38}	O_{40}	O_{46}	O_{47}
⇓	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
~	⇓	~	~	~	~	~	~	~	~	~	~	~	~	~	~
~	~	⇓	~	~	~	~	~	~	~	~	~	~	~	~	~
~	~	~	⇓	~	~	~	~	~	~	~	~	~	~	~	~
~	~	~	~	⇓	~	~	~	~	~	~	~	~	~	~	~
~	~	~	~	~	⇓	~	~	~	~	~	~	~	~	~	~
~	~	~	~	~	~	⇓	~	~	~	~	~	~	~	~	~
~	~	~	~	~	~	~	⇓	~	~	~	~	~	~	~	~
~	~	~	~	~	~	~	~	⇓	~	~	~	~	~	~	~
~	~	~	~	~	~	~	~	~	⇓	~	~	~	~	~	~
~	~	~	~	~	~	~	~	~	~	⇓	~	~	~	~	~
~	~	~	~	~	~	~	~	~	~	~	⇓	~	~	~	~
~	~	~	~	~	~	~	~	~	~	~	~	⇓	~	~	~
~	~	~	~	~	~	~	~	~	~	~	~	~	⇓	~	~
~	~	~	~	~	~	~	~	~	~	~	~	~	~	⇓	~
~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	⇓

FIGURE 5.9 – Table de solutions pour la prédiction de single knock-out pour le réseau de réaction *Leu* avec l'approximation de John.

	John	Flux	Réécriture
<i>Knock – outs</i>	16	14	14
<i>Solutions</i>	25	23	23

FIGURE 5.10 – Comparatif des algorithmes pour la prédiction de double knock-out du réseau de réactions représentant le cycle carbone des acides aminés.

Chapitre 6

Abstraction booléenne de système mixte réel

Dans ce chapitre, nous présentons la première contribution en montrant qu'il est possible de calculer exactement l'abstraction booléenne de système mixte. Nous verrons qu'il fournit un environnement permettant le calcul d'abstractions finies de systèmes linéaires d'équations, en se penchant plus particulièrement sur le calcul de l'abstraction de signes de systèmes linéaires d'équations.

Sommaire

6.1	Introduction	119
6.2	Exemple d'analyse de programme	121
6.3	Exactitude de l'abstraction	122
6.4	Invariance de l'abstraction	127
6.5	Abstraction booléenne de systèmes mixtes	132
6.6	Calcul de l'abstraction de signe	134

6.1 Introduction

Les systèmes d'équations linéaires sont utilisés en interprétation abstraite pour abstraire la sémantique de programmes avec des opérations arithmétiques (Cousot and Halbwachs, 1978 ; Granger, 1991). En biologie des systèmes, les systèmes d'équations linéaires sont utilisés pour décrire les flux d'un réseau réactions à

l'état stationnaire (Lotz et al., 2014 ; Maranas and Zomorodi, 2016). Les deux applications soulèvent une question très similaire sur le calcul de l'abstraction d'un système linéaire d'équations.

Pour l'analyse de programme basée sur l'interprétation abstraite (Miné, 2002), on peut se demander par exemple si la valeur d'une variable x dans un programme est strictement positive lorsque la valeur d'une autre variable y l'est. Cette question concerne l'abstraction de signes d'un ensemble de solutions réelles d'un système d'équations linéaires (qui lui-même est abstrait de la sémantique du programme). L'abstraction de signes $h_{\mathbb{S}} : \mathbb{R} \rightarrow \{-1, 0, 1\}$ satisfait :

$$h_{\mathbb{S}}(r) = \begin{cases} 1 & \text{si } r > 0, \\ h_{\mathbb{S}}(r) = -1 & \text{si } r < 0, \\ h_{\mathbb{S}}(0) = 0 & \text{sinon.} \end{cases}$$

En biologie des systèmes (Allart et al., 2019 ; Coutte et al., 2015 ; John et al., 2013a ; Niehren et al., 2016) on peut vouloir savoir si pour un réseau de flux, un flux x doit augmenter (resp. diminuer) si un autre flux y le fait. Cette question concerne l'abstraction des différences d'une paire de solutions réelles positives d'un système d'équations. Nous montrerons dans le chapitre suivant que le calcul de cette abstraction peut se ramener à un problème d'abstraction booléenne d'un système étendu.

Dans les deux cas, le calcul de ces abstractions de systèmes d'équations linéaires peut se ramener à un problème d'abstraction booléenne de systèmes mixtes, nous reviendrons sur cela plus tard. En admettant cela, si nous souhaitons calculer exactement ces abstractions pour des systèmes d'équations linéaires, il faut montrer que l'abstraction booléenne d'un système mixte est $h_{\mathbb{B}}$ -exact. C'est le fondement de la problématique de ce chapitre.

Dans un premier temps, nous montrons que n'importe quelle équation linéaire de matrice d'entiers $Ax=0$ peut être transformée en une Σ -formule \mathbb{R}_+ -équivalente qui est quasi-positive et quasi-triangulaire et ainsi $h_{\mathbb{B}}$ -exacte. Cette transformation est basée sur le calcul des modes élémentaires (Gagneur and Klamt, 2004 ; Komei and Alain, 1996 ; T.S. et al., 1953 ; Zanghellini et al., 2013) qui peut être fait en pratique par différentes bibliothèques venant de la géométrie algorithmique (Bagnara et al., 2008). La conversion peut prendre un temps exponentiel dans le pire des cas, mais se comporte bien en général.

La correction de la conversion est due au fait que les Σ -algèbre \mathbb{R}_+ et \mathbb{B} permettent la division unique par un nombre naturel non nul.

Dans un second temps, nous introduisons les systèmes $h_{\mathbb{B}}$ -mixte, qui généralisent les systèmes linéaires d'équations, avec les équations polynomiales positives $p=0$ et les inéquations $p \neq 0$ où le polynôme p n'a pas de terme constant. Nous montrons ensuite que les systèmes $h_{\mathbb{B}}$ -mixtes peuvent être convertis en une

formule $h_{\mathbb{B}}$ -exacte elle aussi. Pour faire cela, nous étendons les résultats de la première étape en introduisant la notion de Σ -formules $h_{\mathbb{B}}$ -invariantes, qui englobent les équations polynomiales $p=0$ et les inéquations $p \neq 0$ pour tout polynôme positif p sans terme constant.

Nous verrons ensuite la difficulté du calcul de l'abstraction de signe d'un système d'équations linéaires ϕ avec les méthodes existantes, cela se formalise $h_{\mathbb{S}} \circ \text{sol}^{\mathbb{R}}(\phi)$. A noter que l'ensemble $\text{sol}^{\mathbb{R}}(\phi)$ ne peut pas être énuméré puisqu'il est infini. A la place nous pouvons énumérer l'ensemble d'assignement de signe $\alpha : V \rightarrow \{-1, 0, 1\}$ qui est fini. De plus, puisque $h_{\mathbb{S}}(x) = -1$ est équivalent à $x < 0$ et $h_{\mathbb{S}}(x) = 1$ à $0 < x$, le système $\phi \wedge \bigwedge_{x \in V} h_{\mathbb{S}}(x) = \alpha(x)$ est équivalent à un système d'équations linéaires et d'inéquations strictes. La satisfiabilité d'un tel système peut être décidée en temps exponentiel (Dines, 1926). Cependant, cette méthode n'est pas faisable en pratique étant donné que le nombre d'assignements de signe croît exponentiellement avec le nombre de variables. Donc la question est de savoir s'il existe un algorithme plus efficace pour calculer $h_{\mathbb{S}} \circ \text{sol}^{\mathbb{R}}(\phi)$.

Ainsi dans un troisième temps, nous réécrivons les systèmes ϕ en formules ϕ' $h_{\mathbb{B}}$ -mixtes, en décomposant chaque variable en deux variables représentant son signe, et en se basant sur les deux premières étapes. De cette façon, l'abstraction de signes $h_{\mathbb{S}} \circ \text{sol}^{\mathbb{R}}(\phi)$ peut être calculée à partir de l'abstraction booléenne $h_{\mathbb{B}} \circ \text{sol}^{\mathbb{R}}(\phi') = \text{sol}^{\mathbb{B}}(\phi')$. Il suffit ensuite de calculer l'ensemble de solutions booléennes $\text{sol}^{\mathbb{B}}(\phi')$ avec la programmation par contraintes à domaine fini.

L'approche de réécriture basée sur les résultats autour des systèmes $h_{\mathbb{B}}$ -mixtes a été utilisée (Allart et al., 2019) pour faire le calcul de l'abstraction des différences $h_{\Delta_3} : \mathbb{R}_+^2 \rightarrow \{\uparrow, \downarrow, \sim\}$, et, pour un raffinement avec une Σ -structure finie à 6 éléments, comme nous le verrons dans le chapitre suivant. Cela montre que les résultats présentés dans ce chapitre fournissent un environnement permettant le calcul de différentes abstractions finies de systèmes linéaires d'équations.

6.2 Exemple d'analyse de programme

Nous illustrons nos résultats en appliquant l'abstraction de signes pour l'analyse de programme basée sur l'interprétation abstraite. Nous considérons l'implémentation Python FIGURE 6.1 de la fonction `integral` : $\mathbb{R}^2 \rightarrow \mathbb{R}$ avec $f : \mathbb{R} \rightarrow \mathbb{R}$. Un appel `integral(a, step)` calcule l'approximation de l'intégrale $\int_0^a f(x)dx$ avec un pas `step`. L'interprétation abstraite appliquée à ce programme avec un domaine abstrait peut produire la formule du premier ordre ϕ_{integral} suivante :

$$(\text{throw_exception} = 1 \iff a < 0) \wedge (\text{do_recursion} = 1 \iff \text{step} \leq a) \wedge a_{\text{rec}} = a - \text{step} \wedge \text{step}_{\text{rec}} = \text{step}$$

```

def integral(a: float, step: float):
    if a < 0: raise ValueError('This should never happen')
    if step > a:
        return 0
    else:
        return step * f(a) + integral(a - step, step)

```

FIGURE 6.1 – Fonction Python approximant l'intégrale $\int_0^a f(x)dx$ pour une fonction donnée $f : \mathbb{R} \rightarrow \mathbb{R}$.

Cette formule utilise les variables suivantes: *throw_exception* qui est vraie au déclenchement d'une exception; *do_recursion* qui est vraie quand un appel récursif est fait; les deux variables a_{rec} , $step_{rec}$ représentent les paramètres passés récursivement à *integral*.

Pour savoir si une exception serait levée, nous sommes intéressés par l'abstraction de signe pour cette formule $h_{\mathbb{S}} \circ sol^{\mathbb{R}}(\phi_{\text{integral}})$. Selon le Théorème John (John et al., 2013a), cette abstraction peut être sur-approximée par $sol^{\mathbb{S}}(\phi_{\text{integral}})$ qui elle peut être calculée par la programmation par contraintes à domaine fini. Cependant, cette approximation ne relève pas que a_{rec} peut être strictement négative quand *do_recursion* est vraie, aussi, cette condition n'est pas atteinte par le programme. A l'inverse, cela est correctement reflété par l'abstraction de son interprétation abstraite $h_{\mathbb{S}} \circ sol^{\mathbb{R}}(\phi_{\text{integral}})$, qui peut être calculée en le convertissant en un système $h_{\mathbb{B}}$ -mixte qui est ensuite résolu avec les méthodes citées ci-dessus.

6.3 Exactitude de l'abstraction

Le Théorème de John 34 sur la surapproximation montre que l'ensemble de solutions sur le domaine abstrait $sol^{\Delta}(\phi)$ est une approximation par abstraction de l'ensemble concret de solutions $h(sol^{\mathbb{S}}(\phi))$ pour n'importe quelle abstraction $h : \mathbb{S} \rightarrow \Delta$ d'une structure concrète vers une abstraite et n'importe quelle formule du premier ordre ϕ . On dit que ϕ est *h-exact* si même l'égalité tient.

Définition 60 (Exactitude). *Soit $h : \mathbb{S} \rightarrow \Delta$ une Σ -abstraction, ϕ une Σ -formule et $V \supseteq \mathcal{V}(\phi)$. ϕ est dite *h-exacte* selon V si $h(sol_V^{\mathbb{S}}(\phi)) = sol_V^{\Delta}(\phi)$. ϕ est dite *h-exacte* si ϕ est *h-exacte* selon $\mathcal{V}(\phi)$.*

L'objectif suivant est d'étudier la préservation de la *h-exactitude* par les opérateurs logiques. La principale difficulté est le fait que la *h-exactitude* n'est pas préservée par conjonction. Néanmoins, comme il sera montré plus tard, elle est préservée par disjonction et quantification existentielle.

Montrons que la *h-exactitude* est préservée par l'ajout de variables. Pour cela, il faut supposer que l'abstraction h est surjective, ce qui sera le cas de toutes les abstractions d'intérêt.

Lemme 61 (L'extension de variables préserve l'exactitude). *Soit $h : S \rightarrow \Delta$ une Σ -abstraction qui est surjective, $\phi \in \mathcal{F}_\Sigma$ une formule, et $V \supseteq \mathcal{V}(\phi)$. Alors la h -exactitude de ϕ implique la h -exactitude de ϕ selon V .*

Démonstration. Cela suit le fait que les solutions d'une formule peuvent être étendues arbitrairement aux variables qui n'apparaissent pas librement dans la formule, comme évoqué dans la Proposition qui suit.

Proposition 62. *N'importe quel affectation de variables $\sigma : V \rightarrow \Delta$ satisfait $\sigma \in h \circ \text{sol}_V^S(\phi)$ ssi $\sigma_{\upharpoonright \mathcal{V}(\phi)} \in h \circ \text{sol}^S(\phi)$.*

Dans un sens, soit $\sigma \in h \circ \text{sol}_V^S(\phi)$. Alors il existe $\alpha \in \text{sol}_V^S(\phi)$ tel que $\sigma = h \circ \alpha$. Puisque $V \supseteq \mathcal{V}(\phi)$ il découle que $\alpha_{\upharpoonright \mathcal{V}(\phi)} \in \text{sol}^S(\phi)$. De plus $\sigma_{\upharpoonright \mathcal{V}(\phi)} = h \circ \alpha_{\upharpoonright \mathcal{V}(\phi)}$ donc $\sigma_{\upharpoonright \mathcal{V}(\phi)} \in h \circ \text{sol}^S(\phi)$.

Dans l'autre sens, soit $\sigma_{\upharpoonright \mathcal{V}(\phi)} \in h \circ \text{sol}^S(\phi)$. Alors il existe $\alpha \in \text{sol}^S(\phi)$ tel que $\sigma_{\upharpoonright \mathcal{V}(\phi)} = h \circ \alpha$. Pour n'importe quel $y \in V \setminus \mathcal{V}(\phi)$ soit $s_y \in \text{dom}(S)$ tel que $h(s_y) = \sigma(y)$. Une telle valeur existe puisque h est surjective. On définit $\alpha' = \alpha[y/s_y \mid y \in V \setminus \mathcal{V}(\phi)]$. Puisque $V \supseteq \mathcal{V}(\phi)$ il suit que $\alpha' \in \text{sol}_V^S(\phi)$. De plus, $\sigma = h \circ \alpha'$, donc $\sigma \in h \circ \text{sol}_V^S(\phi)$. \square

Pour le cas de la disjonction, une propriété de l'union, qui échoue pour l'intersection, est nécessaire.

Lemme 63. *Soit V un ensemble de variables, R_1 et R_2 un sous-ensemble d'affectations du type $V \rightarrow \text{dom}(S)$ et $h : S \rightarrow \Delta$ une Σ -abstraction. $h \circ (R_1 \cup R_2) = h \circ R_1 \cup h \circ R_2$.*

Démonstration. Ce Lemme vient des équivalences suivantes :

$$\begin{aligned} \beta \in h \circ (R_1 \cup R_2) &\Leftrightarrow \exists \alpha. \alpha \in R_1 \cup R_2 \wedge \beta = h \circ \alpha \\ &\Leftrightarrow \exists \alpha. (\alpha \in R_1 \vee \alpha \in R_2) \wedge \beta = h \circ \alpha \\ &\Leftrightarrow \exists \alpha. (\alpha \in R_1 \wedge \beta = h \circ \alpha) \vee (\alpha \in R_2 \wedge \beta = h \circ \alpha) \\ &\Leftrightarrow \beta \in h \circ R_1 \vee \beta \in h \circ R_2 \\ &\Leftrightarrow \beta \in h \circ R_1 \cup h \circ R_2 \end{aligned}$$

\square

Proposition 64. *La disjonction de formules h -exactes est h -exacte.*

Démonstration. Soit ϕ_1 et ϕ_2 des formules non négatives h -exactes. Soit $V = \mathcal{V}(\phi_1) \cup \mathcal{V}(\phi_2)$. Le Lemme 61 montre que ϕ_1 et ϕ_2 sont aussi h -exactes selon V , c'est-à-dire, pour $i \in \{1, 2\}$:

$$h \circ \text{sol}_V^S(\phi_i) = \text{sol}_V^\Delta(\phi_i)$$

La h -exactitude de la disjonction $\phi_1 \vee \phi_2$ peut à présent être montrée comme suit :

$$\begin{aligned}
h \circ \text{sol}^S(\phi_1 \vee \phi_2) &= h \circ (\text{sol}_V^S(\phi_1) \cup \text{sol}_V^S(\phi_2)) \\
&= h \circ \text{sol}_V^S(\phi_1) \cup h \circ \text{sol}_V^S(\phi_2) \quad \text{par le Lemme 63} \\
&= \text{sol}_V^\Delta(\phi_1) \cup \text{sol}_V^\Delta(\phi_2) \quad \text{par } h\text{-exactitude des } \phi_i \text{ selon } V \\
&= \text{sol}^\Delta(\phi_1 \vee \phi_2)
\end{aligned}$$

□

Lemme 65 (La projection se propage avec l'abstraction). *Pour n'importe quelle Σ -abstraction $h : S \rightarrow \Delta$, sous-ensemble R d'affectations du type $V \rightarrow S$, et variable $x \in \mathcal{V}$: $h \circ \pi_x(R) = \pi_x(h \circ R)$.*

Démonstration. Pour tout $\alpha : V \rightarrow \text{dom}(S)$ on obtient $h \circ \pi_x(\alpha) = h \circ \alpha|_{V \setminus \{x\}} = (h \circ \alpha)|_{V \setminus \{x\}} = \pi_x(h \circ \alpha)$. □

Proposition 66 (La quantification préserve l'exactitude). *Pour n'importe quelle Σ -abstraction surjective $h : S \rightarrow \Delta$ et formule $\exists x.\phi \in \mathcal{F}_\Sigma$, si ϕ est h -exacte alors $\exists x.\phi$ est h -exact.*

Démonstration. Soit ϕ h -exact. Par définition ϕ est h -exact selon V . Puisque h est supposée surjective, le Lemme 61 implique que ϕ est h -exact selon $V \cup \{x\}$ (indépendamment du fait que x apparaît librement ou non dans ϕ). Ainsi :

$$\begin{aligned}
h(\text{sol}^S(\exists x.\phi)) &= h(\pi_x(\text{sol}^S(\phi))) \\
&= \pi_x(h(\text{sol}^S(\phi))) \quad \text{by Lemma 65} \\
&= \pi_x(\text{sol}^\Delta(\phi)) \quad \text{since } \phi \text{ is } h\text{-exact} \\
&= \text{sol}^\Delta(\exists x.\phi)
\end{aligned}$$

□

La partie qui suit étudie la h -exactitude des systèmes fortement triangulaires de Σ -équations, sous la condition que h est une abstraction entre Σ -algèbres avec division unique.

