Analyse statique et réduction de modèles de voies de signalisation intracellulaire

Jérôme Feret

Chapitre 1

Analyse statique et réduction de modèles de voies de signalisation intracellulaire

Jérôme Feret

Ce chapitre décrit deux applications des méthodes formelles pour Kappa, un langage de réécriture de graphes à sites utilisé en modélisation de systèmes biologiques. La première est une analyse statique pour détecter des motifs qui ne pourront jamais apparaître en cours d'exécution. La seconde réduit les modèles pour obtenir une représentation concise de leurs comportements sous forme de systèmes d'équations différentielles ordinaires.

1.1 Introduction

1.1.1 Contexte et motivations

Décrire et analyser les systèmes à grande échelle et fortement combinatoires qui sont issus de certains modèles mécanistiques de biologie des systèmes est encore hors de portée de l'état de l'art. Dans de tels modèles, le comportement individuel des occurrences de protéines, qui peuvent établir des liaisons et modifier leur capacité d'interactions, est influencé par des compétitions pour des ressources communes. De plus, les occurrences de protéines peuvent former une grande diversité de configurations d'espèces biochimiques différentes. La concurrence entre des interactions à différentes échelles de temps génère des boucles de rétro-actions non linéaires qui contrôlent l'abondance de ces configurations d'espèces biochimiques.

Enfin, ces systèmes font intervenir des interactions entre de très petites molécules, comme des ions ou des ligands et des espèces biochimiques gigantesques comme les brins d'acide désoxyribonucléique, le ribosome, ou le signalosome. Comprendre comment le comportement collectif des populations de protéines qui définit le phénotype, est engendré par le comportement individuel des occurrences de ces protéines reste un problème largement ouvert et un enjeu crucial.

Alors que les progrès technologiques permettent d'obtenir rapidement une quantité toujours plus importante de détails à propos des interactions mécanistiques potentielles entre les occurrences de protéines, et ce, à un prix très accessible, la communauté scientifique est encore bien loin de comprendre globalement comment le comportement macroscopique des systèmes émerge de ces interactions. C'est l'objectif annoncé de la biologie des systèmes. Mais ce but est sans espoir à moins que des méthodes spécifiques et innovantes pour décrire ces systèmes complexes et analyser leur propriété ne soient conçues. Bien entendu, ces méthodes devront passer à l'échelle de la très grande quantité d'informations qui est publiée dans la littérature à un rythme qui augmente de manière exponentielle.

1.1.2 Les langages de modélisation de systèmes d'interactions moléculaires

Les langages formels ont été beaucoup utilisés pour décrire des modèles d'interactions mécanistiques entre occurrences de protéines. Ils procurent des outils mathématiques pour traduire ces interactions et définir rigoureusement le comportement des systèmes ainsi représentés grâce à un choix de sémantiques qualitatives, stochastiques ou différentielles.

Les langages tels que les réseaux réactionnels [38] ou les réseaux de Petri classiques [53], se basent sur le paradigme de la réécriture multi-ensemble. Les interactions consistent à consommer des réactifs en échange de produits. Des constantes cinétiques permettent de préciser soit la vitesse, soit la fréquence moyenne – selon le choix de la sémantique – d'application des différentes réactions. Ceci les rend très utiles pour décrire et formaliser le comportement de systèmes d'interactions de petite ou moyenne taille. Cependant, ces langages peinent à représenter de grands modèles car ils ont besoin d'un nom (ou d'un emplacement dans le cas des réseaux de Petri) par type de configurations d'espèces biochimiques.

La réécriture de graphes à sites [34, 36, 4, 57] exploite le fait que les interactions dépendent généralement de conditions locales sur les configurations des occurrences de protéines au sein des espèces biochimiques. Ces langages permettent ainsi de traduire les systèmes d'interactions entre les

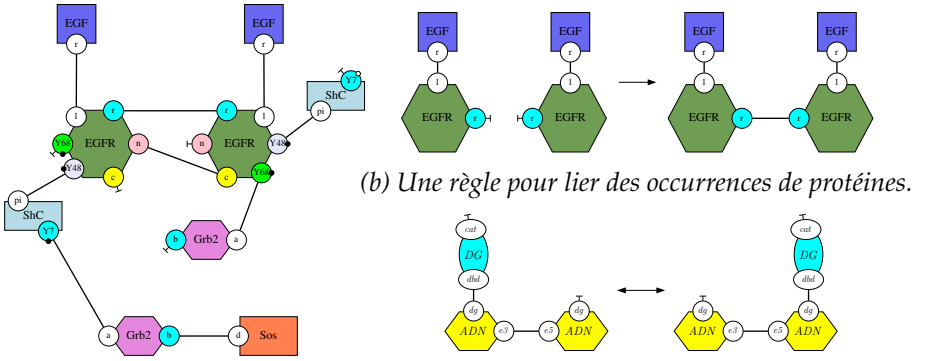
occurrences de protéines de manière plus parcimonieuse : seuls les détails qui importent pour une interaction donnée sont mentionnés.

Il est important de distinguer les approches basées sur les agents de celles basées sur les règles de réécriture. Dans les celles basées sur les agents, chaque entité, que ce soit un processus [21] ou un objet [35], doit contenir la description de tous ses comportements possibles. Les changements entre les configurations des différentes entités se synchronisent par le biais de règles de communication. Ces règles, généralement en très petit nombre, définissent la sémantique opérationnelle des langages. Il est possible de conditionner le comportement d'un agent à des propriétés de l'état d'un autre agent auquel cet agent serait lié, mais cela nécessite de recourir à des processus fictifs pour aller chercher cette information. Cette astuce était en fait déjà utilisée dans les premiers modèles décrits en π-calcul [68]. Cependant, en général, les approches basées sur les agents donnent lieu à des systèmes de processus à états finis [59]. Ceci permet d'étudier leur comportement à l'aide d'outils de vérification symbolique de modèles comme PRISM [63]. Lorsque les occurrences des protéines admettent trop de configurations différentes ou lorsque leurs capacités d'interactions dépendent trop des occurrences des protéines auxquelles elles sont liées, les approches fondées sur les agents ne passent pas à l'échelle, tant au niveau de la description des modèles que pour le calcul de leurs propriétés.

Les approches fondées sur les règles définissent les modèles par des règles d'interactions. Chaque règle définit sous quelles conditions sur les configurations des agents une interaction peut avoir lieu et quels sont les effets de cette interaction. Ainsi l'état des agents ne définit pas une fois pour toute les capacités d'interactions de cet agent. Ce sont les règles du modèle qui le font. Il n'est pas non plus nécessaire de donner la liste exhaustive de toutes les configurations des agents. Les règles peuvent se contenter de ne mentionner que les parties importantes des agents pour l'interaction qu'elles décrivent. Les approches fondées sur les règles passent mieux à l'échelle et facilitent la mise à jour des modèles. De plus, comme il n'est pas nécessaire de spécifier explicitement toutes les capacités d'interactions des occurrences des protéines, elles encouragent à une modélisation sans *a priori* où les interactions émergent des règles au fur et à mesure de la conception du modèle.

1.1.3 Le langage Kappa

Les langages de réécriture de graphes à sites [34, 36, 4, 57] permettent de représenter de manière transparente les réseaux d'interactions entre des occurrences de protéines grâce à leur syntaxe inspirée de la chimie.



(a) Un graphe à sites.

(b) Une règle pour lier des occurrences de protéines.

(c) Une règle pour faire glisser des occurrences de protéines.

FIGURE 1.1 – *En 1.1a est dessiné un graphe à site. Il s'agit de la configuration d'une espèce biochimique composée de deux occurrences du ligand (EGF), de deux occurrences du récepteur membranaire (EGFR), d'une occurrence de la protéine d'échafaudage (Shc), de deux occurrences de la protéine de transport (Grb2) et d'une occurrence de la protéine Sos. En 1.1b est donné un exemple de règle de liaison. Deux occurrences du récepteur membranaire (EGFR), lorsqu'elles sont toutes deux activés par une liaison avec des occurrences du ligand (EGF), peuvent se lier. Les autres sites sont omis car ils ne jouent aucun rôle dans cette interaction. En 1.1c est donnée une règle de déplacement. Une occurrence de l'enzyme Glycolase (DG) peut glisser dans les deux directions (selon une marche aléatoire) le long d'un brin d'ADN.*

Dans Kappa, chaque configuration d'espèces biochimiques est représentée par un graphe à sites. Un exemple de graphe à sites est donné en figure 1.1a. Dans un graphe à sites, des nœuds qui représentent des occurrences de protéines, sont associés à une liste de sites d'interactions. Ces sites peuvent être libres ou liés deux à deux. En outre, certains sites portent une propriété qui peut servir à représenter un niveau d'activation. Les interactions entre occurrences de protéines peuvent modifier leurs conformations en dépliant ou en repliant leurs chaînes de nucléotides, ce qui peut révéler ou cacher des sites d'interactions. Dans Kappa, la structure tri-dimensionnelle des occurrences de protéines n'est pas représentée explicitement. En revanche, les conditions pour qu'un site d'interactions soit visible sont spécifiées dans la description des interactions elles-mêmes.

L'évolution d'un système Kappa se décrit grâce à des règles de réécriture hors-contexte. En figure 1.1b est dessinée une règle pour la formation de dimères. Deux récepteurs (*EGFR*) qui sont tous deux liés à des ligands (*EGF*) peuvent se lier entre eux pour former un dimer. En figure 1.1c est

donnée une autre règle issue d'un modèle de réparation de l'ADN, dans laquelle une enzyme, la Glycolase (*DG*), peut glisser aléatoirement dans les deux sens, le long d'un brin d'ADN [61].

Une règle peut être comprise de manière intentionnelle comme une transformation locale de l'état du système ou de manière extensionnelle comme l'ensemble, qu'il soit fini ou non, des réactions biochimiques qui peuvent être obtenues en spécifiant entièrement les différents contextes d'application de ces règles. De cet ensemble de réactions, diverses sémantiques peuvent être définies pour décrire le comportement des systèmes. Ces sémantiques peuvent être qualitatives, stochastiques ou différentielles, comme pour le cas des réseaux réactionnels et des réseaux de Pétri (les sémantiques quantitatives — stochastiques ou différentielles — nécessitent d'associer une constante d'interactions à chaque règle). Il est toutefois possible de simuler un modèle Kappa directement, sans passer par le réseau réactionnel sous-jacent. La simulation consiste alors à itérer la boucle événementielle suivante (celle-ci correspond à l'algorithme de Gillespie [48]). Étant donné l'état du système, représenté par un graphe à sites, l'ensemble de tous les événements possibles est calculé. Un événement consiste à appliquer une règle dans le graphe à une occurrence du motif qui constitue le membre gauche de cette règle. Chaque événement a une propension qui correspond à la constante de la règle correspondante. Le prochain événement est tiré au hasard selon une probabilité proportionnelle à sa propension, alors que le délai entre deux événements est tiré aléatoirement selon une loi exponentielle dont le paramètre est la somme des propensions de tous les événements potentiels du système. Il n'est pas raisonnable de recalculer la liste des événements potentiels à chaque fois après l'application d'une règle. Cet ensemble peut être mis à jour dynamiquement en tenant compte uniquement des nouveaux événements potentiels et des événements qui ne sont plus possibles du fait de l'application du dernier événement choisi [31]. Le simulateur actuel tire profit au maximum des sous-motifs communs dans les motifs qui apparaissent dans le membre gauche des règles pour découvrir les nouveaux événements et retirer les événements devenus obsolètes plus rapidement [11].