Lemme 67 (Propriété singleton). *Si S est une Σ -algèbre, $e \in \mathcal{E}_\Sigma$, et $\alpha : \mathbf{v}_e \rightarrow S$ un affectation de variables, alors l'ensemble $\llbracket e \rrbracket^{\alpha, S}$ est un singleton.*

Démonstration. Par induction sur la structure des expressions $e \in \mathcal{E}$:

Cas des constantes $c \in \{0, 1\}$. L'ensemble $\llbracket c \rrbracket^{\alpha, S} = \{c^S\}$ est un singleton.

Cas des variables $x \in \mathcal{V}$. L'ensemble $\llbracket x \rrbracket^{\alpha, S} = \{\alpha(x)\}$ est un singleton.

Cas $e_1 \odot e_2$ où $e_1, e_2 \in \mathcal{E}_\Sigma$ et $\odot \in \{+, *\}$.

$$\llbracket e_1 \odot e_2 \rrbracket^{\alpha, S} = \{s \odot^S s' \mid s \in \llbracket e_1 \rrbracket^{\alpha, S}, s' \in \llbracket e_2 \rrbracket^{\alpha, S}\}$$

L'ensemble est un singleton puisque $\llbracket e_1 \rrbracket^{\alpha, S}$ et $\llbracket e_2 \rrbracket^{\alpha, S}$ sont des singletons par hypothèse d'induction. Cela engendre le fait que $s \odot^S s'$ est aussi un singleton puisque S est une Σ -algèbre \square

Une Σ -algèbre est une Σ -structure avec la propriété singleton. Soit ele la fonction qui associe chaque singleton à l'élément qu'il contient.

Définition 68. Une Σ -structure S a la division unique, si elle satisfait la formule du premier-ordre $\forall x. \exists^1 y. ny \doteq x$ pour tout nombre naturel $n \in \mathbb{N} \setminus \{0\}$.

Il est clair que les Σ -algèbres \mathbb{R}_+ et \mathbb{B} ont la division unique. Pour chaque élément s du domaine de la structure S avec la division unique et chaque nombre naturel $n \in \mathbb{N} \setminus \{0\}$, on dénote par $\frac{s}{n}$ l'unique élément de $\{\alpha(y) \mid \alpha \in sol^S(ny \doteq z), \alpha(z) = s\}$.

Lemme 69. Soit $\phi \in \mathcal{F}_\Sigma$ une Σ -formule et S une Σ -algèbre avec la division unique. Pour chaque nombre naturel non nul n , chaque variable $y \notin \mathcal{V}(\phi)$, et expression $e \in \mathcal{E}_\Sigma$ avec $\mathbf{v}_e \subseteq \mathbf{v}_\phi$:

$$sol^S(\phi \wedge ny \doteq e) = \{\alpha[y/\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}] \mid \alpha \in sol^S(\phi)\}$$

Démonstration. On fixe de façon arbitraire $\alpha : \mathbf{v}_\phi \rightarrow dom(S)$. Puisque S est une Σ -algèbre, $\llbracket e \rrbracket^{\alpha, S}$ est un singleton et $\mathbf{v}_e \subseteq \mathbf{V}(\phi)$, $ele(\llbracket e \rrbracket^{\alpha, S})$ est défini de façon unique. De plus, S a la division unique, d'où $\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}$ est un élément de $dom(S)$ bien défini. Ainsi et puisque $y \notin \mathbf{v}_\phi$, $\alpha[y/\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}]$ est l'unique solution de l'équation $ny \doteq e$ qui étend α .

Cas “ \supseteq ”. Soit $\alpha \in sol^S(\phi)$, $y \notin \mathcal{V}(\phi)$, et $\alpha[y/\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}]$ est une solution de $ny \doteq e$, il suit que $\alpha[y/\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}]$ est une solution de $\phi \wedge ny \doteq e$.

Cas “ \subseteq ”. Soit $\alpha \in sol^S(\phi \wedge ny \doteq e)$. Puisque $\alpha[y/\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}]$ est l'unique solution de l'équation $ny \doteq e$ qui étend $\alpha' = \alpha|_{\mathbf{v}_\phi}$, il suit que $\alpha(y) = \frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}$ d'où $\alpha = \alpha'[y/\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}]$ avec $\alpha' \in sol^S(\phi)$. \square

Proposition 70. Soit $\phi \in \mathcal{F}_\Sigma$ une Σ -formule, $n \neq 0$ un nombre naturel, $e \in \mathcal{E}_\Sigma$ une Σ -expression avec $\mathbf{v}_e \subseteq \mathbf{v}_\phi$ et $y \notin \mathbf{v}_\phi$ et la Σ -abstraction $h : S \rightarrow \Delta$ avec S et Δ deux Σ -algèbres avec la division unique. Puis si ϕ est h -exact, cela implique que $\phi \wedge e \doteq ny$ est h -exact.

Idées de la preuve. Il peut être montré que $h(ele(\llbracket e \rrbracket^{\alpha, S})) = ele(\llbracket e \rrbracket^{h\circ\alpha, \Delta})$ et $h(\frac{s}{n}) = \frac{h(s)}{n}$. Donc:

$$\begin{aligned}
h \circ sol^S(\phi \wedge e \doteq ny) &= h \circ \{\alpha[y/\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}] \mid \alpha \in sol^S(\phi)\} && \text{par le Lemme 69} \\
&= \{(h \circ \alpha)[y/h(\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n})] \mid \alpha \in sol^S(\phi)\} && \text{élémentaire} \\
&= \{\sigma[y/h(\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n})] \mid \sigma \in sol^\Delta(\phi)\} && h\text{-exactitude de } \phi \\
&= \{\sigma[y/\frac{h(ele(\llbracket e \rrbracket^{\alpha, S}))}{n}] \mid \sigma \in sol^\Delta(\phi)\} \\
&= \{\sigma[y/\frac{ele(\llbracket e \rrbracket^{h\circ\alpha, \Delta})}{n}] \mid \sigma \in sol^\Delta(\phi)\} \\
&= sol^\Delta(\phi \wedge e \doteq ny) && \text{par le Lemme 69}
\end{aligned}$$

Démonstration. Soit $e \in \mathcal{E}_\Sigma$ une Σ -expression.

Proposition 71. Pour chaque $\alpha : V \rightarrow \mathbb{R}_+$ avec $V \supseteq \mathcal{V}(e)$: $h(ele(\llbracket e \rrbracket^{\alpha, S})) = ele(\llbracket e \rrbracket^{h\circ\alpha, \Delta})$.

Pour chaque $\alpha : \mathcal{V}(\phi) \rightarrow S$ le Théorème 34 sur l'homomorphisme entraîne $h(\llbracket e \rrbracket^{\alpha, S}) \subseteq \llbracket e \rrbracket^{h\circ\alpha, \Delta}$. Puisque S et Δ sont toutes deux des Σ -algèbres, les ensembles $\llbracket e \rrbracket^{\alpha, S}$ et $\llbracket e \rrbracket^{h\circ\alpha, \Delta}$ sont tout deux des singletons par le Lemme 67, ainsi $h(ele(\llbracket e \rrbracket^{\alpha, S})) = ele(\llbracket e \rrbracket^{h\circ\alpha, \Delta})$

Proposition 72. Pour chaque $s \in dom(S)$ et $n \neq 0$ un nombre naturel : $h(\frac{s}{n}) = \frac{h(s)}{n}$.

Puisque S est supposée avoir la division unique, $s' = \frac{s}{n}$ est bien défini comme l'unique élément de $dom(S)$ tel que $\underbrace{s' +^S \dots +^S s'}_n = s$. Ainsi, $h(\underbrace{s' +^S \dots +^S s'}_n) = h(s)$ et puisque h est un homomorphisme, il découle que $\underbrace{h(s') +^\Delta \dots +^\Delta h(s')}_n = h(s)$. Comme Δ est supposée avoir la division unique, cela implique que $h(s') = \frac{h(s)}{n}$.

La Proposition peut à présent être démontrée avec ces deux Propositions. Soit ϕ h -exact, $y \notin \mathcal{V}(\phi)$, et $\mathcal{V}(e) \subseteq \mathcal{V}(\phi)$. Il faut montrer que $\phi \wedge ny \doteq e$ est h -exact aussi:

$$\begin{aligned}
h \circ sol^S(\phi \wedge e \doteq ny) &= h \circ \{\alpha[y/\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n}] \mid \alpha \in sol^S(\phi)\} && \text{par le Lemme 69} \\
&= \{(h \circ \alpha)[y/h(\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n})] \mid \alpha \in sol^S(\phi)\} && \text{élémentaire} \\
&= \{\sigma[y/h(\frac{ele(\llbracket e \rrbracket^{\alpha, S})}{n})] \mid \sigma \in sol^\Delta(\phi)\} && h\text{-exactitude de } \phi \\
&= \{\sigma[y/\frac{h(ele(\llbracket e \rrbracket^{\alpha, S}))}{n}] \mid \sigma \in sol^\Delta(\phi)\} && \text{par la Proposition 72} \\
&= \{\sigma[y/\frac{ele(\llbracket e \rrbracket^{h\circ\alpha, \Delta})}{n}] \mid \sigma \in sol^\Delta(\phi)\} && \text{par la Proposition 71} \\
&= sol^\Delta(\phi \wedge e \doteq ny) && \text{par le Lemme 69}
\end{aligned}$$

□

Proposition 73. *Soit $h : S \rightarrow \Delta$ une Σ -abstraction entre Σ -algèbres avec division unique. Chaque système fortement triangulaire de Σ -équations à coefficients naturels est h -exact.*

Idées de la preuve. Par induction sur le nombre d'équations et la Proposition 70.

Démonstration. Chaque système d'équations fortement triangulaire a la forme $\bigwedge_{i=1}^n e_i \doteq n_i y_i$ où n et $n_i \neq 0$ sont des naturels et y_i est i -frais pour tout $1 \leq i \leq n$. La preuve est par induction sur n . Dans le cas $n = 0$, la conjonction est égal à *true* qui est h -exact car $h(\text{sol}^S(\text{true})) = h(\{\}) = \text{sol}^\Delta(\text{true})$. Dans le cas $n > 0$, par hypothèse d'induction on a $\bigwedge_{j=1}^{i-1} e_j \doteq n_j y_j$ est h -exact. Comme $n_i \neq 0$ avec la Proposition 70 on a $e_i \doteq n_i y_i \wedge \bigwedge_{j=1}^{i-1} e_j \doteq n_j y_j$ est h -exact. □

Notons que la Proposition 73 reste vraie pour les systèmes triangulaires qui ne sont pas fortement triangulaires. A voir dans la prochaine section (Théorème 89 et Proposition 83) car cela requiert différents arguments.

Théorème 74 (Exactitude). *Les systèmes polynomiaux quasi-positifs fortement triangulaires sont $h_{\mathbb{B}}$ -exact.*

Démonstration. Les Σ -algèbres \mathbb{R}_+ et \mathbb{B} ont la division unique, la Proposition 73 peut donc s'appliquer. □

Le Théorème des modes élémentaires 18 montre que n'importe quelle équation matricielle d'entiers $A\mathbf{x} \doteq 0$ est \mathbb{R}_+ -équivalente à un système d'équations linéaire quasi-positif fortement triangulaire. Le Théorème 74 est donc appliqué pour obtenir le corollaire suivant.

Corollaire 75. *Chaque équation matricielle d'entiers peut être convertie en temps au plus exponentiel en une Σ -formule $h_{\mathbb{B}}$ -exacte \mathbb{R}_+ -équivalente.*

Ce corollaire permet de calculer la $h_{\mathbb{B}}$ -abstraction d'une équation matricielle d'entiers en calculant les \mathbb{B} -solutions de la formule \mathbb{R}_+ -équivalente $h_{\mathbb{B}}$ -exacte. Pour calculer les abstractions entre structures sans division unique les résultats doivent être consolidés.

6.4 Invariance de l'abstraction

Le problème principal de ce chapitre est que la conjonction de deux formules h -exactes peut ne pas être h -exacte. La situation change quand en supposant la notion de h -invariance pour au moins une des deux formules.

Définition 76 (Invariance). Soit $h : S \rightarrow \Delta$ une Σ -abstraction et $V \subseteq \mathcal{V}$ un sous-ensemble de variables. Un sous-ensemble R d'affectations de variable de type $V \rightarrow \text{dom}(S)$ est dit h -invariant ssi :

$$\forall \alpha, \alpha' : V \rightarrow \text{dom}(S). (\alpha \in R \wedge h \circ \alpha = h \circ \alpha' \implies \alpha' \in R).$$

Une Σ -formule ϕ est dite h -invariante si son ensemble de solutions $\text{sol}^S(\phi)$ l'est.

L'importance de la notion d'invariance pour l'exactitude de la conjonction – qui est formalisée par la Proposition 83 – vient du Lemme suivant :

Lemme 77. Si soit R_1 ou R_2 est h -invariant alors : $h \circ (R_1 \cap R_2) = h \circ R_1 \cap h \circ R_2$.

Démonstration. Une des inclusions est indépendante de l'invariance.

$$\begin{aligned} h \circ (R_1 \cap R_2) &= \{h \circ \alpha \mid \alpha \in R_1, \alpha \in R_2\} \\ &\subseteq \{h \circ \alpha \mid \alpha \in R_1\} \cap \{h \circ \alpha \mid \alpha \in R_2\} \\ &= h \circ R_1 \cap h \circ R_2 \end{aligned}$$

Pour l'autre, il faut supposer sans perte de généralité que R_1 est h -invariant. Soit $\beta \in h \circ R_1 \cap h \circ R_2$. Alors il existe $\alpha_1 \in R_1$ et $\alpha_2 \in R_2$ tels que $\beta = h \circ \alpha_1 = h \circ \alpha_2$. Par h -invariance de R_1 il résulte que $\alpha_1 \in R_2$. Alors $\alpha_1 \in R_1 \cap R_2$, et ainsi, $\beta \in h \circ (R_1 \cap R_2)$ \square

La suite fournit une caractérisation algébrique de la h -invariance. Étant donné une Σ -abstraction $h : S \rightarrow \Delta$, et un ensemble R d'affectations de variable à $\text{dom}(\Delta)$, il est possible de définir la décomposition gauche de R respectivement à h comme l'ensemble d'affectations de variable à $\text{dom}(S)$ suivant :

$$h \ominus R \stackrel{\text{def}}{=} \{\alpha \mid h \circ \alpha \in R\}$$

Naturellement, $R \subseteq h \ominus (h \circ R)$. L'inclusion inverse caractérise la h -invariance de R .

Lemme 78 (La caractérisation algébrique). Un sous-ensemble R d'affectations de variable du type $V \rightarrow \text{dom}(S)$ est h -invariant pour une Σ -abstraction $h : S \rightarrow \Delta$ ssi $h \ominus (h \circ R) \subseteq R$.

Démonstration. “ \implies ”. Soit R h -invariant et $\alpha \in h \ominus (h \circ R)$. Alors il existe $\alpha' \in R$ tel que $h \circ \alpha = h \circ \alpha'$. La h -invariance de R implique ainsi que $\alpha \in R$.

“ \impliedby ”. Supposons que $h \ominus (h \circ R) \subseteq R$. Soit $\alpha, \alpha' : V \rightarrow \text{dom}(S)$ tels que $h \circ \alpha = h \circ \alpha'$ et $\alpha \in R$. Il faut montrer que $\alpha' \in R$.

Depuis $h \circ \alpha = h \circ \alpha'$ et $\alpha \in R$ il suit que $\alpha' \in h \ominus (h \circ R)$ et donc $\alpha' \in R$ comme requis. \square

Lemme 79 (L'extension de variables préserve l'invariance). *Soit h une abstraction surjective et R un sous-ensemble de fonctions du type $V' \rightarrow \text{dom}(S)$ et V un sous-ensemble de variables disjoint de V' . Si R est h -invariant alors $\text{ext}_V^S(R)$ est h -invariant aussi.*

Démonstration. Ceci découle directement de la caractérisation de la h -invariance du Lemme 78 et les deux Propositions suivantes :

Proposition 80. *Si h est surjective alors $h \circ \text{ext}_V^S(R) = \text{ext}_V^\Delta(h \circ R)$.*

Ceci vient de $h \circ \text{ext}_V^S(R) = \{h \circ \alpha \mid \alpha \in \text{ext}_V^S(R)\} = \text{ext}_V^\Delta(\{h \circ \alpha' \mid \alpha' \in R\})$ où la surjectivité de h est utilisée en dernière étape.

Proposition 81. *$h \circ \text{ext}_V^\Delta(R') = \text{ext}_V^S(h \circ R')$ pour chaque sous-ensemble R' de fonctions du type $V' \rightarrow \text{dom}(\Delta)$.*

$$\begin{aligned} h \circ \text{ext}_V^\Delta(R') &= \{\alpha : V \cup V' \rightarrow \text{dom}(S) \mid h \circ \alpha \in \text{ext}_V^\Delta(R')\} \\ &= \{\alpha : V \cup V' \rightarrow \text{dom}(S) \mid h \circ \alpha|_{V'} \in R'\} \\ &= \text{ext}_V^S(\{\alpha' : V' \rightarrow \text{dom}(S) \mid h \circ \alpha' \in R'\}) \\ &= \text{ext}_V^S(h \circ R') \end{aligned}$$

□

Lemme 82. *Soit $h : S \rightarrow \Delta$ une Σ -abstraction surjective, ϕ une Σ -formule, et $V \supseteq \mathcal{V}(\phi)$. La h -invariance de ϕ entraîne la h -invariance de $\text{sol}_V^S(\phi)$.*

Démonstration. Ceci vient de la cylindrification, Lemme 61, et du fait que l'extension de variables préserve la h -invariance comme montré dans le Lemme 79

□

Proposition 83 (L'exactitude est préservée par la conjonction en présence de l'invariance). *Soit h une Σ -abstraction surjective. Si ϕ_1 et ϕ_2 sont des Σ -formules h -exactes et ϕ_1 ou ϕ_2 est h -invariant alors la conjonction $\phi_1 \wedge \phi_2$ est h -exacte.*

Démonstration. Soit ϕ_1 et ϕ_2 des Σ -formules h -exactes. On suppose sans perte de généralité que ϕ_1 est h -invariante. Soit $V = \mathcal{V}_{\phi_1 \wedge \phi_2}$. Puisque $\mathcal{V}(\phi_2) \subseteq V$ l'ensemble $\text{sol}_V^S(\phi_2)$ est h -invariant aussi par le Lemme 82. On peut montrer que $\phi_1 \wedge \phi_2$ est h -exact comme suit:

$$\begin{aligned} h \circ \text{sol}^S(\phi_1 \wedge \phi_2) &= h \circ (\text{sol}_V^S(\phi_1) \cap \text{sol}_V^S(\phi_2)) \\ &= h \circ \text{sol}_V^S(\phi_1) \cap h \circ \text{sol}_V^S(\phi_2) \quad \text{par le Lemme 77} \\ &= \text{sol}_V^\Delta(\phi_1) \cap \text{sol}_V^\Delta(\phi_2) \quad \text{par } h\text{-exactitude des } \phi_i \text{ selon } V \\ &= \text{sol}^\Delta(\phi_1 \wedge \phi_2) \end{aligned}$$

□

L'objectif suivant est de montrer que les formules h -invariantes sont closes par conjonction, disjonction, négation et quantification existentielle. Les deux premières propriétés viennent des deux propriétés algébriques de la décomposition de l'abstraction qui suivent.