Le langage Kappa souffre de plusieurs limites. Par exemple, les sites d'interactions d'une même occurrence d'une protéine doivent porter des noms différents ; par ailleurs, en ce qui concerne les propriétés géométriques, Kappa ne permet ni de représenter la structure tridimensionnelle des occurrences de protéines, ni leur répartition dans l'espace. Avoir des sites deux à deux différents dans chaque occurrence de protéines facilite grandement la recherche des occurrences des motifs dans les graphes, ce qui est non seulement crucial pour simuler les modèles de manière ef-

ficace, mais est aussi à la base de plusieurs constructions utilisées pour l'analyse statique et la réduction de modèles. Certains langages lèvent cette contrainte soit directement comme dans les langages BNGL [36] et mød [3], soit indirectement en utilisant un codage sous forme d'hyperliens comme dans le langage React(C) [57]. Toutefois, l'efficacité des moteurs de simulation est fortement réduite quand de telles constructions sont utilisées. Pour ce qui est de la géométrie des protéines, les conditions liées aux conformations spatiales des protéines peuvent être encodées dans les règles de réécriture. Certaines extensions du langage permettent de représenter des contraintes sur la position relative des occurrences de protéines et des sites d'interactions dans les configurations des espèces biochimiques afin de restreindre l'ensemble des événements possibles à ceux qui satisfont ces contraintes [33]. Enfin, dans Kappa, la distribution des occurrences de protéines dans l'espace est passée sous silence. Il est fait l'hypothèse que les occurrences de protéines sont parfaitement mélangées. Il est donc impossible de retrouver les phénomènes d'encombrement qui peuvent être dus à des accumulations d'occurrences de protéines dans certaines régions de la cellule. De même, les gradients de concentration locaux qui pourrait être dus à la présence d'une occurrence d'une protéine d'échafaudage ne peuvent pas être représentés (en Kappa, chaque occurrence d'une protéine d'échafaudage n'agit qu'en maintenant des occurrences de protéines dans la même espèce biochimique, une fois libérée, ces occurrences de protéines ne sont pas supposées rester, même pour un court instant dans le même voisinage). Une solution partielle consiste à encoder en Kappa une grille pour représenter de manière discrète les positions potentielles des occurrences de protéines. Ensuite, celles-ci peuvent glisser le long de cette grille grâce à des règles implémentant la diffusion des occurrences de protéines. Le langage SpatialKappa [70] utilise ce procédé de manière transparente. Par ailleurs, le langage ML [54] permet de représenter des modèles d'interactions entre occurrences de protéines qui peuvent se déplacer de manière continue dans un milieu. Il est possible de munir un modèle Kappa d'un ensemble de compartiments statiques. Toutefois, ceci ne permet pas de modéliser le transport d'occurrences de protéines par le biais de vésicules. La machine formelle cellulaire [27] répond à cet enjeu, sans toutefois fournir de moteurs de simulation efficaces.

Les langages de réécriture de graphes à sites permettent de représenter les réseaux d'interactions entre occurrences de protéines, et ce, malgré leur forte combinatoire. Si le comportement de ces réseaux peut être formellement défini et simulé, des abstractions restent nécessaires pour calculer les propriétés du comportement collectif des populations de protéines.

1.1.4 Interprétation abstraite

L'interprétation abstraite a été introduite il y a un peu plus de quarante ans comme un cadre mathématique pour établir des liens formels entre le comportement de programmes, vu à différents niveaux d'abstraction. Depuis, l'interprétation abstraite est utilisée non seulement pour comparer différentes méthodes et algorithmes d'analyse statique [24], mais aussi pour développer des analyseurs statiques pour calculer automatiquement les propriétés sur le comportement des programmes [6, 37]. L'interprétation abstraite s'est désormais développée dans l'industrie (entre autres, Amazon, Facebook, IBM, Google, MicroSoft et MathWorks ont chacune leurs propres analyseurs statiques basés sur l'interprétation abstraite).

L'interprétation abstraite repose sur la démarche suivante. Le comportement d'un programme (ou d'un modèle) peut en général être décrit comme le plus petit point fixe *lfp* $\mathbb{F}$ d'un opérateur $\mathbb{F}$ agissant sur les éléments d'un ensemble appelé le domaine concret D. Le domaine concret est habituellement l'ensemble des parties $\wp(S)$ d'un ensemble d'éléments S, qui peuvent être des états, des traces de calcul, *et cetera*. Une abstraction est alors vue comme un changement de granularité dans la description du comportement des programmes (ou des modèles) et ce changement de granularité peut prendre en langage mathématique diverses formes telles qu'un opérateur de clôture supérieure, une famille d'idéaux, une famille de Moore ou une correspondance de Galois. Les correspondances de Galois se sont vite imposées comme l'outil le plus populaire pour décrire une interprétation abstraite. Un changement du niveau d'observation du comportement d'un programme (ou d'un modèle) peut ainsi être décrit en choisissant un ensemble $D^\sharp$ de propriétés d'intérêt. C'est le domaine abstrait. Cet ensemble est ordonné par un ordre partiel $\sqsubseteq$. Chaque élément $a^\sharp$ de ce domaine abstrait représente intentionnellement l'ensemble des éléments concrets qui satisfont cette propriété. Cet ensemble est noté $\gamma(a^\sharp)$. La fonction γ, ainsi définie, est croissante (si $a^\sharp \sqsubseteq b^\sharp$, alors $\gamma(a^\sharp) \subseteq \gamma(b^\sharp)$). Ainsi, l'ordre $\sqsubseteq$ représente le niveau d'information.

Un élément abstrait $a^\sharp$ est dit être une abstraction d'un ensemble a d'éléments concrets, si et seulement si a est un sous-ensemble de l'ensemble $\gamma(a^\sharp)$. Une correspondance de Galois est obtenue quand chaque sous-ensemble a de l'ensemble S admet une meilleure abstraction, c'est à dire, que pour chaque partie a de l'ensemble S, il existe un élément abstrait, noté $\alpha(a)$ qui est d'une part une abstraction de l'ensemble a et d'autre part, qui est plus petit (pour l'ordre $\sqsubseteq$) que n'importe quelle abstraction de l'ensemble a. Dans un tel cas, n'importe quelle fonction croissante $\mathbb{F}^\sharp$ opérant sur le domaine abstrait $D^\sharp$ et telle que $[\alpha \circ \mathbb{F} \circ \gamma](a^\sharp) \sqsubseteq \mathbb{F}^\sharp(a^\sharp)$

pour chaque élément abstrait $a^\sharp \in D^\sharp$, admet un plus petit point fixe (pour l'ordre $\sqsubseteq$) noté $lfp\ \mathbb{F}^\sharp$. De plus, la concrétisation de ce plus petit point fixe est un sur-ensemble du plus petit point fixe de la fonction $\mathbb{F}$; ainsi le comportement du programme ou du modèle peut être calculé dans le domaine abstrait au prix d'une perte potentielle d'information puisque le résultat final est un sur-ensemble de l'ensemble de tous les comportements possibles. Par construction, l'approche est correcte : aucun comportement de la sémantique concrète n'est oublié. Par contre, quand le sur-ensemble ainsi calculé est un sur-ensemble strict, des comportements fictifs ont été introduits par l'analyse.

Le choix du domaine abstrait est crucial. Du point de vue de l'expressivité, le domaine abstrait doit permettre de décrire les propriétés d'intérêt des programmes (ou des modèles) ainsi que les propriétés intermédiaires qui sont nécessaires pour en établir la preuve de manière inductive. D'un point de vue algorithmique, ils doivent correspondre à des propriétés qui sont relativement simples à manipuler en machine. Enfin, la structure des chaînes croissantes d'éléments abstraits (pour l'ordre $\sqsubseteq$) est également importante pour que puissent être définis des opérateurs d'extrapolation précis, dans le cas où le domaine admettrait des chaînes croissantes infinies.

Plusieurs interprétations abstraites ont été proposées pour calculer automatiquement les propriétés des modèles en biologie des systèmes. Les premières ont été inspirées par les analyses de flot d'information [9, 39] et de dénombrement [65, 40] dans le π-calcul et le calcul des ambients. Ces analyses permettent de détecter avec précision dans quels compartiments des entités peuvent entrer dans des modèles-jouet de virus infectant des cellules. Elles trouvent également des exclusions mutuelles [49, 8]. Les analyses de dénombrement permettent aussi souvent de retrouver les invariants correspondant à la conservation du nombre de chaque sorte de protéines dans les réseaux réactionnels lorsque la composition des configurations d'espèces biochimiques n'est pas représentée explicitement [1]. Ces invariants sont aussi appelés invariants de places dans les réseaux de Petri.

Les modèles biologiques sont fortement concurrents et souffrent de l'explosion combinatoire dans le nombre d'entrelacements potentiels des différents événements possibles. L'interprétation abstraite a été utilisée pour oublier la séquentialité dans les traces d'exécution dans les processus de frappes [66], puis plus généralement pour les réseaux asynchrones discrets booléens ou multivalués [47]. Dans les modèles réseaux booléens ou multivalués, l'interprétation abstraite a également été utilisée pour calculer une approximation des ensembles constituant des trappes [22, 60], dans lesquels les systèmes ne peuvent plus sortir une fois entrés. Ces ensembles facilitent le calcul des trajectoires périodiques des modèles. Dans

les modèles de réseaux métaboliques, l'interprétation abstraite a été utilisée pour décrire une analyse de dépendances, qui calcule l'impact potentiel de l'inhibition éventuelle d'une règle sur la concentration à l'équilibre des composants du système [58, 2].

L'interprétation abstraite peut servir à la calibration d'un modèle [62], en réalisant une partition de l'espace des paramètres en trois régions : une première région dans laquelle le modèle satisfait une propriété temporelle donnée par l'utilisateur, une seconde qui ne la satisfait pas et une troisième pour laquelle l'analyse n'a pu conclure si la propriété était satisfaite ou non.

L'interprétation abstraite est également très utilisée pour le calcul des trajectoires des systèmes hybrides [50].

1.1.5 Contenu du chapitre

Le reste du chapitre décrit le langage Kappa [34] sous forme graphique, ainsi que l'analyse statique qui permet de détecter quels motifs peuvent se former lors de l'exécution des modèles [32, 41, 46, 10] et une méthode de réduction de modèles pour diminuer la complexité combinatoire de la sémantique différentielle de ces modèles [43, 30, 15, 18, 17, 14].

En particulier, la notion de graphe à sites, qui représente l'état des systèmes modélisés, est introduite Sect. 1.2, ainsi que celle de règle de réécriture. Par soucis de simplicité, seul un fragment du langage est considéré. En effet, certaines constructions du langage complet font intervenir des effets de bord (qui peuvent provoquer des transformations de l'état des occurrences de protéines, en dehors des occurrences des motifs de réécriture). S'il est possible d'adapter les différentes définitions pour traiter ces effets de bords, cela n'apporte pas grand chose conceptuellement.

L'analyse statique, présentée en section 1.3, permet de détecter, au sein d'un ensemble de motifs d'intérêt paramètre de l'analyse, lesquels ne peuvent jamais se former quelle que soit l'exécution du système. C'est une analyse approchée. Les motifs déclarés inaccessibles sont bien inaccessibles. Par contre, l'analyse n'apporte aucune information à propos des autres motifs. Par soucis d'efficacité, les ensembles de motifs sont organisés sous la forme d'une collection d'arbres de décision dans lesquels des motifs initiaux sont raffinés peu à peu en ajoutant de l'information contextuelle [46]. Cette analyse est implantée dans l'analyseur statique KaSa [10] et le choix des arbres de décisions, qui paramétrise l'analyse, est fait automatiquement par une pré-analyse.

En section 1.4 est présentée une méthode de réduction de modèles. Les sémantiques différentielles classiques introduisent une variable par catégorie d'espèces biochimiques, ce qui ne passe en général pas à l'échelle.

Le but de cette section est de proposer une méthode pour réduire la dimension de ces systèmes différentiels. Il s'agit donc d'identifier des quantités d'intérêt, en nombre réduit, dont l'évolution peut être décrite de manière autonome sans avoir besoin de la valeur des variables du système différentiel initial. Il s'agit donc de trouver un changement de variables qui réduise la taille de la sémantique différentielle des modèles décrit sous formes d'ensembles de règles de réécriture de graphes à sites. Pour cela, la méthode s'appuie sur la notion de flot d'information entre les sites des configurations d'espèces biochimiques, qui approche supérieurement quels sites influencent l'évolution de quels sites. Du flot d'information peuvent être découvertes des paires de sites pour lesquels la corrélation entre l'état n'a pas d'incidence sur la dynamique du système. Ces corrélation peuvent donc être oubliées, ce qui revient à considérer des morceaux de configurations d'espèces biochimiques, plutôt que des configurations d'espèces biochimiques en entier. Par construction, le résultat est un ensemble de portions de configuration d'espèces biochimiques dont l'évolution peut s'exprimer de manière autonome. L'interprétation de ces portions de configurations comme la combinaison linéaire des configurations d'espèces biochimiques dans lesquelles elles apparaissent pondérés par le nombre de leurs plongements, définit alors un changement de variables. Le système différentiel réduit, qui décrit l'évolution de la valeur de ces combinaisons linéaires peut être alors dérivé directement, sans avoir à générer le système différentiel initial.

Le chapitre se conclut en section 1.5 et quelques perspectives sont données. La description du langage et de l'analyse reste volontairement assez haut niveau. Une formalisation complète et rigoureuse est disponible dans les articles scientifiques qui sont cités dans le corps du texte.

1.2 Réécriture de graphes à sites

La section présente décrit, dans un premier temps, la notion de graphe à sites, qui permettra de représenter à la fois les différents états possibles pour les systèmes modélisés, mais aussi, les motifs qui seront utilisés, dans un second temps, pour décrire, grâce à des règles de réécriture, l'évolution de l'état de ces systèmes.

1.2.1 Signature

Il faut tout d'abord définir la signature des modèles. La signature d'un modèle décrit tous les ingrédients qui peuvent intervenir dans celui-ci.

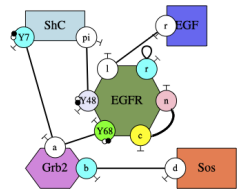


FIGURE 1.2 – *Une carte de contacts. Elle définit la signature d'un modèle en donnant la liste de toutes les sortes de protéines, leurs différents sites d'interactions, les différents états internes que peuvent prendre ces sites et les différentes liaisons potentielles entre ces sites.*

Elle peut être représentée graphiquement par une *carte de contacts*, comme celle dessinée en figure 1.2. Une carte de contacts comprend des nœuds pour représenter les différentes *sortes de protéines*. Ces nœuds sont nommés et adoptent des formes et des couleurs variées pour les distinguer plus facilement. Chaque sorte de protéines est associée à un ensemble de *sites d'interaction*. Ces sites sont représentés en périphérie de chaque sorte de protéines par des cercles colorés et nommés, eux-aussi. En Kappa, une sorte de protéines donnée ne peut avoir deux sites portant le même nom. Chaque site d'interactions est associé à un ensemble de pastilles colorées qui peuvent servir à représenter son *état d'activation*, comme par exemple le fait d'être – ou non – phosphorylé. Un état d'activation peut aussi éventuellement servir à représenter la localisation d'une occurrence d'une protéine au sein d'un ensemble fini et fixe de compartiments cellulaires. Les sites d'interactions peuvent également porter un *état de liaison* : les sites qui portent le symbole ⊣ peuvent potentiellement rester libre ; la carte de contacts contient aussi des arcs non-orientés entre les sites qui peuvent potentiellement être liés deux à deux. En particulier, un site peut être lié à plusieurs sites dans la carte de contacts (il sera expliqué plus tard que de telles liaisons sont en compétition). Par ailleurs, un site peut être lié à lui-même dans une carte de contacts (il sera expliqué plus tard que ceci signifie que deux sites de deux occurrences différentes d'une même sorte de protéines peuvent être liés entre-eux).

Exemple 1.2.1. *En figure 1.2 est donné un exemple de carte de contacts qui correspond aux premières interactions qui interviennent dans l'activation du facteur de croissance de l'épiderme. Cet exemple est inspiré d'un modèle BNGL disponible dans la littérature [7]. Ce modèle a été étendu pour décrire la liaison asymétrique entre les récepteurs EGFR et traduit en Kappa. Cette carte introduit cinq sortes de protéines : des ligands EGF, des récepteurs membranaires EGFR, des protéines d'échafaudage ShC, des protéines de transport Grb2 et des protéines cibles*

Sos (cette dernière sera ensuite phosphorylée ce qui initiera les étapes suivantes de la cascade d'interactions). Chaque occurrence du ligand EGF a un seul site qui est nommé r ; chaque occurrence du récepteur membranaire EGFR à six sites qui sont nommés respectivement l, r, c, n, Y48 et Y68 ; chaque occurrence de la protéine d'échafaudage ShC dispose de deux sites qui sont nommés respectivement Y7 et pi ; chaque occurrence de la protéine de transport Grb2 a deux sites qui sont respectivement nommés a et b ; enfin chaque occurrence de la protéine cible Sos a un seul site qui est nommé d. Seuls les sites Y48 et Y68 des occurrences de la protéine EGFR et le site Y7 des occurrences de la protéine ShC portent un état interne. Ces sites sont annotés par deux pastilles colorées, une blanche et une noire. La pastille blanche indique que ces sites peuvent être dans l'état non-phosphorylé, alors que la noire indique que ces sites peuvent être dans l'état phosphorylé. De plus, chaque site peut être libre (symbole ⊣) ou lié. Les liaisons possibles entre sites sont entre le site r d'une occurrence de la protéine EGF et le site l d'une occurrence de la protéine EGFR ; entre les sites r de deux occurrences différentes de la protéine EGFR ; entre le site c et le n des occurrences de la protéine EGFR (il sera bientôt expliqué que la carte de contacts ne précise pas si ce doit être entre deux occurrences différentes de la protéine EGFR) ; entre le site Y48 d'une occurrence de la protéine EGFR et le site pi d'une occurrence de la protéine ShC ; entre le site a d'une occurrence de la protéine Grb2 et le site Y68 d'une occurrence de la protéine EGFR ; entre le site a d'une occurrence de la protéine Grb2 et le site Y7 d'une occurrence de la protéine ShC (il y a donc conflit entre ces deux liaisons potentielles) ; enfin entre le site b d'une occurrence de la protéine Grb2 et le site d d'une occurrence de la protéine Sos.

1.2.2 Configurations d'espèces biochimiques

Les modèles Kappa décrivent l'évolution d'une soupe d*espèces biochimiques*. Une configuration d'espèces biochimiques est formée de plusieurs occurrences de protéines. Chaque occurrence d'une protéine est associée à un ensemble de sites d'interactions. Chaque site peut éventuellement porter un état d'activation, mais un seul. De ce fait, si un site peut être activé de deux manières différentes, avec un état de phosphorylation et un état de méthylation par exemple, ou si un site peut être doublement activé, doublement phosphorylé par exemple, il est important de définir une pastille différente pour toutes les combinaisons potentielles d'états de ce site. Enfin, chaque site doit être soit libre, soit lié à exactement un autre site. Contrairement à la carte de contacts, un site ne peut pas être lié à lui-même dans une configuration d'espèces biochimiques. De plus, un site ne peut pas être lié simultanément à deux sites. Une configuration d'espèces biochimiques forme un graphe connexe, ce qui signifie qu'il est

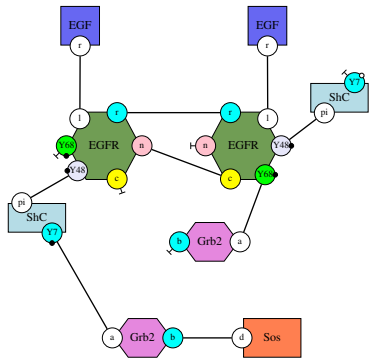


FIGURE 1.3 – *Une configuration d'une espèce biochimique. Elle contient plusieurs occurrences de protéines. Chaque occurrence documente l'ensemble de ses sites d'interactions. Les sites qui peuvent porter un état interne en ont un. Par ailleurs, les sites sont soit libres, soit liés deux à deux.*

possible de passer de n'importe quelle occurrence de protéines à n'importe quelle autre, en suivant zéro, un ou plusieurs liens.

Exemple 1.2.2. *En figure 1.3 est donné l'exemple d'une configuration d'espèces biochimiques. Celle-ci est formée de deux occurrences du ligand EGF, de deux occurrences du récepteur membranaire EGFR, de deux occurrences de la protéine d'échafaudage ShC, de deux occurrences de la protéine de transport Grb2 et d'une occurrence de la protéine Sos.*

La signature d'un modèle restreint l'ensemble des configurations des espèces biochimiques de ce modèle. Toutes les configurations des espèces biochimiques qui sont correctes du point de vue de la syntaxe ne sont ainsi pas adéquates. Ce rôle est assuré par la carte de contacts, qui d'une part, donne la liste de tous les sites d'interactions de chaque sorte de protéines en indiquant lesquels peuvent porter un état de liaison et un état d'activation et d'autre part, résume l'ensemble des états potentiels de ces sites. Plus précisément, toute occurrence de protéines dans la configuration d'une espèce biochimique doit mentionner les mêmes sites que le nœud correspondant dans la carte de contacts. De plus, un site dont le site correspondant dans la carte de contacts admet au moins un état d'activation doit nécessairement avoir un état d'activation. Il en est de même pour l'état de liaison. Ces contraintes assurent que l'état de chaque occurrence de protéines d'une configuration d'espèces biochimiques est entièrement défini. Trois contraintes supplémentaires assurent que l'état des sites est conforme à la carte de contacts : premièrement, un site ne peut porter un état d'activation que si le site correspondant dans la carte de contacts porte

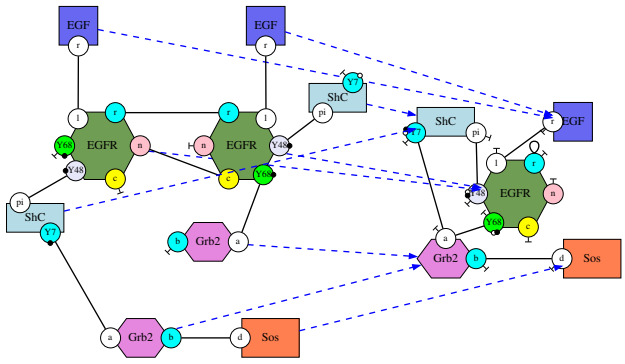


FIGURE 1.4 – *L'unique projection entre la configuration de l'espèce biochimique de la figure 1.3 et la carte de contacts de la figure 1.2. Cette projection est obtenue en associant chaque occurrence de protéines de la configuration de l'espèce biochimique à l'unique sorte de protéines correspondante dans la carte de contacts.*

également cet état d'activation ; deuxièmement, un site ne peut être libre que si le site correspondant dans la carte de contacts peut l'être lui-aussi ; troisièmement, deux sites ne peuvent être liés que si les deux sites correspondants le sont également dans la carte de contacts. Ces trois dernières contraintes peuvent se formaliser par le fait que chaque configuration d'une espèce biochimique se projette sur la carte de contacts : ainsi la fonction qui associe à chaque nœud d'une configuration d'espèces biochimiques l'unique nœud de la même sorte dans la carte de contacts doit être un *homomorphisme*. En d'autres termes, la carte de contacts peut être vue comme un repliage de toutes les configurations d'espèces biochimiques du modèle et chaque nœud de la carte de contacts résume toutes les configurations possibles des protéines du type correspondant.