Lemme 84. *Pour chaque Σ -abstraction $h : S \rightarrow \Delta$, chaque sous-ensemble d'affectations du type $V \rightarrow \text{dom}(S)$ R_1 et R_2 , et V un sous-ensemble de variables:*

- $h \ominus (R_1 \cap R_2) = h \ominus R_1 \cap h \ominus R_2$.
- $h \ominus (R_1 \cup R_2) = h \ominus R_1 \cup h \ominus R_2$.

Démonstration. Le cas de l'union vient directement des définitions:

$$\begin{aligned} h \ominus (R_1 \cup R_2) &= \{\alpha \mid h \circ \alpha \in R_1 \cup R_2\} \\ &= \{\alpha \mid h \circ \alpha \in R_1 \vee h \circ \alpha \in R_2\} \\ &= \{\alpha \mid h \circ \alpha \in R_1\} \cup \{\alpha \mid h \circ \alpha \in R_2\} \\ &= h \ominus R_1 \cup h \ominus R_2 \end{aligned}$$

Le cas de l'intersection est symétrique:

$$\begin{aligned} h \ominus (R_1 \cap R_2) &= \{\alpha \mid h \circ \alpha \in R_1 \cap R_2\} \\ &= \{\alpha \mid h \circ \alpha \in R_1 \wedge h \circ \alpha \in R_2\} \\ &= \{\alpha \mid h \circ \alpha \in R_1\} \cap \{\alpha \mid h \circ \alpha \in R_2\} \\ &= h \ominus R_1 \cap h \ominus R_2 \end{aligned}$$

□

Lemme 85 (L'intersection et l'union préservent l'invariance). *Soit $h : S \rightarrow \Delta$ une Σ -abstraction. Alors l'intersection et l'union de deux sous-ensembles h -invariants R_1 et R_2 d'affectations de variable du type $V \rightarrow \text{dom}(S)$ est h -invariant.*

Démonstration. Ceci découle de la caractérisation algébrique issue du Lemme 78 de l'invariance, En combinaison avec les propriétés algébriques de la composition et de la décomposition donnée par les Lemmes 63, 77, et 84

□

Lemme 86 (La projection se propage avec la décomposition gauche).

$$h \ominus \pi_x(R) = \pi_x(h \ominus R)$$

Démonstration. Pour tout $\alpha : V \rightarrow \text{dom}(\Delta)$ on a

$$\begin{aligned} h \ominus \pi_x(\alpha) &= h \ominus \alpha|_{V \setminus \{x\}} \\ &= (h \ominus \alpha)|_{V \setminus \{x\}} \\ &= \pi_x(h \ominus \alpha) \end{aligned}$$

□

Proposition 87 (L'invariance est préservée par conjonction, disjonction, et quantification). *Si h est une abstraction surjective alors la classe de formules du premier ordre h -invariantes est close sous conjonction, disjonction, et quantification existentielle.*

Démonstration. Soit $h : S \rightarrow \Delta$ une Σ -abstraction.

Cas de la conjonction: Soit ϕ_1 et ϕ_2 h -invariants et $V = \forall_{\phi_1 \wedge \phi_2}$. Par le Lemme 82 les ensembles $sol_V^S(\phi_1)$ et $sol_V^S(\phi_2)$ sont tout deux h -invariants, et par le Lemme 85 leur intersection l'est aussi. Ainsi :

$$\begin{aligned} h \ominus (h \circ sol^S(\phi_1 \wedge \phi_2)) &= h \ominus (h \circ (sol_V^S(\phi_1) \cap sol_V^S(\phi_2))) \\ &\subseteq sol_V^S(\phi_1) \cap sol_V^S(\phi_2) && \text{par } h\text{-invariance} \\ &= sol^S(\phi_1 \wedge \phi_2) && \text{et Lemme 78} \end{aligned}$$

Par le Lemme 78 dans l'autre sens, cela implique que $\phi_1 \wedge \phi_2$ est h -invariant.

Cas de la disjonction: Analogue au cas de la conjonction.

Cas de la quantification existentielle:

$$\begin{aligned} h \ominus (h \circ sol^S(\exists x.\phi_1)) &= h \ominus (h \circ \pi_x(sol^S(\phi_1))) \\ &= h \ominus (\pi_x(h \circ sol^S(\phi_1))) && \text{par le Lemme 65} \\ &= \pi_x(h \ominus (h \circ sol^S(\phi_1))) && \text{par le Lemme 86} \\ &\subseteq \pi_x(sol^S(\phi_1)) && \text{par } h\text{-invariance de } \phi_1 \\ &= sol^S(\exists x.\phi_1) && \text{et le Lemme 78} \end{aligned}$$

Par le Lemme 78, cela implique que $\exists x.\phi_1$ est h -invariant

□

Proposition 88. *Soit h une Σ -abstraction surjective. La classe de Σ -formulas h -exactes et h -invariantes est close par la conjonction, disjonction et quantification existentielle.*

Démonstration. La clôture de la conjonction vient des Propositions 83 et 87, celle de la disjonction vient des Propositions 64 et 87, et la quantification existentielle vient des Propositions 66 et 87. □

Théorème 89 (Invariance). *Chaque équation polynomiale positive $p \doteq 0$ telle que p n'a pas de terme constant est $h_{\mathbb{B}}$ -exacte et $h_{\mathbb{B}}$ -invariante.*

Démonstration. Considérons une équation polynomiale positive $p \doteq 0$ telle que p n'a pas de terme constant et uniquement des coefficients positifs. Ainsi p a la forme $\sum_{j=1}^l n_j \prod_{k=1}^{i_j} x_{j,k}^{m_{j,k}} \doteq 0$ where $l \geq 0$, et $n_j, i_j, m_{j,k} > 0$.

Proposition 90. *Pour les deux algèbres $S \in \{\mathbb{B}, \mathbb{R}_+\}$: $sol^S(p \doteq 0) = sol^S(\bigwedge_{j=1}^l \bigvee_{k=1}^{i_j} x_{j,k} \doteq 0)$.*

Le polynome a une valeur 0 ssi tout ses monomes sont nuls, d'où : $\prod_{k=1}^{i_j} x_{j,k}^{m_{j,k}} = 0$ pour tout $1 \leq j \leq l$. Puisque les termes constants sont exclus, nous avons $i_j \neq 0$. De plus, nous assumons pour tout polynomes que $m_{j,k} \neq 0$. Alors pour tout $1 \leq j \leq l$ il doit exister $1 \leq k \leq i_j$ tel que $x_{j,k} = 0$.

Proposition 91. *L'équation $x \doteq 0$ est $h_{\mathbb{B}}$ -exacte et $h_{\mathbb{B}}$ -invariante.*

Ceci découle directement des définitions. Avec ces deux Propositions nous sommes en position pour prouver le Lemme. Puisque la classe de formules $h_{\mathbb{B}}$ -exactes et $h_{\mathbb{B}}$ -invariantes est close par conjonction et disjonction d'après la Proposition 88, il suit de la Proposition 91 que $\bigwedge_{j=1}^l \bigvee_{k=1}^{i_j} x_{j,k} \doteq 0$ est $h_{\mathbb{B}}$ -exact et $h_{\mathbb{B}}$ -invariant. Puisque cette formule est équivalente sur \mathbb{R}_+ à l'équation polynomiale d'après la Proposition 90, la $h_{\mathbb{B}}$ -invariance couvre $p \doteq 0$. La $h_{\mathbb{B}}$ -exactitude elle aussi la couvre en se basant sur l'équivalence pour les deux structures \mathbb{R}_+ et \mathbb{B} :

$$\begin{aligned} h_{\mathbb{B}} \circ sol^{\mathbb{R}_+}(p \doteq 0) &= h_{\mathbb{B}} \circ sol_{\mathbb{V}^+}^{\mathbb{R}_+}(\bigwedge_{j=1}^l \bigvee_{k=1}^{i_j} x_{j,k} \doteq 0) && \text{par la Proposition 90 pour } \mathbb{R}_+ \\ &= sol^{\mathbb{B}}(\bigwedge_{j=1}^l \bigvee_{k=1}^{i_j} x_{j,k} \doteq 0) && \text{par } h_{\mathbb{B}} \text{ exactitude} \\ &= sol^{\mathbb{B}}(p \doteq 0) && \text{par la Proposition 90 pour } \mathbb{B}. \end{aligned}$$

□

6.5 Abstraction booléenne de systèmes mixtes

Dans cette section, nous prouvons notre principal résultat qui est de savoir comment calculer exactement la $h_{\mathbb{B}}$ -abstraction d'un ensemble de \mathbb{R}_+ -solutions de systèmes mixtes.

Définition 92. *Un système mixte est une formule dans \mathcal{F}_{Σ} de la forme $\exists z. \phi \wedge \phi'$ où ϕ est un système linéaire d'équation et ϕ' une formule du premier ordre $h_{\mathbb{B}}$ -invariante et $h_{\mathbb{B}}$ -exacte.*

A noter que les systèmes d'équations linéaires $A\mathbf{y} \doteq 0$, avec A une matrice d'entiers et \mathbf{y} une séquence de variables une à une distinctes, n'est pas forcément $h_{\mathbb{B}}$ -exact, si A n'est pas positive. Cependant, n'importe quel système linéaire d'équations de cette forme est \mathbb{R}_+ -équivalent à un système polynomial quasi-positif fortement triangulaire, comme montré dans le Théorème Modes élémentaires 18. Et un système polynomial quasi-positif fortement triangulaire est prouvé être $h_{\mathbb{B}}$ -exact par le Théorème Exactitude 74.

Théorème 93 (Principal). *Chaque système mixte peut être converti en temps exponentiel en une Σ -formule \mathbb{R}_+ -équivalente qui est $h_{\mathbb{B}}$ -exacte.*

Démonstration. Considérons un système mixte $\exists \mathbf{x}. (\phi \wedge \phi')$ où ϕ est un système linéaire d'équations et ϕ' une formule du premier-ordre qui est $h_{\mathbb{B}}$ -exacte et $h_{\mathbb{B}}$ -invariante. Selon le Théorème Modes élémentaires 18, le système linéaire d'équations ϕ peut être transformé en temps exponentiel en formule $\exists \mathbf{z}. \phi''$ où ϕ'' est un système d'équation linéaires quasi-positif et fortement triangulaire. De tels systèmes d'équations polynomiales sont $h_{\mathbb{B}}$ -exact par le Théorème 74, donc ϕ'' aussi. La Proposition Invariance 83 montre que la conjonction $\phi'' \wedge \phi'$ est $h_{\mathbb{B}}$ -exacte aussi, puisque ϕ' était supposée être $h_{\mathbb{B}}$ -exacte and $h_{\mathbb{B}}$ -invariante. Finalement, la $h_{\mathbb{B}}$ -exactitude est préservée par la quantification existentielle d'après la Proposition 66, donc la formule $\exists \mathbf{x}. \exists \mathbf{z}. \phi'' \wedge \phi'$ est $h_{\mathbb{B}}$ -exacte aussi. \square

Corollaire 94. *La $h_{\mathbb{B}}$ -abstraction d'un ensemble de \mathbb{R}_+ -solutions d'un système mixte ϕ , noté $h_{\mathbb{B}} \circ \text{sol}^{\mathbb{R}_+}(\phi)$, peut être calculé en temps au plus exponentiel en la taille du système ϕ .*

Démonstration. Etant donné un système mixte ϕ , nous pouvons appliquer le Théorème 93 pour calculer en temps au plus exponentiel une formule \mathbb{R}_+ -équivalente ϕ'' qui est $h_{\mathbb{B}}$ -exacte. Il est ensuite suffisant de calculer $\text{sol}^{\mathbb{B}}(\phi'')$ en temps exponentiel en la taille de ϕ . Cela peut être fait de façon naïve, qui est d'évaluer la formule ϕ'' – qui peut être de taille exponentielle – sur l'ensemble des affectations de variables booléens possible – qui peuvent être exponentiel dans leur nombre. Pour chaque affectation l'évaluation peut être faite dans le PSPACE et cela en temps exponentiel. L'ensemble du temps requit est ainsi un produit de deux exponentiels, ce qui reste exponentiel \square

L'algorithme de la preuve du Corollaire 94 peut être amélioré pour être suffisamment efficace en pratique. Pour cela, les deux étapes avec une complexité exponentielle dans le pire des cas doivent être rendues polynomiales pour les instances particulières. A noter d'abord que le calcul des modes élémentaires (Théorème 18) est connu pour être faisable par ordinateur. Plusieurs algorithmes dans ce but ont été implémentés (Avis and Jordan, 2018 ; Bagnara et al., 2008 ; Fukuda, 1993 ; Klamt et al., 2003) et appliqués avec succès à des problèmes de biologie des systèmes (Gagneur and Klamt, 2004). La seconde étape exponentielle concerne l'énumération de tout les affectations de variables booléens. Cette énumération peut être évitée en utilisant des techniques de programmation par contraintes pour calculer l'ensemble de solutions $\text{sol}^{\mathbb{B}}(\phi'')$. Pour ces systèmes mixtes pour lesquels les deux étapes peuvent être faites en temps polynomial, nous pouvons calculer

l'abstraction booléenne de l'ensemble de solutions en temps polynomial aussi. La faisabilité en pratique de cette approche a été démontré récemment avec l'application pour la prédiction de knock-outs en biologie des systèmes (Allart et al., 2019), pour laquelle auparavant seule une sur-approximations pouvait être calculée.

6.6 Calcul de l'abstraction de signe

Dans cette section, nous montrons comment calculer l'abstraction de signe $h_{\mathbb{S}} \circ \text{sol}^{\mathbb{R}}(\phi)$ de l'ensemble de \mathbb{R} -solutions d'un système d'équations linéaires ϕ . Pour cela, ϕ est convertie en une formule du premier-ordre ϕ' selon le Théorème principal 93. Ainsi $h_{\mathbb{S}} \circ \text{sol}^{\mathbb{R}}(\phi)$ peut être calculé par $\text{sol}^{\mathbb{B}}(\phi')$ en temps polynomial. Dans cette optique, nous faisons d'abord le lien entre l'abstraction de signe et l'abstraction booléenne, et montrons dans un second temps que cette relation peut être définie en logique du premier ordre de façon à pouvoir appliquer le Théorème principal pour l'abstraction booléenne.

Dans un premier temps, relier l'abstraction de signe à l'abstraction booléenne. Pour cela, chaque nombre réel r est décomposé en deux nombres positifs, sa partie négative $\ominus(r)$ et sa partie positive $\oplus(r)$, tel que si $r \geq 0$ alors $\ominus(r) = 0$ et $\oplus(r) = r$, et sinon $\ominus(r) = -r$ et $\oplus(r) = 0$. La fonction de décomposition $\text{dec} : \mathbb{R} \rightarrow (\{0\} \times \mathbb{R}_+) \cup (\mathbb{R}_+ \times \{0\})$ est définie ci-dessous avec $r \in \mathbb{R}$:

$$\text{dec}(r) = (\ominus(r), \oplus(r))$$

Cette fonction est une bijection, elle a donc une fonction inverse $\text{dec}^{-1} : (\{0\} \times \mathbb{R}_+) \cup (\mathbb{R}_+ \times \{0\}) \rightarrow \mathbb{R}$, qui satisfait $\text{dec}^{-1}((r_1, r_2)) = r_2 - r_1$ pour toute paire (r_1, r_2) dans son domaine.

Lemme 95 (Décomposition). $h_{\mathbb{S}} = \text{dec}^{-1} \circ h_{\mathbb{B}}^2 \circ \text{dec}$

Démonstration. Si r est négatif alors $\text{dec}^{-1}(h_{\mathbb{B}}^2(\text{dec}(r))) = \text{dec}^{-1}(h_{\mathbb{B}}^2((-r, 0))) = \text{dec}^{-1}((h_{\mathbb{B}}(-r), 0)) = -h_{\mathbb{B}}(-r) = h_{\mathbb{S}}(r)$. Sinon si r est positif alors $\text{dec}^{-1}(h_{\mathbb{B}}^2(\text{dec}(r))) = \text{dec}^{-1}(h_{\mathbb{B}}^2((0, r))) = \text{dec}^{-1}((0, h_{\mathbb{B}}(r))) = h_{\mathbb{B}}(r) = h_{\mathbb{S}}(r)$

□

Nous allons d'abord montrer que les formules du premier ordre sur les réels peuvent être réécrites de telle sorte que leur interprétation sur les réels positifs est suffisante. Commençons avec une définition de la positivité des réels en logique du premier ordre. Pour chaque variable $x \in \mathcal{V}$ nous définissons la formule $\text{pos}(x) \in \mathcal{F}_{\Sigma}$ par :

$$\text{pos}(x) =_{\text{def}} \exists z. x \doteq z * z$$

Clairement, si $\alpha \in \text{sol}^{\mathbb{R}}(\text{pos}(x))$ alors $\alpha(x) \in \mathbb{R}_+$. Cette formule peut être utilisée pour relier les \mathbb{R}_+ -solutions aux \mathbb{R} -solutions de formules particulières.

Définition 96. Une formule $\phi \in \mathcal{F}_\Sigma$ est dite à domaine positif si ϕ a la forme $\phi' \wedge \bigwedge_{y \in \mathcal{V}(\phi')} pos(y)$ de même pour toute formule ϕ'' pour laquelle $\exists x.\phi''$ est une sous-formule de ϕ .

Lemme 97. Toute formule à domaine positif $\phi \in \mathcal{F}_\Sigma$ satisfait $sol^{\mathbb{R}^+}(\phi) = sol^{\mathbb{R}}(\phi)$.

Démonstration. Par induction sur la structure des formules. L'étape d'induction est directement due au fait que $pos(y)$ impose la positivité de la valeur de y . \square

Nous montrons ensuite comment faire des formules du premier ordre à domaine positif en se basant sur la décomposition de nombres réels en deux nombres positifs présentée auparavant. Nous fixons deux générateurs de variables fraîches $v_\ominus, v_\oplus : \mathcal{V} \rightarrow \mathcal{V}$. Pour chaque $x \in \mathcal{V}$, l'intention est que $v_\oplus(x)$ représente la partie positive de x et $v_\ominus(x)$ sa partie négative. Nous conserverons les invariants que $x = v_\oplus(x) - v_\ominus(x)$ et $v_\oplus(x) * v_\ominus(x) = 0$. De plus, soit $v : \mathcal{V} \rightarrow \mathcal{V}^2$ tel que $v(x) = (v_\oplus(x), v_\ominus(x))$ pour tout $x \in \mathcal{V}$.

Proposition 98 (Positivité). Pour chaque formule $\phi \in \mathcal{F}_\Sigma$ nous pouvons calculer en temps linéaire $dec_v(\phi) \in \mathcal{F}_\Sigma$ tel que :

$$dec \circ sol^{\mathbb{R}}(\phi) = \{\alpha^2 \circ v|_{\mathcal{V}(\phi)} \mid \alpha \in sol^{\mathbb{R}^+}(dec_v(\phi) \wedge \bigwedge_{x \in \mathcal{V}(\phi)} v_\oplus(x) * v_\ominus(x) \doteq 0)\}$$

De plus, si ϕ est un système d'équations linéaires, alors $dec_v(\phi)$ est un système mixte.