Exemple 1.2.3. *En figure 1.4 est représentée la projection entre la configuration de l'espèce biochimique dessinée dans la figure 1.3 et la carte de contacts donnée en figure 1.2. Cette projection montre que cette configuration d'espèces biochimiques est compatible avec cette carte de contacts.*

1.2.3 Motifs

L'évolution des configurations d'espèces biochimiques est décrite par des règles de réécriture. Celles-ci définissent à la fois les conditions qui doivent être réalisées pour qu'une interaction puisse avoir lieu et les effets potentiels de cette interaction. Avant d'expliquer ce que sont ces règles de réécriture, il est nécessaire d'expliquer la notion de motifs. Celle-ci permet de spécifier sous quelles conditions une interaction peut avoir lieu.

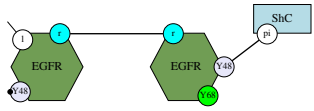


FIGURE 1.5 – *Un motif connexe. Il contient plusieurs occurrences de protéines. Chaque occurrence de protéines documente un sous ensemble de ses sites d'interactions. Chaque site peut éventuellement porter un état interne et éventuellement un état de liaison (en conformité avec la signature du modèle, donnée en figure 1.2). Comme état de liaison, un site peut être libre, lié sans que le site partenaire ne soit précisé ou être lié à un autre site.*

Nous nous concentrons sur les motifs connexes. Des motifs plus élaborés peuvent être obtenus en juxtaposant plusieurs motifs connexes. Un motif connexe est une portion contigüe dans la configuration d'une espèce biochimique. De ce fait, il peut comporter zéro, une ou plusieurs occurrences de chaque sorte de protéines. Chaque occurrence de protéines est associée à un ensemble de sites d'interactions. Chaque site peut éventuellement porter un état d'activation. Enfin chaque site peut être libre, lié sans que le site auquel il est lié ne soit précisé ou lié exactement à un autre site. L'état de liaison d'un site peut également ne pas être spécifié.

Exemple 1.2.4. *En figure 1.5 est donné un exemple de motif connexe. Ce motif est formé de deux occurrences du récepteur membranaire EGFR et d'une occurrence de la protéine d'échafaudage ShC.*

Comme c'était le cas pour les configurations d'espèces biochimiques, la carte de contacts contraint les motifs que l'on peut écrire dans un modèle. Ainsi, une occurrence de protéines dans un motif ne peut comporter que des sites d'interactions qui sont associés à cette sorte de protéines dans la carte de contacts. Un site ne peut porter un état d'activation que si le site correspondant dans la carte de contacts admet cet état d'activation. Un site ne peut être libre que si le site correspondant peut être libre dans la carte de contacts. Un site ne peut être lié sans préciser à quel site que si le site correspondant est lié à au moins un site dans la carte de contacts. Enfin, deux sites ne peuvent être liés ensemble que si les deux sites correspondants sont liés ensemble dans la carte de contact. En d'autres termes, comme c'était le cas pour les configurations d'espèces biochimiques, il doit être possible de projeter le motif sur la carte de contacts.

1.2.4 Plongements entre motifs

Un motif peut contenir plus ou moins d'information. En effet, il est possible d'ajouter des sites dans une occurrence de protéines qui ne mentionne pas tous ses sites. Par ailleurs, il est possible d'ajouter un état de liaison et/ou un état interne à un site qui en manque. Il est possible de préciser à quel site un site est lié quand le partenaire de celui-ci n'est pas précisé. Il est même possible de lier un site au site d'une nouvelle occurrence de protéines. Nous dirons alors que le premier motif apparait dans le second ou encore que le second motif contient une occurrence du premier. Dans ce cas, la relation entre les occurrences de protéines du motif initial et celle du motif ainsi obtenu est formalisée par un plongement. Un plongement d'un motif vers un autre motif est une fonction qui envoie chaque occurrence de protéines du premier motif vers une occurrence de protéines du second tout en préservant la structure des graphes à sites, c'est à dire les sortes de protéines, les sites qui sont mentionnés, les états internes et les états de liaisons qui sont documentés.

Il est intéressant de remarquer que les configurations d'espèces biochimiques sont des motifs connexes particuliers. Dans ces derniers, chaque occurrence de protéines décrit tous ses sites, avec un état interne et un état de liaison quand ils en ont un. Il n'est donc pas possible d'ajouter d'information dans la configuration d'une espèce biochimique. Une configuration d'espèces biochimiques ne peut se plonger dans aucun autre motif connexe.

Exemple 1.2.5. *Un exemple de plongements d'un motif dans une configuration d'espèces biochimiques est donné en figure 1.6. Ce plongement est uniquement caractérisé par l'image de l'occurrence de la protéine d'échafaudage du motif. Celle-ci est associée à l'occurrence de la protéine d'échafaudage de la configuration d'espèces biochimiques dont le site Y7 est libre. L'image des autres occurrences de protéines du motif, par ces plongements, se retrouve de proche en proche en suivant les liaisons entre les sites des occurrences de protéines.*

Il est important de remarquer qu'un plongement d'un motif connexe vers un autre motif est entièrement caractérisé par l'image d'une occurrence de protéines. Pour avoir les autres associations, il suffit de suivre les liens et d'utiliser le fait qu'ils sont nécessairement préservés par le plongement. Cette propriété facilite la recherche d'occurrences de motifs dans les autres. Les graphes Kappa sont dits rigides [30, 67].

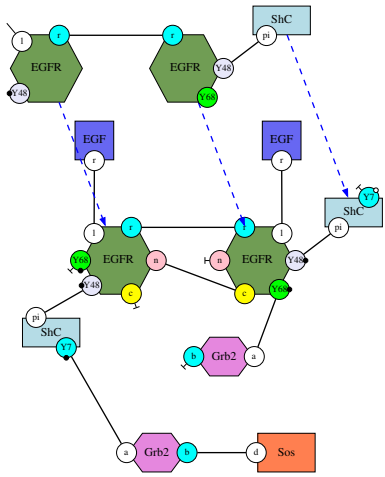


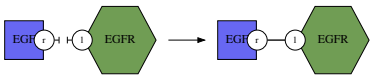
FIGURE 1.6 – *Un plongement entre le motif donné dans la figure 1.5 et la configuration d'espèces biochimiques donnée dans la figure 1.3. L'occurrence de la protéine d'échafaudage est associée à l'occurrence de la protéine d'échafaudage dont le site Y7 est libre.*

1.2.5 Règles d'interaction

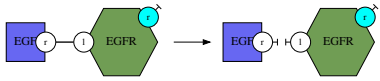
Les motifs permettent de spécifier l'évolution potentielle de l'état des systèmes modélisés en Kappa, grâce à des règles de réécriture. Afin de simplifier la présentation, seul un fragment du langage Kappa est présenté. En particulier, les règles de réécriture qui sont introduites dans cette section n'engendrent pas d'effets de bord. Un effet de bord est une transformation à l'extérieur du membre gauche des règles. Les effets de bords peuvent être dus à des sites libérés sans préciser à quels sites ils sont liés ou à des occurrences de protéines dégradées. Ces constructions n'ont pas été considérées afin de simplifier la présentation et de présenter tous les différents concepts de la syntaxe et de la sémantique de Kappa sous forme graphique.

Les configurations d'espèces biochimiques peuvent se transformer en appliquant des règles d'interactions. Une règle d'interaction est définie par une paire de motifs, qui contiennent exactement les mêmes sortes de protéines. Le premier motif spécifie quelles conditions locales doivent être réalisées pour permettre à l'interaction de se produire. La différence entre ces deux motifs décrit quelle transformation résulte de cette interaction. Aussi le second motif d'une règle doit pouvoir être obtenu à partir du premier en changeant uniquement l'état interne et/ou de liaison de certains sites d'interactions.

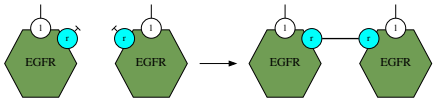
Exemple 1.2.6. *Des exemples de règles d'interactions sont données en figure*



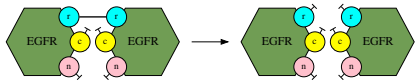
(a) *Activation d'une occurrence du récepteur.*



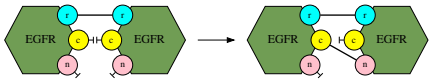
(b) *Désactivation d'une occurrence du récepteur.*



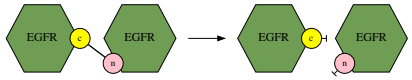
(c) *Liaison symétrique.*



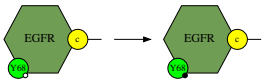
(d) *Déliaison du lien symétrique.*



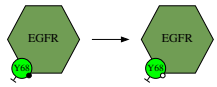
(e) *Liaison asymétrique.*



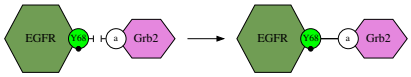
(f) *Déliaison du lien asymétrique.*



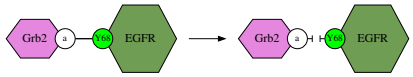
(g) *Phosphorylation d'une occurrence du récepteur.*



(h) *Déphosphorylation d'une occurrence du récepteur.*



(i) *Recrutement d'une occurrence de la protéine transporteur.*



(j) *Déliaison d'une occurrence de la protéine transporteur.*

(k) *Liaison d'une occurrence de la protéine transporteur à une occurrence de la protéine cible.*

(l) *Déliaison d'une occurrence de la protéine transporteur d'une occurrence de la protéine cible.*

FIGURE 1.7 – *Règles d'interactions impliquées dans le recrutement d'une occurrence de la protéine cible par la voie de signalisation courte (sans passer par la protéine d'échafaudage).*

1.7. Celles-ci décrivent les interactions qui sont impliquées dans le recrutement des occurrences de la protéine cible par les occurrences du récepteur membranaire par leur site Y68, dans le modèle des premières étapes de l'acquisition du facteur de croissance de l'épiderme. Le recrutement par le site Y48 implique des règles d'interactions similaires, qui ne seront donc pas détaillées. La colonne de gauche décrit les interactions qui font progresser le recrutement d'une occurrence de la protéine cible. Chacune de ces interactions est réversible. Toutefois, les interactions inverses s'effectuent sous des conditions d'application différentes. Ces interactions sont décrites dans la colonne de droite.