Démonstration. Nous pouvons supposer sans perte de généralité que toutes les équations de ϕ sont aplaties, c'est-à-dire, de la forme $x \doteq x_1 + x_2$, $x \doteq x_1 * x_2$, $x \doteq 0$, ou $x \doteq 1$.

Nous définissons les formules $dec_v(\phi)$, $dec'_v(\phi) \in \mathcal{F}_\Sigma$ pour toute formule ϕ dans \mathcal{F}_Σ avec des équations aplaties récursivement comme ceci :

$$\begin{aligned} dec_v(\exists x.\phi) &=_{\text{def}} \exists v_\ominus(x).\exists v_\oplus(x). dec_v(\phi) \wedge v_\oplus(x) * v_\ominus(x) \doteq 0 \\ &\quad \wedge pos(v_\oplus(x)) \wedge pos(v_\ominus(x)) \\ dec_v(\phi \wedge \phi') &=_{\text{def}} dec_v(\phi) \wedge dec_v(\phi') \\ dec_v(\neg\phi) &=_{\text{def}} \neg dec_v(\phi) \\ dec_v(x \doteq x_1 + x_2) &=_{\text{def}} v_\oplus(x) + v_\ominus(x_1) + v_\ominus(x_2) \\ &\quad \doteq v_\ominus(x) + v_\oplus(x_1) + v_\oplus(x_2) \\ dec_v(x \doteq x_1 * x_2) &=_{\text{def}} v_\oplus(x) + v_\oplus(x_1) * v_\ominus(x_2) + v_\ominus(x_1) * v_\oplus(x_2) \\ &\quad \doteq v_\ominus(x) + v_\oplus(x_1) * v_\oplus(x_2) + v_\ominus(x_1) * v_\ominus(x_2) \\ dec_v(x \doteq 0) &=_{\text{def}} v_\oplus(x) \doteq v_\ominus(x) \\ dec_v(x \doteq 1) &=_{\text{def}} v_\oplus(x) \doteq v_\ominus(x) + 1 \end{aligned}$$

La réécriture pour les équations avec multiplication s'appuie sur la loi de distributivité. Nous utilisons aussi les inverses pour l'addition dans la structure des réels. Soit

$$\text{dec}'_v(\phi) = \text{dec}_v(\phi) \wedge \bigwedge_{x \in \mathcal{V}(\phi)} v_{\oplus}(x) * v_{\ominus}(x) \doteq 0 \wedge \text{pos}(v_{\ominus}(x)) \wedge \text{pos}(v_{\oplus}(x)).$$

Nous pouvons montrer par induction sur la structure des formules que toutes formules $\text{dec}'_v(\phi)$ sont à domaine positif et satisfont $\text{dec} \circ \text{sol}^{\mathbb{R}}(\phi) = \{\alpha^2 \circ v_{\uparrow \mathcal{V}(\phi)} \mid \alpha \in \text{sol}^{\mathbb{R}}(\text{dec}'_v(\phi))\}$. La Proposition vient, avec \mathbb{R}_+ au lieu de \mathbb{R} , du Lemme 97 et de la positivité de domaine de $\text{dec}'_v(\phi)$

□

Théorème 99 (Calcul de l'abstraction de signe). *Pour chaque système d'équation linéaire $\phi \in \mathcal{F}_{\Sigma}$, il est possible de calculer en temps au plus exponentiel une formule $\phi' \in \mathcal{F}_{\Sigma}$ telle que :*

$$h_{\mathbb{S}} \circ \text{sol}^{\mathbb{R}}(\phi) = \{[y/\beta(v_{\oplus}(y)) - \beta(v_{\ominus}(y)) \mid y \in \mathcal{V}(\phi')] \mid \beta \in \text{sol}^{\mathbb{B}}(\phi')\}$$

Démonstration. Soit $\phi \in \mathcal{F}_{\Sigma}$ un système d'équations linéaires. La formule $\text{dec}_v(\phi) \wedge \bigwedge_{x \in \mathcal{V}(\phi)} v_{\oplus}(x) * v_{\ominus}(x) \doteq 0$ est un système mixte par la Proposition de positivité 98, nous pouvons donc appliquer le Théorème principal 93. Ainsi, nous pouvons calculer en temps au plus exponentiel une formule \mathbb{R}_+ -équivalente ϕ' qui est $h_{\mathbb{B}}$ -exacte.

$$\begin{aligned} & h_{\mathbb{S}} \circ \text{sol}^{\mathbb{R}}(\phi) \\ &= \text{dec}^{-1} \circ h_{\mathbb{B}}^2 \circ \text{dec} \circ \text{sol}^{\mathbb{R}}(\phi) && \text{par le Lemme 95} \\ &= \text{dec}^{-1} \circ h_{\mathbb{B}}^2 \circ \{\alpha^2 \circ v_{\uparrow \mathcal{V}(\phi)} \mid \alpha \in \text{sol}^{\mathbb{R}_+}(\text{dec}_v(\phi) \wedge \\ & \quad \bigwedge_{x \in \mathcal{V}(\phi)} v_{\oplus}(x) * v_{\ominus}(x) \doteq 0)\} && \text{par la Proposition 98} \\ &= \text{dec}^{-1} \circ h_{\mathbb{B}}^2 \circ \{\alpha^2 \circ v_{\uparrow \mathcal{V}(\phi')} \mid \alpha \in \text{sol}^{\mathbb{R}_+}(\phi')\} && \text{avec } \phi' \text{ du Théorème 93 principal} \\ &= \{\text{dec}^{-1} \circ \beta^2 \circ v_{\uparrow \mathcal{V}(\phi')} \mid \beta \in \text{sol}^{\mathbb{B}}(\phi')\} && \text{par } h_{\mathbb{B}}\text{-exactitude de } \phi' \\ &= \{[y/\beta(v_{\oplus}(y)) - \beta(v_{\ominus}(y)) \mid y \in \mathcal{V}(\phi')] \mid \beta \in \text{sol}^{\mathbb{B}}(\phi')\} && \text{par définition de } \text{dec}^{-1} \end{aligned}$$

□

Calcul exact des abstractions de différences

Ce chapitre montre comment les problèmes d'abstraction de différences de système d'équations linéaires sur les domaines Δ_3 et Δ_6 sont traduits en questions d'abstraction booléenne de systèmes enrichis mixtes. Le matériel présenté au chapitre précédent permet ainsi de montrer qu'il est possible de calculer ces abstractions exactement. Cela permettra de présenter les algorithmes exactes pour Δ_3 et Δ_6 .

Sommaire

7.1	Introduction	137
7.2	Exact Δ_3-abstraction	138
7.2.1	Systèmes linéaires	139
7.2.2	Les Structures Mixtes	141
7.2.3	Avec contraintes de différences	141
7.3	Exact Δ_6-abstraction	144
7.3.1	Systèmes linéaires	144
7.3.2	Avec contraintes de différences	147

7.1 Introduction

Les heuristiques qui ont été proposées auparavant, et celle présentée dans le chapitre 8, ne contiennent pas toutes les conséquences avec le système abstrait,

c'est-à-dire qu'elles ne calculent pas l'abstraction exacte du système linéaire en général.

Intuitivement, cela se produit car l'approche prend pour acquis que le raisonnement abstrait est basé sur le système linéaire à l'état stationnaire, ce qui nous ramène à l'équation matricielle $A\mathbf{x} = 0$ où A est la matrice de stœchiométrie associée au réseau métabolique, et \mathbf{x} est l'ensemble des flux métaboliques représentant les variables du système. Cependant, il est facile de noter que dès que les différences concrètes sont introduites dans le raisonnement, il n'y a plus un, mais bien 2 systèmes linéaires à considérer: un *avant* les changements d'environnement, et un *après*. Ce qui donne un système pour chaque valeur des pairs représentant un changement concret. Pour faire simple, nous devons considérer une grande équation matricielle incluant les deux systèmes $A\mathbf{x}^{\text{avant}} = 0$ et $A\mathbf{x}^{\text{après}} = 0$.

Cette idée est le point de départ du développement de la contribution majeure de ce thèse présentée dans ce chapitre. Nous présentons donc dans ce chapitre, une méthode pour le calcul exacte de l'abstraction d'un système linéaire, que nous appelons l'algorithme *exact*.

Dans ces algorithmes, propre à Δ_3 et Δ_6 , il s'agit de caractériser les abstractions de différence de l'ensemble de solutions d'un système linéaire d'équations. On montre comment le caractériser par l'ensemble de solutions de formules du premier-ordre interprétées sur la structure finie \mathbb{B} . Ne sachant pas comment trouver une formule h_{Δ} -exacte et équivalente comme celle fournit par le Théorème 93 pour le cas de l'abstraction booléenne, nous utilisons ce Théorème pour trouver un caractérisation booléenne finie des abstractions de différence. Pour ce faire, nous allons fortement nous appuyer sur les propriétés de définition dans la logique du premier-ordre des tuples, commençons donc par l'introduire.

Cette méthode donne une bonne mesure pour étudier la qualité de nos heuristiques. Il sera intéressant de voir que l'heuristique présentée par la suite et cette méthode exacte, repose toutes les deux sur la réécriture d'un système linéaire via l'utilisation de leurs modes élémentaires. La différence clé entre les deux méthodes réside sur le choix du système linéaire utilisé initialement pour calculer ces modes.

7.2 Exact Δ_3 -abstraction

Nous commençons avec la Δ_3 -abstraction, nous donnons en premier lieu les différentes étapes menant à sa caractérisation par un ensemble de solutions de formules du premier-ordre interprétées sur la structure finie \mathbb{B} . Puis nous présentons la logique mixte $\mathbb{B} \cup \Delta_3$ qui permet de traiter également les contraintes cibles additionnelles. Ainsi, nous pourrions présenter le théorème présentant l'algorithme capable de résoudre l'abstraction des différences avec contraintes pour h_{Δ_3} .

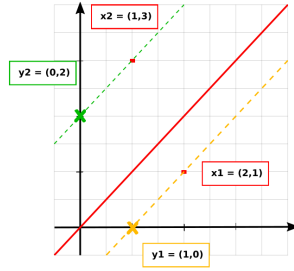


FIGURE 7.1 – Deux exemples de projection sur la diagonale: $y_1 = \text{proj}_G^{\mathbb{R}_+^2}(x_1)$ et $y_2 = \text{proj}_G^{\mathbb{R}_+^2}(x_2)$.

7.2.1 Systèmes linéaires

Nous décomposons d'abord l'abstraction h_{Δ_3} en l'abstraction booléenne $h_{\mathbb{B}}$ et la projection sur la diagonale en \mathbb{R}_+^2 définie par le système mixte dans \mathcal{F}_{Σ}^2 , contenant une équation non positive linéaire et une équation à produit nul, qui est donc non linéaire mais positif comme défini ci-dessous. Pour chaque paire de variables $x, y \in \mathcal{V}$ on définit:

$$\text{proj}_G(x, y) \stackrel{\text{def}}{=} \dot{\pi}_1(x) + \dot{\pi}_2(y) \doteq \dot{\pi}_2(x) + \dot{\pi}_1(y) \wedge \dot{\pi}_1(y) * \dot{\pi}_2(y) \doteq 0$$

La fonction de projection sur la diagonale $\text{proj}_G^{\mathbb{R}_+^2}$ est illustrée de façon géométrique sur la FIGURE 7.1. La valeur de y détermine le signe et la distance de la valeur de x par rapport à la diagonale.

Pour les points sous la diagonale leur distance est prise avec la projection horizontale, et pour les points au-dessus de la diagonale la distance est prise verticalement. Pour chaque solution $\alpha \in \text{sol}^{\mathbb{R}_+^2}(\text{proj}_G(x, y))$, certains composants de $\alpha(y)$ doivent être égal à zéro puisque $\pi_1(\alpha(y)) * \pi_2(\alpha(y)) = 0$. Les autres composants doivent être égal à $|\pi_1(\alpha(x)) - \pi_2(\alpha(x))|$ puisque $\pi_1(\alpha(x)) - \pi_2(\alpha(x)) = \pi_1(\alpha(y)) - \pi_2(\alpha(y))$. Ainsi :

$$\begin{aligned} \text{proj}_G^{\mathbb{R}_+^2} &= \{(r, r'), (0, r' - r) \mid r \leq r'\} \\ &\cup \{(r, r'), (r - r', 0) \mid r \geq r'\} \end{aligned}$$

Lemme 100. $\text{proj}_G^{\mathbb{R}_+^2}$ est une fonction totale du type $\mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2$ qui satisfait $h_{\Delta_3} = h_{\mathbb{B}}^2 \circ \text{proj}_G^{\mathbb{R}_+^2}$.

Démonstration. Par définition $\text{proj}_G^{\mathbb{R}_+^2}$ est une relation binaire sur \mathbb{R}_+^2 . Cette relation binaire est une fonction totale qui satisfait l'équation du Lemme comme le décrit l'équation qui précède le Lemme. □

Pour chaque définition du premier-ordre $G : \mathcal{V}^m \rightarrow \mathcal{F}_\Sigma^2$ nous avons défini dans la Section 3.5 une définition du premier-ordre $proj_G^m(G) : \mathcal{V}^m \rightarrow \mathcal{F}_\Sigma^2$ qui décrit l'application de fonction définie par $proj_G$ aux m composants de la relation définie par G .

Lemme 101. *Pour chaque définition du premier-ordre $G : \mathcal{V}^m \rightarrow \mathcal{F}_\Sigma^2$ et séquence $\mathbf{y} \in \mathcal{V}^m$:*

$$proj_G^{\mathbb{R}_+^2} \circ 2\text{-sol}^{\mathbb{R}_+}(G(\mathbf{y})) = 2\text{-sol}^{\mathbb{R}_+}(proj_G^m(G)(\mathbf{y}))$$

Démonstration. Ce Lemme est une conséquence du fait que $proj_G^{\mathbb{R}_+^2}$ définit une fonction totale par le Lemme 100 et une propriété générale des définitions du premier-ordre énoncé par la Proposition 40 de la Section 3.5. Pour assouplir la lecture de cette thèse nous trouverons cette section en fin de chapitre. Pour l'application de la Proposition 40 nous choisissons $F = proj_G : \mathcal{V}^2 \rightarrow \mathcal{F}_\Sigma^2$, $\ell = 1$, $k = 1$, $n = 2$.

□

Fixons deux générateurs de variables fraîches $\nu_1, \nu_2 : \mathcal{V} \rightarrow \mathcal{V}$ et définissons $\nu(x)$ et $\nu^{-1}(x)$ comme il a été fait auparavant.

Théorème 102. *Pour chaque formule linéaire $L(\mathbf{y}) \in \mathcal{F}_\Sigma$ avec variables libres $\{\mathbf{y}\}$, on peut calculer en temps au plus exponentiel une formule conjonctive positive avec quantificateurs existentiels $\phi(\nu(\mathbf{y})) \in \mathcal{F}_\Sigma$ et variables libres $\{\nu(\mathbf{y})\}$ telle que :*

$$h_{\Delta_3} \circ \text{diff}(sol^{\mathbb{R}_+}(L(\mathbf{y}))) = \nu^{-1}(sol^{\mathbb{B}}(\phi(\nu(\mathbf{y}))))$$

Démonstration. Le temps pour calculer $\phi(\nu(\mathbf{y}))$ est dominé par le temps de calcul des modes élémentaires, qui peut être fait en temps au plus exponentiel.

	$h_{\Delta_3} \circ \text{diff}(sol^{\mathbb{R}_+}(L(\mathbf{y})))$
<i>Proposition 31</i>	$= h_{\Delta_3} \circ sol^{\mathbb{R}_+^2}(L(\mathbf{y}))$
<i>Pair FO Proposition 22</i>	$= h_{\Delta_3} \circ 2\text{-sol}^{\mathbb{R}_+}(L_2(\mathbf{y}))$ with $L_2(\mathbf{y}) = \langle L(\mathbf{y}) \rangle^2$
<i>Decomposition Lemma 100</i>	$= h_{\mathbb{B}}^2 \circ proj_G^{\mathbb{R}_+^2} \circ 2\text{-sol}^{\mathbb{R}_+}(L_2(\mathbf{y}))$
<i>FO-Definition Lemma 101</i>	$= h_{\mathbb{B}}^2 \circ 2\text{-sol}^{\mathbb{R}_+}(proj_G^m(L_2)(\mathbf{y}))$
<i>Proposition 26</i>	$= h_{\mathbb{B}}^2 \circ \nu^{-1}(sol^{\mathbb{R}_+}(\tilde{\nu}(proj_G^m(L_2)(\mathbf{y}))))$
<i>Proposition 29</i>	$= \nu^{-1}(h_{\mathbb{B}} \circ sol^{\mathbb{R}_+}(\tilde{\nu}(proj_G^m(L_2)(\mathbf{y}))))$
<i>Definition of $proj_G^m(L_2)(\mathbf{y})$</i>	$= \nu^{-1}(h_{\mathbb{B}} \circ sol^{\mathbb{R}_+}(\tilde{\nu}(\exists \mathbf{z}. L_2(\mathbf{z}) \wedge \bigwedge_{i=1}^m proj_G(z_i, y_i))))$ où $\mathbf{z} = z_1 \dots z_m$ frais
<i>Théorème de système mixte 93</i>	$= \nu^{-1}(sol^{\mathbb{B}}(\phi(\nu(\mathbf{y}))))$ où $\phi(\nu(\mathbf{y}))$ est une formule conjonctive $h_{\mathbb{B}}$ -exacte et \mathbb{R}_+ -équivalente au système mixte $\tilde{\nu}(\exists \mathbf{z}. L_2(\mathbf{z}) \wedge \bigwedge_{i=1}^m proj_G(z_i, y_i))$

□

A noter que $sol^{\mathbb{B}}(\phi(v(\mathbf{y})))$ peut être calculé par la programmation par contraintes à domaine fini. Cela mène à un algorithme exact pour calculer la Δ_3 -abstraction du système linéaire d'équations $L(\mathbf{y})$.

7.2.2 Les Structures Mixtes

Pour ajouter un traitement des contraintes cinétiques sur Δ_n où $n \in \{3, 6\}$, on considère l'union $\mathbb{B} \cup \Delta_n$ comme une structure relationnelle, réunissant les fonctionnalités des deux structures \mathbb{B} et Δ_n . Pour cela, on définit la signature mixte par:

$$\Sigma_n^{mixte} = \{+^{\mathbb{B}}, *^{\mathbb{B}}, +^{\Delta_n}, *^{\Delta_n}\} \cup \mathbb{B} \cup \Delta_n$$

Ici on réutilise les fonctions binaires de \mathbb{B} et Δ_n comme les symboles de fonction binaire de Σ_n^{mixte} et les valeurs de $\mathbb{B} \cup \Delta_n$ comme les constantes de Σ_n^{mixte} .

Définition 103. Pour chaque $n \in \{3, 6\}$, la structure mixte $\mathbb{B} \cup \Delta_n$ est la Σ_n^{mixte} -structure avec le domaine mixte $\mathbb{B} \cup \Delta_n$ dans lequel tous les symboles de Σ_n^{mixte} sont eux-mêmes, mais maintenant avec le respect du domaine mixte.

7.2.3 Avec contraintes de différences

Chaque paire de booléens dans $\mathbb{B}^2 \setminus (1, 1)$ est une abstraction de différences de Δ_3 , et vice versa. Pour résoudre le problème général en Δ_3 on a besoin d'encapsuler les relations dans la logique du premier-ordre mixte sur $\mathbb{B} \cup \Delta_3$. Pour cela, on considère la fonction partielle $pair_{\Delta_3} \subseteq (\mathbb{B} \cup \Delta_3)^2 \times (\mathbb{B} \cup \Delta_3)$ telle que pour tout $v_1, v_2 \in \mathbb{B} \cup \Delta_3$:

$$pair_{\Delta_3}(v_1, v_2) = \begin{cases} (v_1, v_2) & \text{si } v_1 *^{\mathbb{B}} v_2 = 0 \text{ tel que } (v_1, v_2) \in \Delta_3 \\ \text{indéfini} & \text{sinon} \end{cases}$$

Le domaine de cette fonction partielle est $dom(pair_{\Delta_3}) = \mathbb{B}^2 \setminus \{(1, 1)\} = \Delta^3$. Chaque paire du domaine est associée à elle-même. On peut définir la relation ternaire $pair_{\Delta_3}$ dans la logique du premier-ordre de la structure mixte $\mathbb{B} \cup \Delta_3$ par la fonction $Pair_{\Delta_3} : \mathcal{V}^2 \times \mathcal{V} \rightarrow \mathcal{F}_{\Sigma_3^{mixte}}$ telle que pour toutes variables $y_1, y_2, y \in \mathcal{V}$:

$$\begin{aligned} Pair_{\Delta_3}(y_1, y_2, y) &=_{\text{def}} (y_1 = 0 \wedge y_2 = 0 \wedge y = \approx) \\ &\vee (y_1 = 0 \wedge y_2 = 1 \wedge y = \triangle) \\ &\vee (y_1 = 1 \wedge y_2 = 0 \wedge y = \nabla) \end{aligned}$$

Selon la définition 37, la définition du premier-ordre $Pair_{\Delta_3}$ définie en effet la relation $pair_{\Delta_3}$ dans la structure mixte, qui est $pair_{\Delta_3} = Pair_{\Delta_3}^{\mathbb{B} \cup \Delta_3}$. Dans le but de gérer les deux entrées de $pair_{\Delta_3}$, On considère deux nouveaux générateurs de

variables $\nu_1, \nu_2 : \mathcal{V} \rightarrow \mathcal{V}$. Rappelons que pour chaque sous-ensemble $V \subseteq \mathcal{V}$, structure S , et affectation de variables $\alpha : \nu_1(V) \cup \nu_2(V) \rightarrow \text{dom}(S)$ on définit $\nu^{-1}(\alpha) : V \rightarrow \text{dom}(S)^2$ tel que $\nu^{-1}(\alpha)(y) = (\alpha(\nu_1(y)), \alpha(\nu_2(y)))$ pour tout $y \in V$. Ensuite on définit pour chaque définition du premier-ordre $G : \mathcal{V}^{2m} \rightarrow \mathcal{F}_{\Sigma_3^{\text{mixte}}}$ une définition du premier-ordre $\text{Pair}_{\Delta_3}^m(G) : \mathcal{V}^m \rightarrow \mathcal{F}_{\Sigma_3^{\text{mixte}}}$ telle que $\text{Pair}_{\Delta_3}^m(G)(\mathbf{y})$ décrit une application de Pair_{Δ_3} à tous les composants des solutions de $G(\mathbf{y})$ pour toute séquence de variables $\mathbf{y} = y_1 \dots y_m$:

$$\text{Pair}_{\Delta_3}^m(G)(\mathbf{y}) = \exists v(\mathbf{y}). G(v(\mathbf{y})) \wedge \bigwedge_{i=1}^m \text{Pair}_{\Delta_3}(v_1(y_i), v_2(y_i), y_i)$$

Lemme 104. Soit $m \in \mathbb{N}$, $G : \mathcal{V}^{2m} \rightarrow \mathcal{F}_{\Sigma_3^{\text{mixte}}}$ une définition du premier-ordre, \mathbf{y} une séquence de variables et $\nu(\mathbf{y}) = \nu_1(\mathbf{y})\nu_2(\mathbf{y})$.

$$\nu^{-1}(\text{sol}^{\mathbb{B}}(G(v(\mathbf{y}))) \cap \{\alpha : \{\mathbf{y}\} \rightarrow \Delta_3\}) = \text{sol}^{\mathbb{B} \cup \Delta_3}(\text{Pair}_{\Delta_3}^m(G)(\mathbf{y}))$$

Démonstration. La fonction pair_{Δ_3} associe toutes les paires de booléens de Δ_3 à elles-mêmes, alors qu'elle reste indéfinie pour tout les éléments de $(\Delta_3 \cup \mathbb{B})^2 \setminus \Delta_3$. Ainsi :

$$\nu^{-1}(\text{sol}^{\mathbb{B}}(G(v(\mathbf{y}))) \cap \{\alpha : \{\mathbf{y}\} \rightarrow \Delta_3\}) = \text{pair}_{\Delta_3} \circ \nu^{-1}(\text{sol}^{\mathbb{B} \cup \Delta_3}(G(v(\mathbf{y}))))$$

La composition avec les fonctions partielles pair_{Δ_3} sur la droite est définie dans la Section 2.1. Comme énoncé ici, la composition $\text{pair}_{\Delta_3} \circ \alpha$ est définie seulement pour les fonctions totales α telles que $\text{ran}(\alpha) \subseteq \text{dom}(\text{pair}_{\Delta_3}) = \Delta_3$. Donc une fonction totale $\text{pair}_{\Delta_3} \circ \alpha$ peut appartenir à la partie droite de la composition seulement si α satisfait cette condition.

La Proposition 41 sur les définitions du premier-ordre avec $F = \text{Pair}_{\Delta_3} : \mathcal{V}^2 \times \mathcal{V} \rightarrow \mathcal{F}_{\Sigma_3^{\text{mixte}}}$ montre que :

$$\text{Pair}_{\Delta_3}^{\mathbb{B} \cup \Delta_3} \circ \nu^{-1}(\text{sol}^{\mathbb{B} \cup \Delta_3}(G(v(\mathbf{y})))) = \text{sol}^{\mathbb{B} \cup \Delta_3}(\text{Pair}_{\Delta_3}^m(G)(\mathbf{y}))$$

Ici, les paramètres sont $\ell = 2$, $k = 1$ et $n = 1$. En combinaison avec $\text{pair}_{\Delta_3} = \text{Pair}_{\Delta_3}^{\mathbb{B} \cup \Delta_3}$, ces deux équations cela entraîne ce Lemme. \square

Puisque nous allons travailler sur la structure $\mathbb{B} \cup \Delta_3$, nous avons besoin d'introduire pour chaque formule $\phi \in \mathcal{F}_{\Sigma}$ les formules $\phi^{\mathbb{B}}$ et ϕ^{Δ_3} qui imposent l'utilisation de $*^{\mathbb{B}}$ et $+^{\mathbb{B}}$, et respectivement $*^{\Delta_3}$ et $+^{\Delta_3}$ lors de l'interprétation de ϕ . Il suit que $\text{sol}^{\mathbb{B} \cup \Delta_3}(\phi^{\mathbb{B}}) = \text{sol}^{\mathbb{B}}(\phi)$.

Théorème 105 ((Résoudre l'Abstraction des Différences avec Contraintes pour h_{Δ_3})). *Pour chaque formule linéaire $L(\mathbf{y}) \in \mathcal{F}_{\Sigma}$ avec l'ensemble de variables libres $\{\mathbf{y}\}$, et chaque contrainte $C(\mathbf{y}') \in \mathcal{F}_{\Sigma \cup \Delta_3}$ avec l'ensemble de variables libres $\{\mathbf{y}'\}$, et $V \subseteq \{\mathbf{y}\} \cup \{\mathbf{y}'\} = V'$, on peut calculer en temps au plus exponentiel une formule sur la signature mixte $M \in \mathcal{F}_{\Sigma_3^{\text{mixte}}}$ telle que :*

$$\text{sol}^{\mathbb{B} \cup \Delta_3}(M) = \{\beta|_V \mid \beta \in h_{\Delta_3} \circ \text{sol}_{V'}^{\mathbb{R}_+^2}(L(\mathbf{y})) \cap \text{sol}_{V'}^{\Delta_3}(C(\mathbf{y}'))\}$$

Démonstration. Sans perte de généralités on peut assumer que $\mathbf{y} = \mathbf{y}'$. Si ce n'est pas le cas, cela peut être obtenu naïvement en ajoutant pour tout $z \in V'$ une équation redondante $v(z) \doteq v(z)$ en conjonction avec $L(\mathbf{y})$ et $C(\mathbf{y}')$. Une fois que cela est fait on a $V' = \{\mathbf{y}\} = \{\mathbf{y}'\}$.

Par le Théorème 102 on peut calculer en temps au plus exponentiel une formule $\phi(v(\mathbf{y})) \in \mathcal{F}_{\Sigma}$ telle que :

$$h_{\Delta_3} \circ \text{diff}(\text{sol}^{\mathbb{R}_+}(L)) = v^{-1}(\text{sol}^{\mathbb{B}}(\phi(v(\mathbf{y}))))$$

La formule $M \in \mathcal{F}_{\Sigma_3^{\text{mixte}}}$ peut ensuite être définie comme suit :

$$M =_{\text{def}} \exists V' \setminus V. \text{Pair}_{\Delta_3}^m(\phi(v(\mathbf{y}))^{\mathbb{B}}) \wedge C(\mathbf{y}')^{\Delta_3}$$

A noter que M peut être calculé en temps linéaire à partir de $\phi(v(\mathbf{y}))$ et $C(\mathbf{y}')$. On a besoin de montrer que la formule M satisfait l'équation du Théorème. Tout d'abord notons que :

$$\text{sol}^{\mathbb{B}}(\phi(\mathbf{y})) = \text{sol}^{\mathbb{B} \cup \Delta_3}(\phi(\mathbf{y})^{\mathbb{B}})$$

Le Lemme 104 sur la définition du premier-ordre de pair_{Δ_3} montre que :

$$v^{-1}(\text{sol}^{\mathbb{B} \cup \Delta_3}(\phi(\mathbf{y})^{\mathbb{B}})) \cap \{\alpha : \{\mathbf{y}\} \rightarrow \Delta_3\} = \text{sol}^{\mathbb{B} \cup \Delta_3}(\text{Pair}_{\Delta_3}^m(\phi(\mathbf{y})^{\mathbb{B}}))$$

Ainsi :

$$v^{-1}(\text{sol}^{\mathbb{B} \cup \Delta_3}(\phi(\mathbf{y})^{\mathbb{B}})) \cap \text{sol}^{\mathbb{B} \cup \Delta_3}(C(\mathbf{y})^{\Delta_3}) = \text{sol}^{\mathbb{B} \cup \Delta_3}(\text{Pair}_{\Delta_3}^m(\phi(\mathbf{y})^{\mathbb{B}}) \cap \text{sol}^{\mathbb{B} \cup \Delta_3}(C(\mathbf{y})^{\Delta_3}))$$

Avec les équations on peut à présent conclure que :

$$\begin{aligned} & \{\beta|_V \mid \beta \in v^{-1}(\text{sol}^{\mathbb{B}}(\phi(\mathbf{y}))) \cap \text{sol}^{\Delta_3}(C(\mathbf{y}))\} \\ &= \{\beta|_V \mid \beta \in v^{-1}(\text{sol}^{\mathbb{B} \cup \Delta_3}(\phi(\mathbf{y})^{\mathbb{B}})) \cap \text{sol}^{\mathbb{B} \cup \Delta_3}(C(\mathbf{y})^{\Delta_3})\} \\ &= \{\beta|_V \mid \beta \in \text{sol}^{\mathbb{B} \cup \Delta_3}(\text{Pair}_{\Delta_3}^m(\phi(\mathbf{y})^{\mathbb{B}}) \cap \text{sol}^{\mathbb{B} \cup \Delta_3}(C(\mathbf{y})^{\Delta_3}))\} \\ &= \{\beta' \mid \beta' \in \text{sol}^{\mathbb{B} \cup \Delta_3}(\exists V' \setminus V. \text{Pair}_{\Delta_3}^m(\phi(\mathbf{y})^{\mathbb{B}}) \wedge C(\mathbf{y})^{\Delta_3})\} \\ &= \text{sol}^{\mathbb{B} \cup \Delta_3}(M) \end{aligned}$$

□

L'ensemble $\text{sol}^{\mathbb{B}\cup\Delta_3}(M)$ peut être calculé par la programmation par contraintes à domaine fini, puisque $\mathbb{B}\cup\Delta_3$ est une structure finie. C'est pourquoi le Théorème 105 retourne un algorithme pour résoudre le problème générale de l'abstraction des différences avec contraintes pour le cas Δ_3 .

7.3 Exact Δ_6 -abstraction

Le cas de Δ_6 suit la même approche que pour Δ_3 , mais il est bien plus évolué dans l'usage des définitions du premier-ordre. Il utilise d'ailleurs la caractérisation booléenne de Δ_3 dans sa propre caractérisation, il est donc intéressant de traiter le cas de Δ_3 avant celui-ci. Après analyse de chacune des étapes de transformation du système, nous pourrions voir le théorème 112 qui nous donne l'algorithme exact permettant de résoudre l'abstraction des différences avec contraintes pour h_{Δ_6} .

7.3.1 Systèmes linéaires

Considérons l'abstraction h_{Δ_6} comme un élément de la Σ -algèbre de fonction totale du type $\mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2$, où $\Sigma_{\text{pos-arith}} = F_{\text{pos-arith}}^{(2)} \cup F_{\text{pos-arith}}^{(1)} \cup C_{\text{pos-arith}}$. Pour chacune des deux fonctions $f, f' : \mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2$, l'addition est définie par $f +_{\mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2} f'(p) = f(p) +_{\mathbb{R}_+^2} f'(p)$ pour chaque $p \in \mathbb{R}_+^2$, et de façon similaire pour la multiplication, elle est définie par $f *_{\mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2} f'(p) = f(p) *_{\mathbb{R}_+^2} f'(p)$. Le Lemme suivant montre que h_{Δ_6} est la somme de h_{Δ_3} et $h_{\mathbb{B}}^2$ dans la Σ -algèbre.

Lemme 106. $h_{\Delta_6} = h_{\mathbb{B}}^2 + h_{\Delta_3}$ où $+ = +_{\mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2}$

Démonstration. Soit $p = (r, r') \in \mathbb{R}_+^2$. Nous distinguons les cas pour toutes les valeurs possibles de $h_{\Delta_6}(p) \in \Delta_6$.

Cas $h_{\Delta_6}(p) = \uparrow$. Alors $0 < r < r'$ tel que $h_{\mathbb{B}}^2(p) = (1, 1)$ et $h_{\Delta_3}(p) = \Delta = (0, 1)$.

D'où $(h_{\mathbb{B}}^2 + h_{\Delta_3})(p) = (1, 2) = \uparrow = h_{\Delta_6}(p)$.

Cas $h_{\Delta_6}(p) = \uparrow\uparrow$. Alors $0 = r < r'$ tel que $h_{\mathbb{B}}^2(p) = (0, 1)$ et $h_{\Delta_3}(p) = \Delta = (0, 1)$.

D'où $(h_{\mathbb{B}}^2 + h_{\Delta_3})(p) = (0, 2) = \uparrow\uparrow = h_{\Delta_6}(p)$.

Cas $h_{\Delta_6}(p) = \downarrow$. Alors $0 < r' < r$ tel que $h_{\mathbb{B}}^2(p) = (1, 1)$ et $h_{\Delta_3}(p) = \nabla = (1, 0)$.

D'où $(h_{\mathbb{B}}^2 + h_{\Delta_3})(p) = (2, 1) = \downarrow = h_{\Delta_6}(p)$.

Cas $h_{\Delta_6}(p) = \downarrow\downarrow$. Alors $0 = r' < r$ tel que $h_{\mathbb{B}}^2(p) = (1, 0)$ et $h_{\Delta_3}(p) = \nabla = (1, 0)$.

D'où $(h_{\mathbb{B}}^2 + h_{\Delta_3})(p) = (2, 0) = \downarrow\downarrow = h_{\Delta_6}(p)$.

Cas $h_{\Delta_6}(p) = \sim$. Alors $0 < r = r'$ tel que $h_{\mathbb{B}}^2(p) = (1, 1)$ et $h_{\Delta_3}(p) = \approx = (0, 0)$.

D'où $(h_{\mathbb{B}}^2 + h_{\Delta_3})(p) = (1, 1) = \sim = h_{\Delta_6}(p)$.

Cas $h_{\Delta_6}(p) = \approx$. Alors $0 = r = r'$ tel que $h_{\mathbb{B}}^2(p) = (0, 0)$ et $h_{\Delta_3}(p) = \approx = (0, 0)$.

D'où $(h_{\mathbb{B}}^2 + h_{\Delta_3})(p) = (0, 0) = \approx = h_{\Delta_6}(p)$.

□

Soit $id-proj_G^{\mathbb{R}_+^2} : \mathbb{R}_+^2 \rightarrow (\mathbb{R}_+^2)^2$ tel que pour chaque $p \in \mathbb{R}_+^2$ $id-proj_G^{\mathbb{R}_+^2}(p) = (p, proj_G^{\mathbb{R}_+^2}(p))$. De plus, nous définissons pour chaque fonctions $g : A \rightarrow B \times C$ et $f : B \times C \rightarrow D$ la pseudo composition $f \bullet g : A \rightarrow D$ telle que pour tout $a \in A$: $(f \bullet g)(a) = f(\pi_1(g(a)), \pi_2(g(a)))$. La Σ -abstraction $h_{\mathbb{B}}^2 : \mathbb{R}_+^2 \rightarrow \mathbb{B}^2$ nous permet de définir $(h_{\mathbb{B}}^2)^2 : (\mathbb{R}_+^2)^2 \rightarrow (\mathbb{B}^2)^2$

Lemme 107 (Décomposition). $h_{\Delta_6} = +^{\mathbb{R}_+^2} \bullet (h_{\mathbb{B}}^2)^2 \circ id-proj_G^{\mathbb{R}_+^2}$.