Dans le langage complet, il est possible de détruire un lien entre deux occurrences de protéines en ne spécifiant qu'un seul des deux sites de

liaisons. De plus, une règle peut également détruire des occurrences de protéines. Ces constructions peuvent induire des effets de bord, puisqu'appliquer de telles interactions est susceptible de libérer des sites qui ne sont pas décrits dans les membres gauches des règles correspondantes. Par ailleurs, le langage complet permet aussi de synthétiser de nouvelles occurrences de protéines.

1.2.6 Réactions induites par une règle d'interaction

Comme signalé précédemment, le membre gauche d'une règle d'interactions spécifie dans quel contexte cette interaction peut avoir lieu. Il est alors possible d'ajouter des contraintes sur les conditions d'application d'une règle en raffinant les motifs qui apparaissent dans les membres gauches et droits des règles exactement de la même manière. Une règle d'interactions qui ne peut plus être raffinée (sans ajouter de nouvelles composantes connexes) est alors appelée une règle-réaction [52].

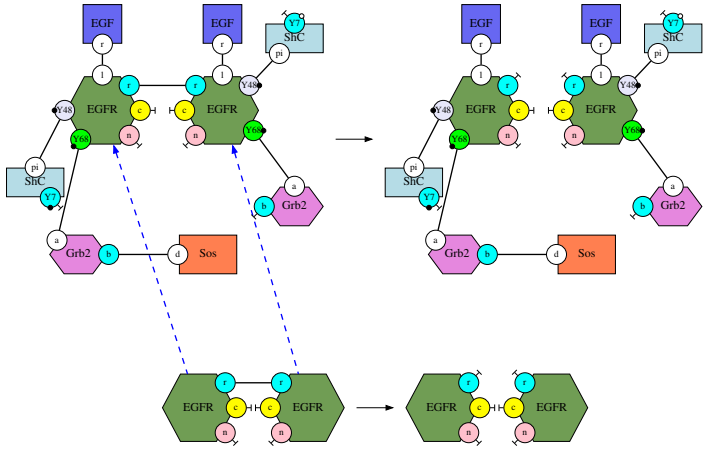


FIGURE 1.8 – *Raffinement d'une règle d'interactions en une règles-réaction. La règle-réaction est obtenue en ajoutant dans le membre gauche et dans le membre droit de la règle d'interactions exactement la même information sur le contexte d'application de la règle.*

Exemple 1.2.7. *En figure 1.8 est montré un exemple de raffinement pour la règle d'interactions qui permet de casser, en l'absence de lien asymétrique, le lien symétrique entre deux occurrences du récepteur membranaire (voir en figure 1.7d). Le résultat est une règle-réaction.*

1.2.7 Réseaux de réactions sous-jacents

Un ensemble de règles peut alors être traduit en un ensemble – éventuellement infini – de règles-réactions en remplaçant chaque règle d'interactions par l'ensemble des règles-réactions qui peuvent être obtenues comme raffinement de ces règles. Ensuite, quitte à donner un nom à chaque configuration d'espèces biochimiques qui peut intervenir dans les règles-réactions ainsi obtenues, ces règles-réactions peuvent être assimilée à un réseau de réactions (éventuellement infini), dans lequel chaque réaction est spécifiée par une liste de réactifs et une liste de produits parmi un ensemble d'espèces biochimiques représentées uniquement par des noms (en passant sous silence leurs structures biochimiques). Ce réseau de réactions est défini de manière unique modulo le choix des noms associés aux espèces biochimiques.

Le choix d'une sémantique en terme de réseaux réactionnels a été fait pour simplifier la présentation. C'était ainsi que le langage BNGL avait été implanté initialement [7]. Une telle sémantique est toutefois assez peu utile en pratique, car un modèle Kappa engendre en général un trop grand nombre de réactions. Par contre, la sémantique de Kappa peut être formalisée directement, soit sous forme d'une algèbre de processus [32, 44], soit dans un cadre catégorique [28, 42]. La première méthode est plus opérationnelle alors que la seconde abstrait au contraire beaucoup de détails.

La simulation d'un modèle Kappa opère directement par réécriture du graphe qui représente l'état du système, sans avoir à considérer le réseau de réactions sous-jacent [31, 11].

1.3 Analyse des motifs accessibles

Si la carte de contacts (e.g. voir en figure 1.2 à la page 9) donne un aperçu rapide de toutes les interactions potentielles entre les différents sites des occurrences des protéines dans un modèle, elle n'est en général pas suffisante pour décrire précisément la structure de la configuration de ses espèces biochimiques. En effet, l'état des différents sites d'interactions de la configuration d'une espèce biochimique est souvent contraint par des invariants structurels. Par exemple, dans le modèle des premières étapes de l'acquisition du facteur de croissance de l'épiderme, les sites *Y48* et *Y68* des occurrences du récepteur membranaire, ainsi que le site *Y7* des occurrences de la protéine d'échafaudage, ne peuvent être liés à un autre site sans être phosphorylés (à moins que ce soit le cas dans l'état initial). Par ailleurs, lorsque les deux sites *r* et *c* d'une occurrence du récepteur sont

liés simultanément, ils sont nécessairement liés respectivement au site *r* et au site *n* d'une même occurrence du récepteur (ce qui forme une double liaison). Un autre exemple concerne les modèles avec des compartiments, comme, par exemple, une cellule dont on distingue le noyau du cytoplasme. La localisation de chaque occurrence de protéines peut alors être spécifiée comme l'état d'activation d'un site fictif. Dans de tels modèles, toutes les occurrences de protéines de la même occurrence d'une espèce biochimique sont en général localisées dans un même compartiment, ce qui se traduit par la contrainte que le site fictif de deux occurrences de protéines liées entre elles doit toujours être dans le même état. Dans certains cas, il est toutefois possible d'avoir des espèces biochimiques transmembranaires avec des portions localisées dans des compartiments voisins, c'est à dire de part et d'autre d'une membrane.

Dans cette section est décrite une analyse statique qui permet de détecter automatiquement ces contraintes. Le but est de vérifier que les propriétés auxquelles peut s'attendre le modélisateur sont bien vérifiées ou bien de détecter certaines erreurs de modélisation. En particulier, cette analyse permet de trouver des *règles mortes*. Ce sont des règles qui ne peuvent jamais s'appliquer dans un modèle, car les contraintes qui sont exprimées dans leurs membres gauches ne sont pas réalisables. C'est souvent la conséquence d'erreurs typographiques (par exemple, quand une même sorte de protéines est désignée par deux noms différents dans l'encodage d'un modèle), d'un état initial incomplet, d'interactions manquantes dans le modèle (par exemple, quand l'activation d'un site n'est pas décrite, alors qu'elle est nécessaire pour la suite de la cascade d'interactions) ou de conditions causales plus complexes qu'il faut alors élucider.

Cette analyse est implantée dans l'analyseur statique KaSa [10] et intégrée dans la plate-forme de modélisation en ligne dédiée au langage Kappa [13]. Ceci permet d'assister le modélisateur pendant l'écriture du modèle en lui fournissant les contraintes structurelles qui sont vérifiées par les configurations des espèces biochimiques et en l'avertissant de la présence de règles mortes, après chaque ajout ou modification d'une règle d'interactions.

1.3.1 Accessibilité dans un réseau réactionnel

La première étape consiste à définir l'ensemble des états accessibles dans un modèle Kappa. Comme nous l'avons vu dans la section 1.2.7 page 18, un modèle Kappa induit un réseau réactionnel, ce qui permet de définir directement l'ensemble des états accessibles d'un modèle Kappa sans recourir à des constructions compliquées.

Soit un réseau réactionnel, c'est à dire un ensemble d'espèces biochimiques $\mathcal{S}$ et un ensemble de réactions $\mathcal{R}$. Chaque réaction est donnée par deux listes d'espèces biochimiques : ses réactifs et ses produits. Ce réseau induit un système de transitions dans lequel l'*état* du réseau est défini comme un certain nombre (éventuellement nul) d'occurrences de chacune des espèces biochimiques – c'est à dire une fonction de l'ensemble $\mathcal{S}$ vers l'ensemble $\mathbb{N}$ des entiers naturels — et les *transitions* permettent de sauter d'un état à un autre en consommant les réactifs d'une réaction et en ajoutant les produits de cette même réaction (en tenant compte de leur multiplicité). Une transition n'est possible que si l'état courant du système contient tous les réactifs qui sont nécessaires à la réaction (en tenant compte, une nouvelle fois, de leur multiplicité). Une transition d'un état q vers un autre état q' est alors notée $q \to q'$.

Étant donné un ensemble d'états initiaux potentiels, $\mathcal{I} \subseteq \mathcal{S}^{\mathbb{N}}$, nous définissons l'ensemble des états accessibles comme étant ceux susceptibles d'être atteints à partir d'un état initial (de l'ensemble $\mathcal{I}$) en appliquant un nombre arbitraire (éventuellement nul) de transitions. Cet ensemble peut se définir comme le plus petit point-fixe de la fonction suivante :

$$\mathbb{F} : \begin{cases} \wp(\mathcal{S}^{\mathbb{N}}) & \to \wp(\mathcal{S}^{\mathbb{N}}) \\ X & \to \mathcal{I} \cup \{q' \mid \exists q \in X,\ q \to q'\}. \end{cases}$$

Il faut noter que la fonction $\mathbb{F}$ est croissante, ce qui signifie que si X_1 et X_2 sont deux ensembles d'états tels que l'ensemble X_1 soit un sous-ensemble de l'ensemble X_2, alors l'ensemble $\mathbb{F}(X_1)$ est nécessairement un sous-ensemble de l'ensemble $\mathbb{F}(X_2)$ lui-aussi. Comme, de plus, cette fonction est définie sur l'ensemble des parties d'un ensemble, le *théorème de Tarski* [71] assure que la fonction $\mathbb{F}$ admet un point fixe, plus petit que tout autre point fixe de $\mathbb{F}$. Ce plus-petit point fixe, que l'on note *lfp* $\mathbb{F}$, est en fait l'ensemble des états accessibles.

Malheureusement, le calcul de ce plus petit point fixe peut être coûteux, voire ne pas terminer. Ceci motive la construction d'abstractions pour calculer un sur-ensemble des états accessibles en un temps de calcul acceptable.

1.3.2 Abstraction d'un ensemble d'états

Lorsqu'un réseau est induit par un modèle Kappa, la structure biochimique associée aux espèces de ce réseau peut être utilisée pour construire une abstraction. Une possibilité consiste à choisir un ensemble de motifs connexes afin d'abstraire les ensembles d'états par le sous-ensemble parmi

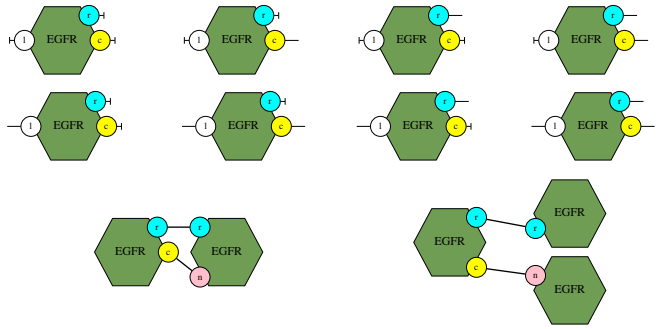


FIGURE 1.9 – *Un ensemble de motifs d'intérêt pour l'analyse des configurations accessibles des espèces biochimiques dans le modèle des premières interactions qui interviennent dans l'acquisition du facteur de croissance de l'épiderme.*

ces motifs de ceux qui apparaissent au moins une fois dans au moins un état de cet ensemble. Le choix des motifs connexes considérés est important : il définit le compromis entre l'expressivité de l'abstraction, c'est à dire son niveau d'approximation, et sa complexité, c'est à dire le coût pour effectuer des calculs à ce niveau d'abstraction.