Démonstration. Pour chaque $p \in \mathbb{R}_+^2$, nous avons:

$$\begin{aligned} h_{\Delta_6}(p) &= h_{\mathbb{B}}^2(p) +^{\mathbb{R}_+^2} h_{\mathbb{B}}^2(proj_G^{\mathbb{R}_+^2}(p)) \\ &= +^{\mathbb{R}_+^2}(h_{\mathbb{B}}^2(p), h_{\mathbb{B}}^2(proj_G^{\mathbb{R}_+^2}(p))) \\ &= +^{\mathbb{R}_+^2}((h_{\mathbb{B}}^2)^2(p, proj_G^{\mathbb{R}_+^2}(p))) \\ &= +^{\mathbb{R}_+^2}((h_{\mathbb{B}}^2)^2(id(p), proj_G^{\mathbb{R}_+^2}(p))) \\ &= +^{\mathbb{R}_+^2} \bullet (h_{\mathbb{B}}^2)^2 \circ id-proj_G^{\mathbb{R}_+^2}(p) \end{aligned}$$

□

On peut définir la relation ternaire $id-proj_G^{\mathbb{R}_+^2} : \mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2 \times \mathbb{R}_+^2$ dans la logique des paires du premier-ordre par $id-proj_G : \mathcal{V} \times \mathcal{V}^2 \rightarrow \mathcal{F}_{\Sigma}^2$ telle que pour tout $x, y_1, y_2 \in \mathcal{V}$:

$$id-proj_G(x, y_1, y_2) =_{\text{def}} \langle x = y_1 \rangle^2 \wedge proj_G(x, y_2)$$

Nous définissons ensuite les applications de fonction définie par $id-proj_G$ en logique du premier-ordre. Pour pouvoir gérer les deux arguments de sortie, nous utilisons deux générateurs de variables fraîches $\mu_1, \mu_2 : \mathcal{V} \rightarrow \mathcal{V}$. Pour chaque définition du premier-ordre $G : \mathcal{V}^m \rightarrow \mathcal{F}_{\Sigma}^2$ on définit une définition du premier-ordre $id-proj_{G_{\mu}}^m(G) : \mathcal{V}^{2m} \rightarrow \mathcal{F}_{\Sigma}^2$, telle que pour chaque séquence de variables $\mathbf{y} \in \mathcal{V}^m$ et avec $\mu(\mathbf{y}) = \mu_1(\mathbf{y})\mu_2(\mathbf{y})$:

$$id-proj_{G_{\mu}}^m(G)(\mu(\mathbf{y})) =_{\text{def}} \exists \mathbf{y}. G(\mathbf{y}) \wedge \bigwedge_{i=1}^m id-proj_G(y_i, \mu_1(y_i), \mu_2(y_i))$$

Lemme 108.

$$id-proj_G^{\mathbb{R}_+^2} \circ 2-sol^{\mathbb{R}_+}(G(\mathbf{y})) = \left\{ \begin{array}{l} [\mathbf{y}/(\alpha(\mu_1(\mathbf{y})), \alpha(\mu_2(\mathbf{y}))) \mid \\ \alpha \in 2-sol^{\mathbb{R}_+}(id-proj_{G_{\mu}}^m(G)(\mu(\mathbf{y}))) \end{array} \right\}$$

Démonstration. Ce Lemme est une conséquence de la propriété de définition en logique des tuples du premier-ordre que nous pouvons voir dans la Proposition 42 de la Section 3.5. Ici nous choisissons comme paramètres la définition du premier-ordre $F = id-proj_{G_\mu} : \mathcal{V} \times V^2 \rightarrow \mathcal{F}_\Sigma^2$, et $\ell = 1, k = 2, n = 2$. □

Théorème 109. *Pour chaque formule linéaire $L(\mathbf{y})$ avec ses variables libres distinctes dans \mathbf{y} , nous pouvons calculer en temps au plus exponentiel une formule conjonctive positive avec quantifieurs existentiels $\phi(v(\mu(\mathbf{y}))) \in \mathcal{F}_\Sigma$. Celle-ci, avec ses variables libres dans $v(\mu(\mathbf{y}))$, est telle que $h_{\Delta_6} \circ diff(sol^{\mathbb{R}^+}(L(\mathbf{y})))$ peut être calculer en temps polynomial à partir de $sol^{\mathbb{B}}(\phi(v(\mu(\mathbf{y}))))$. De plus :*

$$\begin{aligned} & h_{\Delta_6} \circ diff(sol^{\mathbb{R}^+}(L(\mathbf{y}))) \\ &= pair-sum_{\Delta_6} \circ \left\{ \begin{array}{l} [y/(\beta^2(v(\mu_1(\mathbf{y}))) + \mathbb{R}_+^2 \beta^2(v(\mu_2(\mathbf{y})))) \mid y \in \{\mathbf{y}\}] \\ \mid \beta \in sol^{\mathbb{B}}(\phi(v(\mu(\mathbf{y})))) \end{array} \right\} \end{aligned}$$

Démonstration. Le fait que $h_{\Delta_6} \circ diff(sol^{\mathbb{R}^+}(L(\mathbf{y})))$ peut être calculé en temps polynomial à partir de $sol^{\mathbb{B}}(\psi')$ vient de l'égalité énoncé dans le Théorème, donc cette égalité est ce qu'il faut prouver ici. Soit $L : \mathcal{V}^m \rightarrow \mathcal{F}_\Sigma$ la définition du premier-ordre qui une fois appliquée à \mathbf{y} retourne la formule linéaire $L(\mathbf{y})$.

$$\begin{array}{l|l} \text{Proposition 31} & h_{\Delta_6} \circ diff(sol^{\mathbb{R}^+}(L(\mathbf{y}))) \\ \text{Proposition 22} & = h_{\Delta_6} \circ sol^{\mathbb{R}_+^2}(L(\mathbf{y})) \\ \text{Dec. Lemma 107} & = h_{\Delta_6} \circ 2-sol^{\mathbb{R}^+}(L_2(\mathbf{y})) \text{ with } L_2(\mathbf{y}) = \langle L(\mathbf{y}) \rangle^2 \\ & = +_{\mathbb{R}_+^2} \bullet (h_{\mathbb{B}}^2)^2 \circ id-proj_G^{\mathbb{R}_+^2} \circ 2-sol^{\mathbb{R}^+}(L_2(\mathbf{y})) \\ \text{FO Lemma 108} & = +_{\mathbb{R}_+^2} \bullet (h_{\mathbb{B}}^2)^2 \circ \left\{ \begin{array}{l} [\mathbf{y}/(\alpha(\mu_1(\mathbf{y})), \alpha(\mu_2(\mathbf{y})))] \\ \mid \alpha \in 2-sol^{\mathbb{R}^+}(id-proj_{G_\mu}^m(L_2)(\mu(\mathbf{y}))) \end{array} \right\} \\ & = +_{\mathbb{R}_+^2} \circ \left\{ \begin{array}{l} [\mathbf{y}/(\beta(\mu_1(\mathbf{y})), \beta(\mu_2(\mathbf{y})))] \\ \mid \beta \in h_{\mathbb{B}}^2 \circ 2-sol^{\mathbb{R}^+}(id-proj_{G_\mu}^m(L_2)(\mu(\mathbf{y}))) \end{array} \right\} \\ & = \left\{ \begin{array}{l} [\mathbf{y}/(\beta(\mu_1(\mathbf{y})) + \mathbb{R}_+^2 \beta(\mu_2(\mathbf{y}))) \\ \mid \beta \in h_{\mathbb{B}}^2 \circ 2-sol^{\mathbb{R}^+}(id-proj_{G_\mu}^m(L_2)(\mu(\mathbf{y}))) \end{array} \right\} \end{array}$$

On peut calculer l'abstraction $h_{\mathbb{B}}^2$ de l'ensemble de solutions ci-dessus de façon similaire au cas de Δ_3 .

$$\begin{array}{l|l} \text{Proposition 26} & h_{\mathbb{B}}^2 \circ 2-sol^{\mathbb{R}^+}(id-proj_{G_\mu}^m(L_2)(\mu(\mathbf{y}))) \\ \text{Proposition 29} & = h_{\mathbb{B}}^2 \circ v^{-1}(sol^{\mathbb{R}^+}(\tilde{v}(id-proj_{G_\mu}^m(L_2)(\mu(\mathbf{y})))))) \\ \text{Def. of } id-proj_{G_\mu}^m(L_2) & = v^{-1}(h_{\mathbb{B}} \circ sol^{\mathbb{R}^+}(\tilde{v}(id-proj_{G_\mu}^m(L_2)(\mu(\mathbf{y})))))) \\ & = v^{-1}(h_{\mathbb{B}} \circ sol^{\mathbb{R}^+}(\tilde{v}(\exists \mathbf{y}. L_2(\mathbf{y})))) \\ & \quad \wedge \bigwedge_{i=1}^m id-proj_G(y_i, \mu_1(y_i), \mu_2(y_i)) \\ \text{Théorème 93} & = v^{-1}(sol^{\mathbb{B}}(\phi'(v(\mu(\mathbf{y})))))) \end{array}$$

où $\phi'(v(\mu(\mathbf{y})))$ est une formule conjonctive équivalente au système $h_{\mathbb{B}}$ -mixte

$$\tilde{v}(\exists \mathbf{y}. L_2(\mathbf{y}) \wedge \bigwedge_{i=1}^m id-proj_G(y_i, \mu_1(y_i), \mu_2(y_i)))$$

Finalement on obtient :

$$h_{\Delta_6} \circ diff(sol^{\mathbb{R}^+}(L(\mathbf{y}))) \\ \{[y/(\beta^2(v(\mu_1(y)))) +^{\mathbb{R}^2} \beta^2(v(\mu_2(y)))] \mid y \in \{\mathbf{y}\} \mid \beta \in sol^{\mathbb{B}}(\phi(v(\mu(\mathbf{y}))))\}$$

□

7.3.2 Avec contraintes de différences

Considérons maintenant la fonction partielle $pair-sum_{\Delta_6} \subseteq (\mathbb{B} \cup \Delta_6)^4 \times (\mathbb{B} \cup \Delta_6)$ qui associe deux paires de booléens à l'abstraction de différences en Δ_6 sous le sens que pour tout $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{B}^2$:

$$pair-sum_{\Delta_6}(\mathbf{b}_1, \mathbf{b}_2) = \begin{cases} \mathbf{b}_1 +^{\mathbb{R}^2} \mathbf{b}_2 & \text{si } \mathbf{b}_1, \mathbf{b}_2 \in \mathbb{B}^2 \text{ et } \mathbf{b}_1 +^{\mathbb{R}^2} \mathbf{b}_2 \in \Delta_6 \\ \text{indéfini} & \text{sinon} \end{cases}$$

En utilisant cette fonction partielle et le Théorème 109 on peut réécrire $h_{\Delta_6} \circ diff(sol^{\mathbb{R}^+}(L(\mathbf{y})))$ pour chaque système d'équation linéaire $L(\mathbf{y})$ comme suit, avec μ, v, ϕ qui sont choisis comme décrit dans le Théorème :

$$\{[y/pair-sum_{\Delta_6}(\beta^2(v(\mu_1(y))), \beta^2(v(\mu_2(y)))) \mid y \in \{\mathbf{y}\} \mid \beta \in sol^{\mathbb{B}}(\phi(v(\mu(\mathbf{y}))))\}$$

On définit ensuite les relations des paires $pair-sum_{\Delta_6}$ dans la logique des paires du premier-ordre et mixte $\mathbb{B} \cup \Delta_6$ par la fonction $Pair-Sum_{\Delta_6} : \mathcal{V}^4 \times \mathcal{V} \in \mathcal{F}_{\Sigma_6^{mixte}}$ telle que pour toutes variables $y_1, y_2, y_3, y_4, y \in \mathcal{V}$:

$$Pair-Sum_{\Delta_6}(y_1, y_2, y_3, y_4, y) =_{\text{def}} \begin{aligned} & (y_1 \doteq 0 \wedge y_2 \doteq 0 \wedge y_3 \doteq 0 \wedge y_4 \doteq 0 \wedge y \doteq \approx) \\ & \vee (y_1 \doteq 1 \wedge y_2 \doteq 1 \wedge y_3 \doteq 0 \wedge y_4 \doteq 0 \wedge y \doteq \sim) \\ & \vee (y_1 \doteq 1 \wedge y_2 \doteq 0 \wedge y_3 \doteq 0 \wedge y_4 \doteq 1 \wedge y \doteq \uparrow) \\ & \vee (y_1 \doteq 1 \wedge y_2 \doteq 1 \wedge y_3 \doteq 0 \wedge y_4 \doteq 1 \wedge y \doteq \uparrow) \\ & \vee (y_1 \doteq 0 \wedge y_2 \doteq 1 \wedge y_3 \doteq 1 \wedge y_4 \doteq 0 \wedge y \doteq \downarrow) \\ & \vee (y_1 \doteq 1 \wedge y_2 \doteq 1 \wedge y_3 \doteq 1 \wedge y_4 \doteq 0 \wedge y \doteq \downarrow) \end{aligned}$$

Selon la Définition 37, la définition du premier-ordre $Pair-Sum_{\Delta_6}$ définit bien la fonction partielle $pair-sum_{\Delta_6}$ dans la structure mixte, qui est

$$pair-sum_{\Delta_6} = Pair-Sum_{\Delta_6}^{\mathbb{B} \cup \Delta_6}$$

On continue avec quatre générateurs de variables fraîches $\mu_1, \mu_2, \nu_1, \nu_2 : \mathcal{V} \rightarrow \mathcal{V}$. Pour chaque définition du premier-ordre $G : \mathcal{V}^{4m} \rightarrow \mathcal{F}_{\Sigma_6^{mixte}}$ on définit une définition du premier-ordre $Pair-Sum_{\Delta_6}^m(G) : \mathcal{V}^m \rightarrow \mathcal{F}_{\Sigma_6^{mixte}}$. Pour toute séquence de variables $\mathbf{y} = y_1 \dots y_m$, la formule $Pair-Sum_{\Delta_6}^m(G)(\mathbf{y})$ décrit l'application de la fonction partielle, défini par $Pair-Sum_{\Delta_6}$, sur les paires de paires définies par $G(v(\mu(\mathbf{y})))$, c'est-à-dire :

$$Pair-Sum_{\Delta_6}^m(G)(\mathbf{y}) = \exists v(\mu(\mathbf{y})). G(v(\mu(\mathbf{y}))) \wedge \bigwedge_{i=1}^m Pair-Sum_{\Delta_6}(v(\mu_1(y_i)), v(\mu_2(y_i)), y_i)$$

Lemme 110.

$$\begin{aligned} & sol^{\mathbb{B} \cup \Delta_6}(Pair-Sum_{\Delta_6}^m(G)(\mathbf{y})) \\ &= \left\{ \begin{array}{l} [y/pair-sum_{\Delta_6}(\beta^2(v(\mu_1(y))), \beta^2(v(\mu_2(y)))) \mid y \in \{\mathbf{y}\}] \\ \mid \beta \in sol^{\mathbb{B} \cup \Delta_6}(G(v(\mu(\mathbf{y})))) \end{array} \right\} \end{aligned}$$

Démonstration. On utilise le fait que $pair-sum_{\Delta_6} = Pair-Sum_{\Delta_6}^{\mathbb{B} \cup \Delta_6}$ et la propriété générale de la définition du premier-ordre de la Proposition 41 avec $F = Pair-Sum_{\Delta_6}$, $\ell = 4$, $k = 1$ et $n = 1$. Les quatre générateurs de variables fraîches que nous utiliserons sont $\{v_i \circ \mu_j \mid i, j \in \{1, 2\}\}$.

$$\begin{aligned} & sol^{\mathbb{B} \cup \Delta_6}(Pair-Sum_{\Delta_6}^m(G)(\mathbf{y})) \\ &= pair-sum_{\Delta_6} \circ \left\{ \begin{array}{l} \alpha : \{\mathbf{y}\} \rightarrow \mathbb{B} \cup \Delta_6 \mid \exists \beta \in sol^{\mathbb{B} \cup \Delta_6}(G(v(\mu(\mathbf{y})))) \\ \forall y \in \{\mathbf{y}\}. \alpha(y) = \beta^2(v(\mu_1(y))), \beta^2(v(\mu_2(y))) \end{array} \right\} \end{aligned}$$

□

Proposition 111. Pour chaque formule $\phi(v(\mu(\mathbf{y}))) \in \mathcal{F}_{\Sigma}$ et contrainte $C(\mathbf{y}) \in \mathcal{F}_{\Sigma \cup \Delta_6}$ avec les mêmes variables libres $\{\mathbf{y}\}$, et chaque sous-ensemble $V \subseteq \{\mathbf{y}\}$ on peut calculer en temps linéaire une formule $M \in \mathcal{F}_{\Sigma_6^{mixte}}$ avec $fv(M) = V$ telle que :

$$\begin{aligned} & sol^{\mathbb{B} \cup \Delta_6}(M) = \\ & \left\{ \begin{array}{l} \alpha_{|V} \mid \alpha = [y/pair-sum_{\Delta_6}(\beta^2(v(\mu_1(y))), \beta^2(v(\mu_2(y)))) \mid y \in \{\mathbf{y}\}], \\ \beta \in sol^{\mathbb{B}}(\phi(v(\mu(\mathbf{y})))) \end{array} \right\} \\ & \cap \{\alpha_{|V} \mid \alpha \in sol^{\Delta_6}(C(\mathbf{y}))\} \end{aligned}$$

Démonstration. On peut choisir $M = \exists \{\mathbf{y}\} \setminus V. Pair-Sum_{\Delta_6}^m(L(\mathbf{y})) \wedge C(\mathbf{y})^{\Delta_6}$.

$$\begin{aligned} & \left\{ \begin{array}{l} \alpha_{|V} \mid \alpha = [y/pair-sum_{\Delta_6}(\beta^2(v(\mu_1(y))), \beta^2(v(\mu_2(y)))) \mid y \in \{\mathbf{y}\}], \\ \beta \in sol^{\mathbb{B}}(\phi(v(\mu(\mathbf{y})))) \end{array} \right\} \\ & \cap \{\alpha_{|V} \mid \alpha \in sol^{\Delta_6}(C(\mathbf{y}))\} \\ &= sol^{\mathbb{B} \cup \Delta_6}(Pair-Sum_{\Delta_6}^m(\phi(\mathbf{y}))) \cap \{\alpha_{|V} \mid \alpha \in sol^{\mathbb{B} \cup \Delta_6}(C(\mathbf{y})^{\Delta_6})\} \quad \text{By Lemma 110} \\ &= sol^{\mathbb{B} \cup \Delta_6}(M) \end{aligned}$$

□

Théorème 112 ((Résoudre l'Abstraction de Différences avec Contraintes pour h_{Δ_6})). *Pour chaque formule linéaire $L(\mathbf{y}) \in \mathcal{F}_{\Sigma}$ avec l'ensemble de variables libres $\{\mathbf{y}\}$, et chaque contrainte $C(\mathbf{y}') \in \mathcal{F}_{\Sigma \cup \Delta_6}$ avec l'ensemble de variables libres $\{\mathbf{y}'\}$, et $V \subseteq \{\mathbf{y}\} \cup \{\mathbf{y}'\} = V'$ on peut calculer en temps au plus exponentiel une formule sur la signature mixte $M \in \mathcal{F}_{\Sigma_6^{\text{mixte}}}$ telle que :*

$$\text{sol}^{\mathbb{B} \cup \Delta_6}(M) = \{\beta|_V \mid \beta \in h_{\Delta_6} \circ \text{sol}_{V'}^{\mathbb{R}_+^2}(L(\mathbf{y})) \cap \text{sol}_{V'}^{\Delta_6}(C(\mathbf{y}'))\}$$

Démonstration. Comme pour l'abstraction de différences avec contraintes en Δ_3 , sans perte de généralité on peut assumer que $\mathbf{y} = \mathbf{y}'$. Une fois cela fait on a $V' = \{\mathbf{y}\} = \{\mathbf{y}'\}$. Par le Théorème 109, on peut calculer en temps au plus exponentiel une formule $\phi(v(\mu(\mathbf{y}))) \in \mathcal{F}_{\Sigma}$ telle que:

$$\begin{aligned} & h_{\Delta_6} \circ \text{diff}(\text{sol}^{\mathbb{R}_+}(L(\mathbf{y}))) \\ &= \{[y/(\beta^2(v(\mu_1(y))) + \beta^2(v(\mu_2(y)))) \mid y \in \{\mathbf{y}\}] \mid \beta \in \text{sol}^{\mathbb{B}}(\phi(v(\mu(\mathbf{y}))))\} \end{aligned}$$

Avec la définition de $\text{pair-sum}_{\Delta_6}$ on obtient :

$$\begin{aligned} & h_{\Delta_6} \circ \text{diff}(\text{sol}^{\mathbb{R}_+}(L(\mathbf{y}))) \\ &= \left\{ \alpha|_V \mid \begin{array}{l} \alpha = [y/\text{pair-sum}_{\Delta_6}(\beta^2(v(\mu_1(y))), \beta^2(v(\mu_2(y)))) \mid y \in \{\mathbf{y}\}], \\ \beta \in \text{sol}^{\mathbb{B}}(\phi(v(\mu(\mathbf{y})))) \end{array} \right\} \end{aligned}$$

Finalement, à partir de la Proposition 111, on peut calculer en temps linéaire une formule $M \in \mathcal{F}_{\Sigma_6^{\text{mixte}}}$ telle que :

$$\text{sol}^{\mathbb{B} \cup \Delta_6}(M) = \left\{ \alpha|_V \mid \begin{array}{l} \alpha = [y/\text{pair-sum}_{\Delta_6}(\beta^2(v(\mu_1(y))), \beta^2(v(\mu_2(y)))) \mid y \in \{\mathbf{y}\}], \\ \beta \in \text{sol}^{\mathbb{B}}(\phi(v(\mu(\mathbf{y})))) \end{array} \right\} \cap \{\alpha|_V \mid \alpha \in \text{sol}^{\Delta_6}(C(\mathbf{y}))\}$$

□

L'ensemble $\text{sol}^{\mathbb{B} \cup \Delta_6}(M)$ peut être calculé par la programmation par contraintes à domaine fini, puisque $\mathbb{B} \cup \Delta_6$ est une structure finie. En combinant le Théorème 109 et la Proposition 111 on obtient un algorithme pour résoudre le problème général pour la cas de Δ_6 .