Exemple 1.3.1. *Un exemple de motifs d'intérêt pour le modèle des premières interactions de l'acquisition du facteur de croissance de l'épiderme est donné en figure 1.9. Les huit premiers motifs se concentrent sur l'analyse des relations potentielles entre l'état des sites l, r et c dans les occurrences du récepteur membranaire. Ils correspondent à toutes les combinaisons syntaxiquement possibles pour l'état de liaison de ces* 3 *sites. Ce sont des* vues locales *(ou plus précisément des sous-vues locales) [32]. Elles permettent d'abstraire un ensemble de configurations d'espèces biochimiques par l'ensemble de toutes les configurations potentielles de toutes ses occurrences de protéines, vues indépendamment les unes des autres. Ceci revient à garder uniquement l'information à propos de l'état de liaison et l'état d'activation de chaque site dans chaque occurrence de protéines tout en passant sous silence à quel site chaque site lié l'est.*

La formation de dimères dans ce modèle fait intervenir des doubles liaisons. Il est légitime de se demander s'il est possible de former des chaines comportant successivement au moins trois occurrences du récepteur membranaire. C'est le but des deux derniers motifs de l'ensemble. Ils permettent de distinguer le cas d'une double liaison entre deux occurrences du récepteur de celui de trois occurrences du récepteur liées consécutivement, en s'interrogeant pour chaque occurrence du récepteur membranaire dont les sites r et c sont liés, si elle peut être liée à une même occurrence du récepteur ou si elle peut être liée à deux occurrences différentes. En toute rigueur, pour s'assurer qu'une chaine d'au moins trois occurrences du

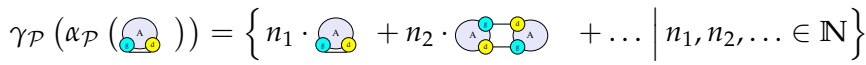
récepteur ne peut pas se former, il faut également considérer des motifs d'intérêt similaires pour la paire de sites r et n et la paire de sites c et n.

Plus précisément, l'abstraction est paramétrée par le choix d'un ensemble $\mathcal{P}$ de motifs connexes. L'ensemble $\mathcal{P}$ regroupe des motifs d'intérêt, ainsi que des motifs qui seront utilisés de manière intermédiaire dans la preuve que certains de ces motifs d'intérêt sont inaccessibles. Une sous-partie de l'ensemble $\mathcal{P}$ est appelée une propriété abstraite. Chaque propriété abstraite représente un ensemble d'états concrets : un état concret q sera dît compatible avec une propriété abstraite $X^\sharp$ si et seulement si aucun motif qui est dans l'ensemble $\mathcal{P}$ sans être dans l'ensemble $X^\sharp$ n'apparait dans la configuration d'une espèce biochimique présente dans l'état q. L'ensemble de tous les états concrets compatibles avec la propriété abstraite $X^\sharp$ est alors noté $\gamma_\mathcal{P}(X^\sharp)$. Qui peut le plus, peut le moins : plus nombreux sont les motifs autorisés, plus nombreuxs sont les configurations d'espèces biochimiques compatibles. La fonction $\gamma_\mathcal{P}$ est donc croissante. Elle permet de définir formellement la notion d'abstraction d'un ensemble d'état : une propriété abstraite $X^\sharp$ sera dite être une abstraction d'un ensemble d'état X si et seulement si l'ensemble X est inclus dans l'ensemble $\gamma_\mathcal{P}(X^\sharp)$. La fonction $\gamma_\mathcal{P}$ est couramment appelée la *fonction de concrétisation*. De plus l'image d'une propriété abstraite par cette fonction, est appelée sa *concrétisation*.

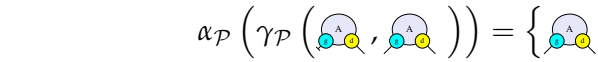
Réciproquement, étant donné un ensemble d'états X, l'ensemble des éléments de l'ensemble $\mathcal{P}$ qui apparaissent dans au moins une configuration d'espèces biochimiques d'un état élément de l'ensemble X sera noté $\alpha_\mathcal{P}(X)$. La fonction $\alpha_\mathcal{P}(X)$ est croissante également. La propriété abstraite $\alpha_\mathcal{P}(X)$ est en fait la *meilleure approximation* de l'ensemble d'états X, ce qui signifie que d'une part c'est une abstraction de l'ensemble X (i.e. $X \subseteq \gamma_\mathcal{P}(\alpha_\mathcal{P}(X))$) et que d'autre part c'est un sous-ensemble de toute autre abstraction de X (i.e. pour tout sous-ensemble Y de l'ensemble $\mathcal{P}$ tel que $X \subseteq \gamma_\mathcal{P}(Y)$, l'inclusion $\alpha_\mathcal{P}(X) \subseteq Y$ est vérifiée). La paire de fonctions $(\alpha_\mathcal{P}, \gamma_\mathcal{P})$ est alors appelée une *correspondance de Galois* [25, 23].

Exemple 1.3.2. *En figure 1.10 est introduit un exemple jouet pour mieux comprendre le comportement des fonctions d'abstraction et de concrétisation. La signature de ce modèle peut être consultée en figure 1.10a. Il existe une seule sorte de protéines, qui est appelée A. Cette protéine est munie de deux sites g et d (pour gauche et droite). La carte de contacts spécifie que chaque site peut être libre et qu'un site g peut être lié à un site d d'une même ou d'une autre occurrence de la protéine A. L'abstraction induite par l'ensemble des motifs d'intérêt donné en figure 1.10b repose sur les vues locales des occurrences de cette protéine. Elle permet de se poser la question de l'existence ou non, d'une relation entre l'état de liaison des sites g et d dans chaque occurrence de la protéine A.*

(a) Carte de contacts. (b) Motifs d'intérêt.



(c) Exemple d'application de la fonction $\gamma_{\mathcal{P}} \circ \alpha_{\mathcal{P}}$.



(d) Exemple d'application de la fonction $\alpha_{\mathcal{P}} \circ \gamma_{\mathcal{P}}$.

FIGURE 1.10 – *Un exemple jouet pour mieux comprendre le comportement des fonctions d'abstraction et de concrétisation. En 1.10a, la signature du modèle : une seule sorte de protéines, A, avec deux sites pouvant être libres ou liés à l'autre site de la même ou d'une autre occurrence de la protéine A. En 1.10b, le domaine abstrait est formé des vues locales de l'unique sorte de protéines : toutes les configurations pour les occurrences de la protéine A sont considérées selon que chaque site soit libre ou lié. Un exemple d'application de la composée de fonctions $\gamma_{\mathcal{P}} \circ \alpha_{\mathcal{P}}$ est montré en 1.10c. Celui-ci montre que l'abstraction ne permet pas de distinguer des ensembles d'anneaux d'occurrences de la protéine A et ce quels que soient leurs tailles et leurs nombres. En 1.10d donne un exemple d'application de la composée de fonctions $\alpha_{\mathcal{P}} \circ \gamma_{\mathcal{P}}$. Cette fonction calcule que la vue locale avec le site g libre et le site d lié ne peut pas apparaître dans une espèce biochimique qui ne contiendrait pas la vue avec le site g lié et le site d libre.*

Les fonctions $\alpha_{\mathcal{P}}$ et $\gamma_{\mathcal{P}}$ se composent dans les deux sens. Ces compositions sont révélatrices des traits principaux du choix de l'abstraction. La composée $\gamma_{\mathcal{P}} \circ \alpha_{\mathcal{P}}$ caractérise le niveau d'approximation. En effet, pour tout ensemble d'états X, $\gamma_{\mathcal{P}}(\alpha_{\mathcal{P}}(X))$ est le plus grand ensemble d'état qui a la même meilleure approximation que X. Il est impossible ainsi de distinguer ces deux ensembles en terme de propriétés abstraites. En revanche, la composée $\alpha_{\mathcal{P}} \circ \gamma_{\mathcal{P}}$ témoigne d'une certaine combinatoire dans le domaine abstrait. Elle associe à chaque propriété abstraite, la plus petite propriété abstraite qui est satisfaite par le même ensemble d'états concrets. Appliquer cette composée permet donc de raffiner une propriété abstraite, par déduction, et ce sans perdre le moindre état concret.

Exemple 1.3.3. *Appliquée à l'ensemble formé d'un seul état composé uniquement d'un anneau de taille* 1*, la composée de fonctions $\gamma_{\mathcal{P}} \circ \alpha_{\mathcal{P}}$ donne l'ensemble des états formés uniquement d'anneaux d'occurrences de la protéine A. En effet, la meilleure approximation d'un anneau de taille* 1*, est l'ensemble de vues locales*

composé uniquement de la vue dont les deux sites sont liés. Or la concrétisation de cet ensemble de vues locales est l'ensemble de tous les états formés uniquement d'anneaux. Ainsi le niveau d'abstraction ne permet de distinguer, ni le nombre d'occurrences, ni la taille des anneaux d'occurrences de la protéine A.

Exemple 1.3.4. *Appliquée à l'ensemble formé exactement des deux vues locales, la première avec le site g libre et le site d lié, la seconde avec les deux sites liés, la composée de fonctions $\alpha_{\mathcal{P}} \circ \gamma_{\mathcal{P}}$ retourne l'ensemble formé d'une seule vue locale, celle avec les deux sites liés. En effet, la première vue ne peut apparaître dans un état sans que celui-ci ne contienne une occurrence de la vue locale avec le site d libre et le site g lié. De ce fait, elle ne peut apparaître dans aucun état de la concrétisation de l'ensemble formé par ces deux vues locales et n'est donc pas un élément de la meilleure approximation de l'ensemble de ces états. Ainsi, un état abstrait peut contenir des motifs d'intérêt, qui ne peuvent apparaître dans aucune configuration d'espèces biochimiques sans contenir des occurrences de motifs d'intérêt interdits. Retirer ces motifs ne change pas l'ensemble des états concrets qui satisfont la propriété abstraite, mais cette étape peut requérir un temps de calcul substantiel.*

1.3.3 Transferts de point-fixes

Le plus petit point fixe qui définit l'ensemble des configurations d'espèces biochimiques accessibles dans un réseau réactionnel, pour un état initial donné, peut se calculer au niveau des propriétés abstraites grâce à la correspondance de Galois ($\alpha_{\mathcal{P}}$,$\gamma_{\mathcal{P}}$).

Pour cela, il faut tout d'abord construire la *contre-partie abstraite* de la fonction $\mathbb{F}$, qui agira, non pas sur des ensembles d'états concrets, mais directement sur les propriétés abstraites. Cette contre-partie abstraite se définit de manière systématique : il suffit, pour chaque propriété abstraite $X^\sharp$, de considérer l'ensemble des états concrets $\gamma_{\mathcal{P}}(X^\sharp)$ qui vérifient la propriété $X^\sharp$, puis d'appliquer la fonction $\mathbb{F}$ à cet ensemble et enfin d'appliquer à ce résultat la fonction $\alpha_{\mathcal{P}}$ pour en calculer la meilleure approximation. C'est même la manière correcte la plus précise de procéder : la fonction $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$ est, en effet, la meilleure contre-partie abstraite de la fonction $\mathbb{F}$ [26]. Elle permet de déléguer le calcul des états accessibles au domaine abstrait en contre-partie d'une perte éventuelle de précision. Pour ce faire, il suffit de remarquer que la fonction $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$ est croissante (comme composée de fonctions croissantes) et définie sur l'ensemble des parties d'un ensemble. Elle admet donc un plus petit point fixe qui sera noté $lfp\ (\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}})$. L'inclusion suivante : $lfp\ \mathbb{F} \subseteq \gamma_{\mathcal{P}}(lfp\ (\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}))$ se prouve alors par induction [26]. Autrement dit le plus petit point fixe de la fonction $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$ est une abstraction de l'ensemble des états accessibles

du modèle considéré. C'est à dire que la propriété abstraite $lfp\ (\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}})$ est satisfaite par chaque état accessible du modèle.

Le calcul des itérations de la fonction $[\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}]$ peut prendre beaucoup de temps. Il est possible d'ajuster le compromis entre précision et temps de calcul en remplaçant celle-ci par une fonction moins précise. En effet, pour toute fonction croissante $\mathbb{F}^\sharp$ telle que $[\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}](Y) \subseteq \mathbb{F}^\sharp(Y)$ pour tout ensemble de motifs $Y \subseteq \mathcal{P}$, l'inclusion $lfp\ \mathbb{F} \subseteq \gamma_{\mathcal{P}}(lfp\ (\mathbb{F}^\sharp))$ est également satisfaite [26].

Une telle fonction $\mathbb{F}^\sharp$ peut être dérivée à la main. Pour cela, il faut d'abord donner une définition plus explicite de la fonction $[\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}]$. Appliquée à un sous-ensemble $Y \subseteq \mathcal{P}$ de motifs d'intérêt, cette fonction ajoute l'ensemble des nouveaux motifs d'intérêt qui peuvent apparaître dans un état accessible en une étape de réécriture à partir d'un état qui ne contient aucun motif de l'ensemble $\mathcal{P}$ qui ne serait pas dans l'ensemble de motifs Y. Or, une telle étape de réécriture est nécessairement induite par une règle-réaction, elle-même induite par une règle du modèle. Chaque nouveau motif P doit donc apparaître dans le membre droit d'une règle-réaction et l'ensemble d'états singleton formé du membre gauche de cette règle-réaction ne doit contenir aucune occurrence de motifs de l'ensemble $\mathcal{P} \setminus Y$. Dans cette règle-réaction, l'occurrence du motif P dont il est question et l'image du membre droit de la règle sous-jacente ont nécessairement au moins une occurrence de protéines en commun (sinon le motif P apparaitrait également dans le membre gauche de la règle-réaction et ne serait dont pas un nouveau motif). Il est alors possible de fixer le motif P au membre droit de cette règle, en unifiant les occurrences de protéines du motif P et de la règle du modèle qui sont communes dans la règle-réaction. Ceci forme alors un raffinement du membre droit de la règle. Un raffinement de la règle peut alors être construit en ajoutant toute information présente dans le motif P qui n'est pas déjà présente dans le membre droit initial de la règle, dans le membre gauche de la règle. Le résultat est une spécialisation de la règle à la production du motif P à cette position particulière. Par construction, le membre gauche de la règle raffinée apparait dans un état dans la concrétisation de Y.

Ainsi, pour calculer les nouveaux motifs d'intérêt de l'ensemble $[\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}](Y)$, il suffit de calculer tous les *chevauchements* possibles entre un nouveau motif d'intérêt potentiel (dans $\mathcal{P} \setminus Y$) et un membre droit d'une règle du modèle. Chaque chevauchement induit un raffinement de la règle correspondante. Si le membre gauche de la règle raffinée apparait dans un état de l'ensemble $\gamma_{\mathcal{P}}(Y)$, alors ce motif appartient bien à l'ensemble $[\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}]Y$.

Lors du calcul de l'ensemble $[\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}](Y)$, l'étape la plus coûteuse

en temps de calcul est de vérifier que les membres gauches des règles raffinées peuvent apparaître dans un état de l'ensemble $\gamma_{\mathcal{P}}(Y)$. La section suivante a pour but de réduire ce coût moyennant une approximation supplémentaire.

1.3.4 Analyse par ensembles de motifs orthogonaux

Ajouter des hypothèses sur l'ensemble des motifs d'intérêt et simplifier le test de réalisabilité du membre gauche des raffinements de règles en le remplaçant par une condition nécessaire, mais pas toujours suffisante, permet de rendre ce calcul plus efficace au prix d'une perte de précision de l'analyse. Ceci permet de définir une approximation correcte de la fonction $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$.

Pour ce faire, l'ensemble des motifs d'intérêt peut être organisé sous la forme d'un ensemble fini d'ensembles finis de motifs orthogonaux [46]. Chaque *ensemble de motifs orthogonaux* est un arbre de décision raffinant progressivement un motif initial, dans le but de répondre à une question spécifique. Un ensemble de motifs orthogonaux est construit de manière à ce que toute occurrence du motif initial dans une configuration d'espèce biochimiques, puisse être complétée en exactement une occurrence d'un de ces raffinements. En conséquence, les raffinements du motif initial sont deux à deux incompatibles et ils recouvrent, en quelque sorte, tous les cas possibles pour le motif initial.

Le choix exact des ensembles de motifs orthogonaux repose sur une analyse préliminaire qui calcule, par inspection des règles du modèle, quelles questions intéressantes se posent. Trois catégories de questions sont considérées par défaut dans l'analyseur KaSa (mais il est possible de paramétrer l'analyse pour en désactiver une ou deux). La première infère des relations entre les états des différents sites de chaque type de protéines, cela correspond à l'analyse des vues locales [32]. La seconde permet de détecter des relations entre l'état des sites dans des occurrences de protéines qui partagent un lien [46] dans le but d'analyser le déplacement des occurrences des espèces biochimiques lorsque celui-ci est codé par des transformations de l'état d'activation de sites fictifs. L'analyse permet alors de vérifier si oui ou non deux occurrences de protéines sont toujours localisées dans le même compartiment quand elles sont liées entre elles. La troisième permet de détecter si une même occurrence de protéines peut être liée simultanément à deux occurrences différentes de protéines ou si une même occurrence de protéines peut être liée au moins doublement à une autre occurrence de protéines [46]. Une quatrième sorte d'ensembles de motifs orthogonaux est en cours d'implantation. Elle se concentre sur

la formation des espèces biochimiques cycliques : son but est de prouver l'absence de espèces biochimiques de taille non bornée [12].

Les ensembles finis de motifs orthogonaux peuvent être construits récursivement, en remplaçant un des motifs par plusieurs motifs le raffinant. Il suffit de choisir une information non spécifiée dans ce motif et de considérer tous les cas possibles pour cette information, d'où la représentation sous forme d'arbre de décision. L'ensemble de motifs orthogonaux est alors formé par les feuilles de cet arbre, alors que les nœuds de cet arbre représentent les motifs intermédiaires qui ont été remplacés par des motifs plus précis.

Exemple 1.3.5. *L'ensemble des motifs d'intérêt introduit en figure 1.9 est inclus dans la réunion de deux ensembles de motifs orthogonaux. En effet, l'ensemble des vues locales peut être obtenu, en partant d'une occurrence de la protéine EGFR sans aucun site, en se demandant successivement si le site l est libre ou non, si le site r est libre ou non et si le site c est libre ou non. Les deux derniers motifs d'intérêt sont obtenus en se demandant si un récepteur peut établir des liaisons doubles. Partant d'une occurrence de la protéine EGFR sans aucun site, il faut se demander si le site r est libre ou non, puis dans le cas où le site r est lié, si le site c est lié ou non, et enfin, dans le cas où le site c est également lié, si ces deux sites sont liés à une même occurrence de récepteur membranaire ou à deux occurrences différentes.*

Les différents ensembles de motifs orthogonaux collaborent au sein de l'analyse, qui effectue ainsi une induction mutuelle sur ces derniers. Ceci présente deux avantages par rapport à des analyses séparées ou en cascades (où chacune utiliserait le résultat des analyses précédentes). D'une part, il n'est pas nécessaire de définir quel ensemble de motifs orthogonaux doit être analysé avant quel autre. D'autre part, une induction mutuelle est strictement plus expressive. La collaboration entre les différents ensembles de motifs orthogonaux permet de prouver plus souvent que le membre gauche des règles raffinées n'est pas réalisable étant donné les motifs qui sont autorisés à un moment donné de l'analyse, et donc que la règle peut être ignorée à ce moment de l'analyse. Pour faire cette preuve, le raffinement d'une règle est construit de la manière habituelle. Il suffit ensuite de trouver une occurrence de protéines dans le membre gauche de la règle raffinée qui soit incompatible avec l'état actuel de l'analyse sur au moins un des ensembles de motifs orthogonaux pris en paramètre de l'analyse. Pour cela, la racine de l'ensemble de motifs orthogonaux doit être de la même sorte que l'occurrence de la protéine en question et l'information contextuelle de cette occurrence de protéines dans ce membre gauche de la règle raffinée ne doit être compatible avec aucun des motifs de cet ensemble de motifs

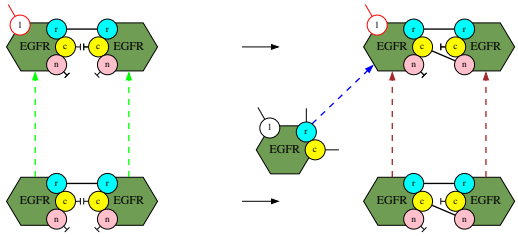


FIGURE 1.11 – *Procédure de décision approchée pour savoir que l'on ne peut pas prouver l'inaccessibilité d'un motif. Il suffit de s'assurer, pour chaque occurrence de protéines dans ce motif et chaque ensemble de motifs orthogonaux portant sur ce type de protéines si il contient un motif compatible déjà découvert par l'analyse.*

orthogonaux déjà déjà déclarés potentiellement accessibles par l'analyse. Dans le cas contraire, l'analyseur ne peut pas prouver que le motif est inaccessible. Le motif est alors considéré comme potentiellement accessible pour la suite de l'analyse. Il s'agit bien entendu d'une approximation.

Outre le fait de ne pas vérifier l'existence de la configuration d'une espèce biochimique qui pourrait compléter le collage obtenu entre les motifs connexes du membre gauche de la règle raffinée et les motifs déjà déclarés potentiellement accessibles par l'analyse, il est intéressant de remarquer que la procédure de décision approchée évite le calcul de tous les chevauchements entre les motifs d'intérêt non encore découverts par l'analyse, en se focalisant sur la racine de chaque ensemble de motifs orthogonaux. Ce sont les deux sources de pertes d'information dues à l'affaiblissement de la procédure de décision.

Exemple 1.3.6. *Le résultat de l'itération pour le modèle formé des règles qui avaient été décrites en figure 1.7 pour les ensembles de motifs orthogonaux qui avaient été introduits en figure 1.9, est donné en figure 1.12. Cette itération a été initialisée avec une quantité arbitraire d'occurrences de protéines de chaque sorte, mais avec tous leurs sites libres. Pour ce qui est des vues locales , seules* 4 *configurations sont possibles pour l'état des sites l, r, et c des récepteurs membranaires. Ainsi, le site c ne peut être lié sans que le site r ne le soit et le site r ne peut être lié sans que le site l ne le soit. De son côté, l'analyse des doubles liaisons montre qu'il est impossible de former des chaînes d'au moins trois récepteurs membranaires.*

Il est important de rappeler que l'analyse ne donne qu'une sur-approximation des états accessibles. Ainsi, tout motif prouvé comme non accessible est bien inaccessible. Par contre, il n'y a aucune garantie qu'un motif non prouvé inaccessible puisse apparaître dans un état accessible depuis un des états initiaux.