Heuristique et comparaison

Dans ce chapitre, nous voyons une heuristique pour le calcul de l'interprétation abstraite de réseau de réactions qui repose une fois de plus sur l'ajout d'équations à notre système linéaire avant son abstraction. Son implémentation permet une comparaison avec l'algorithme exact, et nous verrons qu'elle n'est pas exacte, mais qu'elle reste tout de même une bonne heuristique si l'on compare son temps de calcul avec celui de l'algorithme exact.

Sommaire

8.1	Heuristique des conséquences à support minimal	151
8.2	Résultats expérimentaux	153

8.1 Heuristique des conséquences à support minimal

Je propose une nouvelle heuristique pour approximer le problème d'abstraction de différence avec contraintes. Nous verrons plus tard que cette heuristique est exacte si l'on regarde les applications, et requiert peu de temps de calcul en comparaison avec l'algorithme exact.

Soit $h : \mathbb{R}_+^2 \rightarrow \Delta$ une Σ -abstraction dans une Σ -structure finie Δ . Voyons l'idée générale des heuristiques existantes (voir Niehren et al. (2016)) pour approximer le problème d'abstraction des différences avec contraintes 112. Soit un système linéaire d'équations $L \in \mathcal{F}_\Sigma$, une contrainte $C \in \mathcal{F}_{\Sigma[\text{dom}(\Delta)]}$ et un sous-ensemble

de variables $V \subseteq V' = \mathcal{V}(L) \cup \mathcal{V}(C)$. Premièrement, il faut calculer un système linéaire d'équations $L' \in \mathcal{F}_\Sigma$ qui est une conséquence logique de L sur \mathbb{R}_+ avec $\mathcal{V}(L') \subseteq V'$. Ensuite vient le calcul de l'ensemble de solutions abstraites :

$$\text{sol}^\Delta(\exists V. (L \wedge L' \wedge C))$$

Ceci est fait par l'intermédiaire de la programmation par contrainte à domaine fini. Avec le Théorème de John dans le corollaire 35, cet ensemble est une surapproximation de la cible du problème $\{\beta|_V \mid \beta \in \text{sol}_{V'}^{\mathbb{R}_+}(L)^\Delta \cap \text{sol}_{V'}^\Delta(C)\}$.

Le choix de quelle \mathbb{R}_+ -conséquence L' de L à ajouter à L est compliqué. En général, L' est une conjonction finie d'équations linéaires qui sont des \mathbb{R}_+ -conséquences de L . Celles-ci sont toutes des combinaisons linéaires des équations de L . Cependant, il existe une infinité de combinaisons linéaires, et pour obtenir L' il faut choisir un sous-ensemble fini.

Une équation $E \in \mathcal{F}_\Sigma$ est dite *linéaire* si E est de la forme $n_1x_1 + \dots + n_kx_k = m_1y_1 + \dots + m_ly_l$, avec des variables x_i et y_j distinctes, $k, l \geq 0$ des nombres naturels et $n_i, m_j > 0$ des nombres naturels non nuls. E est dit *non trivial* si n'y a pas $k = 0$ et $l = 0$. Le support de E est l'ensemble de ses variables libres $\mathcal{V}(E)$. E est *normalisé* s'il n'existe pas de nombres naturels $p, n'_1, \dots, n'_l, m'_1, \dots, m'_n$ tels que $n_i = p *^{\mathbb{N}} n'_i$ et $m_j = p *^{\mathbb{N}} m'_j$ pour tout i, j .

Définition 113 ($(\mathbb{R}_+$ -conséquences linéaire à support minimal)). *Une équation linéaire $E \in \mathcal{F}_\Sigma$ est une \mathbb{R}_+ -conséquence linéaire à support minimal du système d'équations linéaires $L \in \mathcal{F}_\Sigma$ si elle satisfait les trois conditions suivantes :*

- *E est une \mathbb{R}_+ -conséquence non triviale de L ,*
- *il n'existe pas d'autre \mathbb{R}_+ -conséquence non triviale de L avec un support plus petit que E , et*
- *E est normalisé.*

Naturellement, deux \mathbb{R}_+ -conséquences différentes de L ne peuvent pas avoir le même support. De ce fait, l'ensemble de \mathbb{R}_+ -conséquences de L est fini et à une cardinalité maximale de $2^{|\mathcal{V}(E)|}$. Étant donné un système d'équations linéaires $L \in \mathcal{F}_\Sigma$, la conjonction de toutes les \mathbb{R}_+ -conséquences linéaires à support minimal s'écrit :

$$L_{\text{msc}} \in \mathcal{F}_\Sigma$$

Nous pouvons maintenant voir comment calculer L_{msc} à partir de L . Tout d'abord, nous transformons en temps linéaire L en une matrice d'entiers A telle que L est équivalente à $A\mathbf{x} = 0$, où $\mathcal{V}(L) = \{\mathbf{x}\}$. Les \mathbb{R}_+ -conséquences de L sont donc des combinaisons linéaires des lignes de A . Comme nous souhaitons combiner les lignes de A , et non ses colonnes, nous allons utiliser la matrice transposée A^T . Fixons une séquence de variables fraîches \mathbf{z} dont le nombre est égal au nombre

de lignes. Les combinaisons linéaires des lignes de A (le *rowspace*) peuvent être vues comme l'ensemble de solutions entières suivant :

$$\text{sol}^{\mathbb{Z}}(\exists \mathbf{y}. A^T \mathbf{y} \doteq \mathbf{z})$$

Puisque le *rowspace* est le complément orthogonal du *nullspace* de A , $\text{sol}^{\mathbb{Z}}(A\mathbf{x} \doteq 0)$, tout vecteur correspond à une \mathbb{R}_+ -conséquence de L doit être orthogonal à chaque vecteur du *nullspace*, et donc à n'importe quelle base de celui-ci.

Soit A^\perp une base du *nullspace* de A , qui peut être calculer par le biais de l'algorithme de Gauß. Le *rowspace* est ensuite donné par :

$$\text{sol}^{\mathbb{Z}}(A^\perp \mathbf{z} \doteq 0)$$

Puisque nous sommes uniquement intéressé par le sous-ensemble de solutions de ce système qui sont non nulles, normalisées et avec un support minimal, il s'agit dans ce cas particulier de calculer les modes élémentaires du complément orthogonal du *nullspace* de A . Il faut prendre garde à calculer aussi les solutions non positives pour ce cas.

L'outil habituel de calcul des modes élémentaires peut donc être utilisé sur A^\perp pour calculer ces modes élémentaires "réversibles" [Terzer and Stelling \(2008\)](#). De manière équivalente, ce problème peut se ramener au calcul des direction extrêmes d'un cône et être résolu avec n'importe quelle librairie concernant les polytopes telles que [Avis \(2000\)](#) ; [Fukuda and Prodon \(1995\)](#).

8.2 Résultats expérimentaux

Comparons expérimentalement les trois algorithmes pour surapproximer l'objectif $\{\beta|_V \mid \beta \in h_{\Delta_6} \circ \text{sol}_{V'}^{\mathbb{R}_+^2}(L(\mathbf{y})) \cap \text{sol}_V^{\Delta_6}(C(\mathbf{y}'))\}$ étant donné un système linéaire d'équations $L(\mathbf{y})$, des contraintes cinétiques $C(\mathbf{y}')$, et un ensemble de variables observables V . Le premier algorithme applique directement l'interprétation abstraite pure à $L(\mathbf{y})$ pour calculer $\text{sol}^{\Delta_6}(\exists \bar{V}. (L(\mathbf{y}) \wedge C(\mathbf{y}')))$ par la programmation par contraintes sur un domaine fini où $\bar{V} = \text{fv}(L(\mathbf{y})) \setminus V$, comme l'énonce le Théorème [34](#) il surapproxime l'objectif. Le second algorithme enrichit le système linéaire $L(\mathbf{y})$ avec ses conséquences linéaires à support minimal, voir la Section [8.1](#), avant d'appliquer l'interprétation abstraite.

Le troisième algorithme est l'algorithme exact qui découle du Théorème [109](#).

Les résultats expérimentaux sont synthétisés dans la [FIGURE 8.1](#). La première instance vérifie nos attentes sur le réseau métabolique simple contenant un cycle, voir la [FIGURE 1.12](#), sans contrainte cinétique et $V = \{X, Y\}$ représente les variables observables. L'algorithme exact montre qu'il y a 6 solutions abstraites, une pour

Réseau	Type	Abstr. pure Interpr.	Support min. Conséquences	Exact
Cycle simple (FIGURE 1.12)	solutions	19	6	6
surproduction de Leucine	knockouts	16	14	14
	solutions	292	228	228
Contre-exemple	solutions	≥ 10000	4454	4374

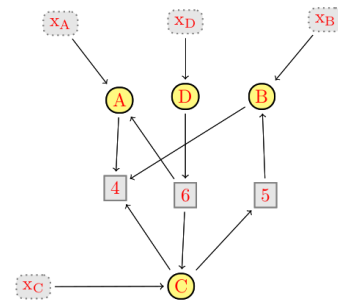
FIGURE 8.1 – Prédications pour l’analyse de réseaux de ce papier, obtenu respectivement par l’interprétation abstraite pure, l’heuristique basée sur les conséquences à support minimale et l’algorithme exact.

chaque valeur de Δ_6 . L’heuristique avec les conséquences linéaires à support minimal trouve exactement les mêmes 6 solutions abstraites. Et en appliquant l’interprétation abstraite pure on trouve 19 solutions abstraites (sur les 36 assignements possibles), c’est donc une grande surapproximation.

La seconde instance à l’échelle traite de la surproduction de la Leucine avec le réseau de la Figure FIGURE 1.11. L’heuristique et l’algorithme exact produisent le même résultat avec 226 solutions abstraites, alors que pour l’interprétation abstraite pure 292 solutions abstraites ont été trouvées. Ainsi, les deux nouveaux algorithmes enlèvent les 2 mêmes *knok-outs* invalides qui étaient retournés par l’algorithme de base de l’interprétation abstraite pure.

Cependant, l’heuristique avec les conséquences à support minimal n’est pas toujours exacte : Nous avons trouvé un contre exemple, donné sur la droite, qui surapproxime légèrement l’ensemble de solutions exactes.

D’un autre côté, l’algorithme (heuristique) avec les *EFM-consequences* est bien plus rapide que l’algorithme exact –pour référence sur la surapproximation de la Leucine, on met 5 minutes au lieu de 5 heures – tout en restant équivalent dans la plupart des cas.



Conclusion et perspectives

L'objectif principal de cette thèse était de pouvoir proposer un algorithme qui calcule de façon exacte l'*abstraction de systèmes linéaires* dans le but d'améliorer l'*approche qualitative* pour la prédiction de *knock-outs de gènes* qui entraînent une surproduction d'un métabolite cible pour un organisme donné.

Conclusion

Pour ce faire, nous avons introduit la notion de Σ -abstraction en utilisant des homomorphismes entre Σ -structures relationnelles. Cela a permis de considérer des abstractions de différences comme des homomorphismes de la Σ -structure \mathbb{R}_+^2 vers des Σ -structures finies.

Puis, nous avons développé des *techniques* de logique du premier-ordre, qui permettent de décomposer des abstractions de différences dans l'abstraction booléenne et des applications de fonctions définies en logique du premier-ordre.

Une première avancée essentielle a été de montrer formellement qu'il était possible de réécrire des *systèmes mixtes* comme des formules *exactes* pour l'abstraction booléenne. Cette réécriture est basée sur une application non triviale des modes élémentaires.

Une autre contribution importante est le développement de l'heuristique à support minimal, qui consiste en une application des modes élémentaires différente de la première, et la comparaison de la précision de l'heuristique à support minimal avec l'algorithme exacte.

Ces deux algorithmes montrent la même précision pour les applications existantes. Par contre, cette heuristique est plus efficace en temps de calcul, elle ne prend que quelques minutes là où l'algorithme exact mettra quelques heures.

Pour montrer cela, nous avons implanté l'algorithme exact et l'heuristique à support minimal, et nous les avons intégrés à l'outil Reactions-network de l'équipe

BioComputing.

Perspectives

Premièrement, il serait intéressant de raffiner les abstractions de différence vers un traitement plus quantitatif. Dans ce cas, il faudrait des systèmes de réactions chimiques avec des informations cinétiques plus quantitatives également. Il serait possible, par exemple, de préciser les types de cinétique. Concernant le raffinement des abstractions, on peut penser à des méthodes par intervalles.

Deuxièmement, être en mesure de prédire des knock-outs multiples aurait un grand impact, comme de telles prédictions ne peuvent pas être effectuées *in vivo* compte tenu de la combinatoire du problème. Cependant, l'obtention de telle capacité demanderait une augmentation de la précision des algorithmes de prédiction existants conséquente.

Troisièmement, même si l'algorithme exact calcule les abstractions de différences sans perte, il est possible d'imaginer l'ajout de contraintes supplémentaires pour améliorer les prédictions de l'algorithme. Ou alors, on peut chercher à trouver des cibles plus pertinentes, comme par exemple, comment supprimer tous les *feedbacks négatifs* dans la régulation du métabolite à surproduire.

Et finalement, les algorithmes mis en place devraient être testés sur d'autres modèles biologiques pour avoir plus de retours expérimentaux. Cela nécessite la modélisation de réseaux s'intéressant à d'autres métabolites ou micro-organismes, ou encore à développer des applications dans d'autres domaines, tel que la santé par exemple.

Table des figures

1.1	Modèle d'EDOs de la dégradation enzymatique.	16
1.2	Trace de la simulation du modèle d'EDOs de la FIGURE 1.1 avec les valeurs des concentrations initiales fixées comme suit : $C(0) = 0, P(0) = 0, E(0) = 10, S(0) = 20$; et les valeurs des paramètres : $k_1 = 10.0, k_{-1} = 5.0, k_2 = 8.0$	16
1.3	Exemple de modèle informel du SubtiWiki, représentant une partie de la voie de synthèse des acides gras chez la bactérie B. Subtilis.	18
1.4	Graphe de la réaction chimique avec cinétique (n).	18
1.5	Graphe du réseau de réactions avec cinétique de la dégradation enzymatique, avec les paramètres cinétiques k_1, k_2, k_{-1} respectifs des réactions nommées 1, 2, et -1	19
1.6	Graphe du sous-réseau métabolique de la bactérie B. Subtilis produisant la leucine à partir de pyruvate.	23
1.7	La réaction nommée 1 contient un substrat S , un produit P , un activateur A et un inhibiteur I . La réaction 1 est en gris et n'est donc pas candidate pour un knock-out. La réaction 2 est similaire sauf qu'elle est en orange et est donc candidate pour un knock-out.	24
1.8	L'abstraction de différences $h_{\Delta_3} : \mathbb{R}_+^2 \rightarrow \Delta_3$	26
1.9	L'abstraction de différences $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_6$	26
1.10	Représentation graphique du réseau de réactions avec cinétique partielle pour la régulation d'un promoteur du métabolisme de la bactérie B. Subtilis.	27
1.11	Le graphe d'une partie d'un réseau métabolique de la bactérie B. Subtilis avec sa régulation, produisant la Leucine.	40
1.12	Réseau métabolique avec un cycle simple.	41
1.13	Les trois solutions en Δ_6 du système linéaire ϕ compatibles avec $out-B \doteq \uparrow$	41

2.1	Interprétation ensembliste des Σ -expressions. Étant donné une Σ -structure S et un assignement de variables $\alpha : V \rightarrow \text{dom}(S)$ pour toutes les variables libres nous définissons l'interprétation $\llbracket e \rrbracket^{\alpha, S} \subseteq \text{dom}(S)$	49
2.2	Formules de la logique du premier-ordre	56
2.3	Interprétation booléenne des formules $\llbracket \phi \rrbracket^{\alpha, S} \in \mathbb{B}$	56
3.1	La Σ -abstraction $h_{\Delta_3} : \mathbb{R}_+^2 \rightarrow \Delta_3$	62
3.2	La Σ -abstraction $h_{\Delta_6} : \mathbb{R}_+^2 \rightarrow \Delta_3$	62
3.3	Interprétation de la Σ -structure Δ_3	63
3.4	Interprétation de la Σ -structure Δ_6	64
3.5	Les expressions et formules de la logique du premier-ordre avec n -uplets.	65
3.6	Sémantique des Σ -formules du premier-ordre avec n -uplets, où S est une Σ -structure et $\beta : \mathcal{V} \rightarrow \text{dom}(S)^n$ un affectation de variables.	65
4.1	Graphe du réseau de réactions de la dégradation enzymatique.	81
4.2	Le réseau de la dégradation enzymatique.	81
4.3	La matrice de stœchiométrie du réseau de la dégradation enzymatique.	81
4.4	Représentation graphique d'un réseau de réactions simple.	84
4.5	Le réseau de réactions de l'exemple simple.	84
4.6	La matrice de stœchiométrie du réseau de l'exemple simple.	84
4.7	Graphe du réseau de réactions avec cinétique de la dégradation enzymatique, avec $k_1, k_2, k_{-1} \in \mathcal{E}_{\Sigma_{arith}}$ les paramètres cinétiques respectifs des réactions $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_{-1}$	86
4.8	Graphique représentant l'évolution de la cinétique de la réaction \mathbf{r}_{MA_k} entre deux états stationnaires différents en fonction de la concentration de l'espèce réactante S	88
4.9	Représentation graphique du réseau de réaction avec cinétique partielle $\mathbf{P}_{Ilv-Leu}$ pour la régulation du promoteur $P_{Ilv-Leu}$	94
4.10	Réseau de réactions $\text{sim}(\mathbf{P}_{Ilv-Leu})$ avec cinétique modulo similarité.	94
4.11	Système de contraintes non linéaires de similarité cinétique $C_{\mathbf{P}_{Ilv-Leu}}$: des $\Sigma_{\text{pos-arith}}[\text{dom}(\Delta_6)]$ -équations à interpréter sur Δ_6	95
4.12	Système d'équations linéaires $L_{\mathbf{P}_{Ilv-Leu}}$ de $\Sigma_{\text{pos-arith}}$ -équations à l'état stable, à interpréter sur \mathbb{R}_+	95
4.13	Représentation graphique d'une partie du métabolisme de la bactérie <i>B. subtilis</i> avec sa régulation.	99
4.14	Modèle du métabolisme de la bactérie <i>B. subtilis Leu</i>	101

5.1	Explication de la prédiction du changement d'influx $v_{in-CodY} = \downarrow$ pour obtenir la cible $v_{out-Leu} = \uparrow$	104
5.2	Explication de la prédiction du changement d'influx $v_{in-TrnA} = \downarrow$ pour obtenir la cible $v_{out-Leu} = \uparrow$	105
5.3	Le problème algorithmique ABSTRACTION EXACTE.	108
5.4	Le problème algorithmique APPROXIMATION DE JOHN.	108
5.5	Définition de l'opérateur abstrait de somme en Δ_6 dans le langage MiniZinc.	110
5.6	Ensemble de solutions abstraites calculé à partir de l'ensemble de contraintes avec cible inféré du réseau $P_{Ily-Leu}$ sur Δ_6	111
5.7	Table des 11 meilleures solutions de l'heuristique approximative de John.	112
5.8	Modèle du métabolisme avec régulation de la bactérie <i>B. Subtilis Leu</i>	116
5.9	Table de solutions pour la prédiction de single knock-out pour le réseau de réaction <i>Leu</i> avec l'approximation de John.	117
5.10	Comparatif des algorithmes pour la prédiction de double knock-out du réseau de réactions représentant le cycle carbone des acides aminés.	117
6.1	Function Python approximant l'intégrale $\int_0^a f(x)dx$ pour une fonction donnée $f : \mathbb{R} \rightarrow \mathbb{R}$	122
7.1	Deux exemples de projection sur la diagonale: $y_1 = proj_G^{\mathbb{R}_+^2}(x_1)$ et $y_2 = proj_G^{\mathbb{R}_+^2}(x_2)$	139
8.1	Prédictions pour l'analyse de réseaux de ce papier, obtenu respectivement par l'interprétation abstraite pure, l'heuristique basée sur les conséquences à support minimale et l'algorithme exact.	154

Références

- Allart, E., Niehren, J., and Versari, C. (2019). Computing difference abstractions of metabolic networks under kinetic constraints. In Bortolussi, L. and Sanguinetti, G., editors, *Computational Methods in Systems Biology - 17th International Conference, CMSB 2019, Trieste, Italy, September 18-20, 2019, Proceedings*, volume 11773 of *Lecture Notes in Computer Science*, pages 266–285. Springer.
- Arkin, A., Ross, J., and McAdams, H. H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage λ -infected escherichia coli cells. *Genetics*, 149(4):1633–1648.
- Avis, D. (2000). A revised implementation of the reverse search vertex enumeration algorithm. In *Polytopes—combinatorics and computation*, pages 177–198. Springer.
- Avis, D. and Jordan, C. (2018). mplrs: A scalable parallel vertex/facet enumeration code. *Mathematical Programming Computation*, 10(2):267–302.
- Bagnara, R., Hill, P. M., and Zaffanella, E. (2008). The parma polyhedra library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Sci. Comput. Program.*, 72(1-2):3–21.
- Baier, C. and Katoen, J.-P. (2008). *Principles of model checking*. MIT Press, Cambridge, Mass.
- Batmanov, K. (2014). *Spatial modelling of community effects. (Modélisation spatiale des effets de communauté)*. PhD thesis, Lille University of Science and Technology, France.
- Bonarius, H. P., Schmid, G., and Tramper, J. (1997). Flux analysis of underdetermined metabolic networks: the quest for the missing constraints. *Trends in Biotechnology*, 15(8):308–314.
- Broy, M. and Steinbrüggen, R. (1999). The calculational design of a generic abstract interpreter. *Calculational system design*, 173:421.

- Burgard, A. P., Pharkya, P., and Maranas, C. D. (2003). Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and bioengineering*, 84(6):647–657.
- Calzone, L., Fages, F., and Soliman, S. (2006). BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807.
- Chelliah, V., Laibe, C., and Le Novère, N. (2013). Biomodels database: a repository of mathematical models of biological processes. In *In Silico Systems Biology*, pages 189–199. Springer.
- Cohen, I. R. and Harel, D. (2007). Explaining a complex living system: dynamics, multi-scaling and emergence. *Journal of the royal society interface*, 4(13):175–182.
- Cousot, P. and Halbwachs, N. (1978). Automatic discovery of linear restraints among variables of a program. In Aho, A. V., Zilles, S. N., and Szymanski, T. G., editors, *Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages, Tucson, Arizona, USA, January 1978*, pages 84–96. ACM Press.
- Coutte, F., Niehren, J., Dhali, D., John, M., Versari, C., and Jacques, P. (2015). Modeling Leucine’s Metabolic Pathway and Knockout Prediction Improving the Production of Surfactin, a Biosurfactant from *Bacillus Subtilis*. *Biotechnology Journal*, 10(8):1216–34.
- Covert, M. W., Famili, I., and Palsson, B. O. (2003). Identifying constraints that govern cell behavior: a key to converting conceptual to computational models in biology? *Biotechnology and bioengineering*, 84(7):763–772.
- Covert, M. W., Schilling, C. H., and Palsson, B. (2001). Regulation of gene expression in flux balance models of metabolism. *Journal of theoretical biology*, 213(1):73–88.
- Dalla Man, C., Rizza, R. A., and Cobelli, C. (2007). Meal simulation model of the glucose-insulin system. *IEEE Transactions on biomedical engineering*, 54(10):1740–1749.
- Danos, V., Feret, J., Fontana, W., Harmer, R., and Krivine, J. (2010). Abstracting the differential semantics of rule-based models: exact and automated model reduction. In *2010 25th Annual IEEE Symposium on Logic in Computer Science*, pages 362–381. IEEE.
- De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103.
- Dines, L. L. (1926). On positive solutions of a system of linear equations. *Annals of Mathematics*, 28(1/4):386–392.
- Fages, F., Gay, S., and Soliman, S. (2015). Inferring reaction systems from ordinary differential equations. *Theor. Comput. Sci.*, 599:64–78.
- Fages, F. and Soliman, S. (2008). Abstract interpretation and types for systems

- biology. *Theor. Comput. Sci.*, 403(1):52–70.
- Fages, F., Soliman, S., and Chabrier-Rivier, N. (2004). Modelling and querying interaction networks in the biochemical abstract machine biocham. *Journal of Biological Physics and Chemistry*, 4:64–73.
- Feinberg, M. (1987). Chemical reaction network structure and the stability of complex isothermal reactors—I. the deficiency zero and deficiency one theorems. *Chemical Engineering Science*, 42(10):2229 – 2268.
- Feinberg, M. (2019). *Foundations of chemical reaction network theory*. Springer.
- Florez, L., Gunka, K., Polania, R., Tholen, S., and Stulke, J. (2011). SPABBATS: A pathway-discovery method based on Boolean satisfiability that facilitates the characterization of suppressor mutants. *BMC Systems Biology*, 5(1):5+.
- Flórez, L. A., Gunka, K., Polanía, R., Tholen, S., and Stülke, J. (2011). Spabbats: A pathway-discovery method based on boolean satisfiability that facilitates the characterization of suppressor mutants. *BMC systems biology*, 5(1):5.
- Forbus, K. D. (1997). Qualitative reasoning. In Tucker, A. B., editor, *The Computer Science and Engineering Handbook*, pages 715–733. CRC Press.
- Fukuda, K. (1993). cdd. c: C-implementation of the double description method for computing all vertices and extremal rays of a convex polyhedron given by a system of linear inequalities. *Department of Mathematics, Swiss Federal Institute of Technology, Lausanne, Switzerland*.
- Fukuda, K. and Prodon, A. (1995). Double description method revisited. In *Franco-Japanese and Franco-Chinese Conference on Combinatorics and Computer Science*, pages 91–111. Springer.
- Gagneur, J. and Klamt, S. (2004). Computation of elementary modes: a unifying framework and the new binary approach. *BMC bioinformatics*, 5(1):1.
- Gianchandani, E. P., Papin, J. A., Price, N. D., Joyce, A. R., and Palsson, B. O. (2006). Matrix Formalism to Describe Functional States of Transcriptional Regulatory Systems. *PLoS Comput Biol*, 2(8):e101+.
- Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434.
- Granger, P. (1991). Static analysis of linear congruence equalities among variables of a program. In Abramsky, S. and Maibaum, T. S. E., editors, *TAPSOFT'91: Proceedings of the International Joint Conference on Theory and Practice of Software Development, Brighton, UK, April 8-12, 1991, Volume 1: Colloquium on Trees in Algebra and Programming (CAAP'91)*, volume 493 of *Lecture Notes in Computer Science*, pages 169–192. Springer.
- Herajy, M., Liu, F., Rohr, C., and Heiner, M. (2017). Snoopy’s hybrid simulator: a tool to construct and simulate hybrid biological models. *BMC Syst. Biol.*, 11(1):71:1–71:16.
- Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu,

- L., Mendes, P., and Kummer, U. (2006). Copasi—a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074.
- Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., et al. (2003). The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531.
- John, M., Lhoussaine, C., Niehren, J., and Uhrmacher, A. M. (2008). The attributed pi-calculus. In Heiner, M. and Uhrmacher, A. M., editors, *Computational Methods in Systems Biology, 6th International Conference, CMSB 2008, Rostock, Germany, October 12-15, 2008. Proceedings*, volume 5307 of *Lecture Notes in Computer Science*, pages 83–102. Springer.
- John, M., Nebut, M., and Niehren, J. (2013a). Knockout Prediction for Reaction Networks with Partial Kinetic Information. In *14th International Conference on Verification, Model Checking, and Abstract Interpretation*, pages 355–374, Rom, Italy.
- John, M., Nebut, M., and Niehren, J. (2013b). Knockout Prediction for Reaction Networks with Partial Kinetic Information. In *14th International Conference on Verification, Model Checking, and Abstract Interpretation*, Rom, Italy.
- Kaern, M., Elston, T. C., Blake, W. J., and Collins, J. J. (2005). Stochasticity in gene expression: from theories to phenotypes. *Nature Reviews Genetics*, 6(6):451–464.
- Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., et al. (2007). Kegg for linking genomes to life and the environment. *Nucleic acids research*, 36(suppl_1):D480–D484.
- Karp, P. D., Riley, M., Saier, M., Paulsen, I. T., Collado-Vides, J., Paley, S. M., Pellegrini-Toole, A., Bonavides, C., and Gama-Castro, S. (2002). The eco-cyc database. *Nucleic acids research*, 30(1):56–58.
- Kauffman, K. J., Prakash, P., and Edwards, J. S. (2003). Advances in flux balance analysis. *Current opinion in biotechnology*, 14(5):491–496.
- Kemeny, J. G. and Snell, J. L. (1976). *Markov chains*. Springer-Verlag, New York.
- Kim, J. and Reed, J. (2010). OptORF: Optimal metabolic and regulatory perturbations for metabolic engineering of microbial strains. *BMC Systems Biology*, 4(1):53+.
- Kitano, H. (2002). Computational systems biology. *Nature*, 420(6912):206–210.
- Klamt, S. and Stelling, J. (2003). Two approaches for metabolic pathway analysis? *Trends in biotechnology*, 21(2):64–69.
- Klamt, S., Stelling, J., Ginkel, M., and Gilles, E. D. (2003). FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. *Bioinformatics*, 19(2):261–269.

- Kolmogorov, A., Petrovskii, I., and Piskunov, N. (1937). A study of the diffusion equation with increase in the amount of substance, and its application to a biological problem in selected works of an kolmogorov. 1:242–270.
- Komei, F. and Alain, P. (1996). Double description method revisited. In Deza, M., Euler, R., and Manoussakis, I., editors, *Combinatorics and Computer Science*, pages 91–111, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Kuttler, C. and Niehren, J. (2006). Gene Regulation in the Pi Calculus: Simulating Cooperativity at the Lambda Switch. pages 24–55.
- Lee, J. M., Gianchandani, E. P., and Papin, J. A. (2006). Flux balance analysis in the era of metabolomics. *Briefings in bioinformatics*, 7(2):140–150.
- Leloup, J.-C. and Goldbeter, A. (2000). Modeling the molecular regulatory mechanism of circadian rhythms in drosophila. *BioEssays*, 22(1):84–93.
- Lotz, K., Hartmann, A., Grafahrend-Belau, E., and Schreiber, F. and Junker, B. (2014). Elementary flux modes, flux balance analysis, and their application to plant metabolism. *Plant Metabolism. Methods in Molecular Biology (Methods and Protocols)*.
- Madelaine, G. (2017). *Simplifications Exactes et Structurelles de Réseaux de Réactions Biologiques*. Theses, Université de Lille.
- Madelaine, G., Lhoussaine, C., and Niehren, J. (2014). Attractor Equivalence: An Observational Semantics for Reaction Networks. In *First International Conference on Formal Methods in Macro-Biology*, Lecture Notes in Bioinformatics, pages 82–101, Nouméa, New Caledonia. Springer-Verlag.
- Mäder, U., Schmeisky, A. G., Florez, L. A., and Stülke, J. (2012). Subti wiki—a comprehensive community resource for the model organism bacillus subtilis. *Nucleic acids research*, 40(D1):D1278–D1287.
- Mandon, H., Su, C., Haar, S., Pang, J., and Paulevé, L. (2019). Sequential reprogramming of boolean networks made practical. In *International Conference on Computational Methods in Systems Biology*, pages 3–19. Springer.
- Maranas, C. D. and Zomorodi, A. R. (2016). *Flux Balance Analysis and LP Problems*, chapter 3, pages 53–80. Wiley-Blackwell.
- Miné, A. (2002). A few graph-based relational numerical abstract domains. In Hermenegildo, M. V. and Puebla, G., editors, *Static Analysis, 9th International Symposium, SAS 2002, Madrid, Spain, September 17-20, 2002, Proceedings*, volume 2477 of *Lecture Notes in Computer Science*, pages 117–132. Springer.
- Niehren, J., John, M., Versari, C., Coutte, F., and Jacques, P. (2015). Qualitative Reasoning about Reaction Networks with Partial Kinetic Information. In *Computational Methods for Systems Biology*, page 12, Nantes, France.
- Niehren, J., Versari, C., John, M., Coutte, F., and Jacques, P. (2016). Predicting Changes of Reaction Networks with Partial Kinetic Information. *BioSystems*, 149:113–124.

- Orth, J. D., Thiele, I., and Palsson, B. O. (2010). What is flux balance analysis? *Nature biotechnology*, 28(3):245–248.
- Palsson, B. (2000). The challenges of in silico biology. *Nature biotechnology*, 18(11):1147–1150.
- Papin, J. A., Stelling, J., Price, N. D., Klamt, S., Schuster, S., and Palsson, B. O. (2004). Comparison of network-based pathway analysis methods. *Trends in biotechnology*, 22(8):400–405.
- Patil, K. R., Rocha, I., Förster, J., and Nielsen, J. (2005). Evolutionary programming as a platform for in silico metabolic engineering. *BMC bioinformatics*, 6(1):308.
- Paulevé, L., Kolcák, J., Chatain, T., and Haar, S. (2020). Reconciling qualitative, abstract, and scalable modeling of biological networks. *bioRxiv*.
- Paulevé, L. and Richard, A. (2012). Static analysis of boolean networks based on interaction graphs: A survey. *Electronic Notes in Theoretical Computer Science*, 284:93 – 104. Proceedings of the 2nd International Workshop on Static Analysis and Systems Biology (SASB 2011).
- Petri, C. A. (1966). Communication with automata.
- Pharkya, P., Burgard, A. P., and Maranas, C. D. (2004). Optstrain: a computational framework for redesign of microbial production systems. *Genome research*, 14(11):2367–2376.
- Pharkya, P. and Maranas, C. D. (2006). An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems. *Metabolic engineering*, 8(1):1–13.
- Price, N. D., Reed, J. L., and Palsson, B. Ø. (2004). Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature Reviews Microbiology*, 2(11):886–897.
- Ramakrishna, R., Edwards, J. S., McCulloch, A., and Palsson, B. O. (2001). Flux-balance analysis of mitochondrial energy metabolism: consequences of systemic stoichiometric constraints. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 280(3):R695–R704.
- Ranganathan, S., Suthers, P. F., and Maranas, C. D. (2010). OptForce: An Optimization Procedure for Identifying All Genetic Manipulations Leading to Targeted Overproductions. *PLoS Comput Biol*, 6(4):e1000744+.
- Rao, A., Hero III, A. O., States, D. J., and Engel, J. D. (2007). Using directed information to build biologically relevant influence networks. In *Computational Systems Bioinformatics: (Volume 6)*, pages 145–156. World Scientific.
- Regev, A., Silverman, W., and Shapiro, E. (2000). Representation and simulation of biochemical processes using the π -calculus process algebra. In *Biocomputing 2001*, pages 459–470. World Scientific.
- Rendl, A., Guns, T., Stuckey, P. J., and Tack, G. (2015). Minisearch: A solver-independent meta-search language for minizinc. In Pesant, G., editor, *Prin-*

- principles and Practice of Constraint Programming - 21st International Conference, CP 2015, Cork, Ireland, August 31 - September 4, 2015, Proceedings*, volume 9255 of *Lecture Notes in Computer Science*, pages 376–392. Springer.
- Schulte, C. and Carlsson, M. (2006). Finite domain constraint programming systems. In *Foundations of Artificial Intelligence*, volume 2, pages 495–526. Elsevier.
- Schulte, C., Tack, G., and Lagerkvist, M. Z. (2010). Modeling and programming with gecode. *Schulte, Christian and Tack, Guido and Lagerkvist, Mikael*, 1.
- Schuster, S. and Hilgetag, C. (1994). On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, 2(02):165–182.
- Segre, D., Vitkup, D., and Church, G. M. (2002). Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences*, 99(23):15112–15117.
- Terzer, M. and Stelling, J. (2008). Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics*, 24(19):2229–2235.
- Thieffry, D. and De Jong, H. (2002). Modélisation, analyse et simulation des réseaux génétiques-bio-informatique (3). *médecine/sciences*, 18(4):492–502.
- Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563 – 585.
- Théâtre, A., Cano-Prieto, C., and Bartolini, M. (2020). The surfactin-like lipopeptides from bacillus spp.: natural biodiversity and synthetic biology for a broader application range.
- T.S., M., H., R., Thompson, G., and R.M., T. (1953). The double description method. In Kuhn, H. and A.W.Tucker, editors, *Contributions to theory of games*, volume 2. Princeton University Press.
- Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72.
- Tyson, J. J. and Novak, B. (2001). Regulation of the eukaryotic cell cycle: molecular antagonism, hysteresis, and irreversible transitions. *Journal of theoretical biology*, 210(2):249–263.
- Varma, A. and Palsson, B. O. (1993). Metabolic capabilities of escherichia coli ii. optimal growth patterns. *Journal of Theoretical Biology*, 165(4):503–522.
- Varma, A. and Palsson, B. O. (1994). Metabolic flux balancing: basic concepts, scientific and practical use. *Bio/technology*, 12(10):994–998.
- Zanghellini, D., Ruckerbauer, D. E., Hanscho, M., and Jungreuthmayer, C. (2013). Elementary flux modes in a nutshell: Properties, calculation and applications. *Biotechnology Journal*, pages 1009–1016.