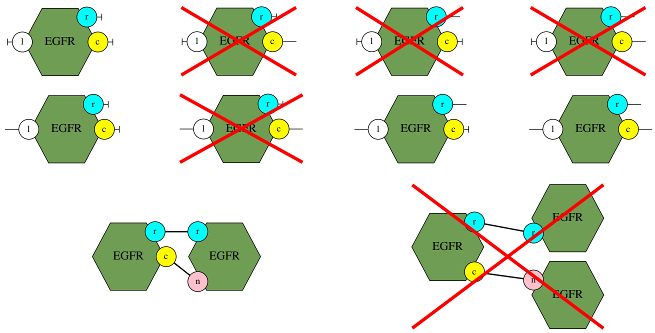
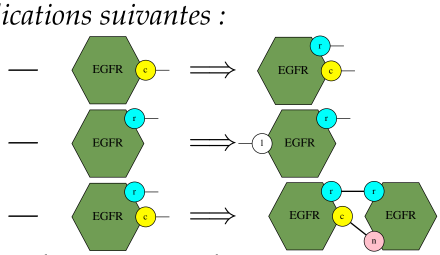


FIGURE 1.12 – *Résultat de l'analyse pour l'ensemble des motifs donnés en Fig. 1.9.*

1.3.5 Post-traitement et visualisation des résultats

L'itération de point-fixe est suivie d'une phase de traitement du résultat. Le but est essentiellement de rendre le résultat de l'analyse plus compréhensible pour l'utilisateur. Dans un premier temps, un parcours de chaque arbre de décision est effectué et chaque nœud dont tous les fils sont déclarés inaccessibles est déclaré inaccessible lui-aussi. Ensuite, tous les nœuds des arbres de décision sont explorés en répertoriant ceux dont les enfants n'ont pas tous le même statut. Ceci témoigne d'une propriété intéressante puisque dans ce cas, un des raffinements d'un motif accessible n'est pas accessible. Cette information est alors présentée sous la forme d'une implication, appelée *lemme de raffinement*, entre un motif (le nœud en question) et une liste de motifs (ses fils qui n'ont pas été prouvés inaccessibles). Une telle implication s'interprète de la manière suivante : chaque occurrence du motif de la précondition dans une configuration accessible d'une espèce biochimique peut toujours se raffiner en l'un des motifs de la postcondition.

Exemple 1.3.7. *Le résultat de l'analyse décrit en figure 1.12 donne lieu aux implications suivantes :*



Cela prouve que dans une occurrence du récepteur membranaire, le site c ne peut être lié sans que le site r ne le soit également, et que le site r ne peut être lié sans que le site l ne le soit aussi. De plus, une occurrence du récepteur dont les

sites r et c sont tous deux liés, est nécessairement liée doublement à une même occurrence du récepteur.

Par ailleurs, l'analyseur vérifie pour chaque règle si son membre gauche est compatible avec le résultat de l'analyse (avec la procédure de décision simplifiée présentée Sec. 1.3.4). Les règles pour lesquelles ce n'est pas le cas sont reportées à l'utilisateur.

Les informations trouvées par l'analyse statique sont utiles. Qu'ils soient écrits à la main ou assemblés automatiquement par fouille automatique de la littérature, la démarche suivante permet d'améliiorer la qualité des modèles. La première étape est la vérification des règles mortes. Ces règles sont souvent la conséquence, soit d'erreurs typographiques, soit d'états initiaux incomplets, soit de règles manquantes, soit de relations de causalité qui ne peuvent pas être satisfaites. La lecture des contraintes trouvées par l'analyseur permet de mieux comprendre leur origine. Elle permet également de vérifier que les invariants structurels auquel le modélisateur peut s'attendre sont bien vérifiés. L'étape suivante est d'étudier comment une occurrence de protéines peut passer d'une configuration à une autre. L'analyse des *traces locales* [45] calcule des systèmes de transitions à partir des vues locales. Ceci permet d'avoir une cartographie des changements de configuration de chaque occurrence de protéines en faisant abstraction de l'état des occurrences de protéines auxquelles cette occurrence est liée. En particulier, une étude de ces systèmes de transitions permet de calculer efficacement des transitions qui sont définitives : c'est à dire celles qui transforment la configuration d'une occurrence de protéines, sans retour possible, quel que soit le nombre de transitions ultérieures.

1.4 Réduction de modèles

Les sémantiques quantitatives permettent de définir et d'étudier le comportement des modèles au cours du temps. Dans cette section, seule la sémantique différentielle sera considérée. Elle décrit l'évolution des quantités de chaque constituant du modèle sous la forme d'équations différentielles ordinaires. Elle est donc déterministe. La sémantique différentielle demande de préciser la vitesse des réactions biochimiques. Il suffit pour cela d'associer à chaque règle de réécriture une constante pour spécifier la vitesse des réactions induites par ces règles. Toutefois, une telle sémantique ne passe pas à l'échelle des grands systèmes d'interactions, car elle requiert une variable par configuration d'espèces biochimiques. Il est donc nécessaire de proposer des méthodes de réduction pour définir des systèmes différentiels de dimensions plus petites.

La méthode présentée ici s'appuie sur l'analyse du flot d'information entre les différents sites des espèces biochimiques. Elle permet d'identifier des paires de sites dont la corrélation entre les états n'a aucune influence sur la dynamique du système considéré. Chaque configuration d'une espèce biochimique peut alors être séparée en portions plus petites qui se comportent de manière autonome. L'explosion combinatoire due à la taille des espèces biochimiques est alors coutournée.

En section 1.4.1 est illustrée une des raisons principales de l'explosion combinatoire en nombre de configurations potentielles des espèces biochimiques dans les modèles de réécriture de graphes à sites. En section 1.4.2 les calculs pour réduire la dimension d'un cas d'étude jouet sont effectués à la main. En section 1.4.3 la notion de réduction de modèle différentiel est formalisée. Ce cadre générique est ensuite appliqué aux modèles écrits en Kappa. en section 1.4.4.

1.4.1 Une brèche dans le mur de la combinatoire

Les causes de l'explosion combinatoire

Le début de cette section explique une des raisons de l'explosion combinatoire dont souffrent les systèmes d'équations différentielles qui émergent des modèles d'interactions biochimiques entre protéines.

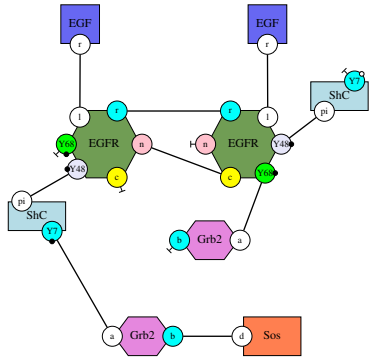


FIGURE 1.13 – *Un exemple de configuration pour un dimère. Cette configuration contient deux occurrences du site Y48 et deux occurrences du site Y68 chacune suceptible de recruter une occurrence de la protéine SoS, d'où l'explosion combinatoire du nombre potentiel de configurations d'espèces biochimiques.*

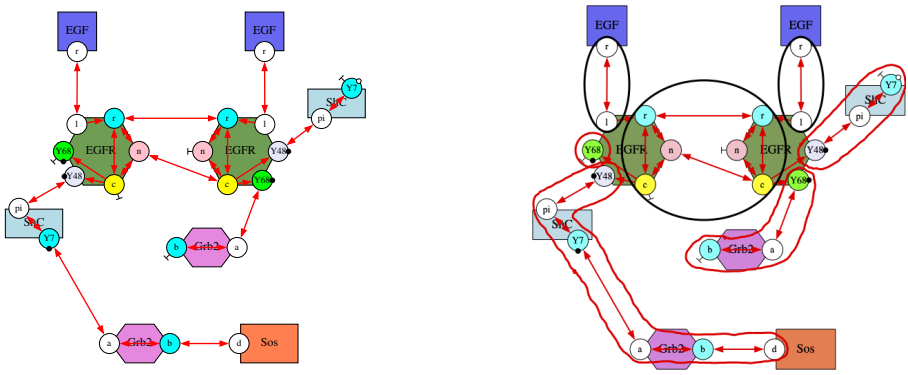
En figure 1.13 est dessinée une configuration typique d'une occurrence d'un dimère dans le modèle des premières étapes dans l'acquisition du facteur de croissante de l'épiderme [7]. Dans ce modèle, il est intéressant

de mesurer la quantité de la protéine cible *Sos* qui est attachée à la membrane de la cellule. C'est à dire la quantité de cette protéine attachée soit directement à la membrane par le biais d'une occurence de la protéine de transport *Grb2*, soit indirectement par l'intermédiaire d'une occurrence de la protéine d'échafaudage *ShC*. Chaque occurrence d'un dimère contient deux occurrences du récepteur *EGFR* et chaque occurrence du récepteur contient deux sites *Y48* et *Y68* susceptibles de recruter une occurrence de la protéine *Sos*. Les deux occurrences de la protéine recepteur étant distinguées dans le dimère, il y aura donc $n_{short}^2 \cdot n_{long}^2$ où n_{short} est le nombre d'états intermédiaires dans l'acquisition de la protéine *Sos* par la voie directe et n_{long} est le nombre d'état intermédiaire dans l'acquisition de la protéine *Sos* par la voie indirecte.

Toutefois, plutôt que de mesurer la quantité de chaque configuration d'espèces biochimiques en multipliant par le nombre d'occurrences de la protéine *Sos* que chacune a recruté, chaque occurrence des sites *Y48* et *Y68* peut être vue indépendamment en oubliant qu'ils apparaissent sur la même occurrence d'une configuration de dimères. Ceci revient à considérer la configuration d'un dimère comme un composant qui contient quatre processus, mais que savoir que ces quatres processus sont sur la même configuration d'un dimère n'est pas une information primordiale. Ces processus auraient tout autant pu se trouver sur des occurrences différentes, l'évolution de la quantité globale de la protéine *Sos* recrutée par la membrane aurait été la même. Ces processus ont alors $2 \cdot (n_{short} + n_{long})$ états différents (le facteur 2 vient du fait que les deux occurrences de la protéine récepteur sont distinguées par leur liaison asymétrique).

Le flot d'information

Le *flot d'information* entre les sites d'interactions de la configuration d'une espèce biochimique permet d'établir l'indépendance de ces quatre processus. Intuitivement, le flot d'information est une relation entre les sites d'une configuration d'espèces biochimiques qui indique l'état de quels sites est susceptible d'influencer la modification de l'état de quels autres sites. Elle prend la forme d'un graphe dont les nœuds sont les sites d'interactions d'une configuration d'espèces biochimiques et les arcs (orientés) peuvent relier soit deux sites sur une même occurrence de protéines, soit deux sites liés entre eux. Un chemin d'un site source vers un site cible témoigne que l'état du site source influence potentiellement la capacité à modifier l'état du site cible. Par contre, l'absence d'un tel chemin signifie que l'état du site source n'a aucune influence sur la modification éventuelle de l'état du site



(a) Flot d'information. *(b) Composantes fortement connexes.*

FIGURE 1.14 – *Flot d'information dans la configuration d'un dimère. En 1.14a, des flèches rouges relient deux sites sur une même occurrence de protéine ou deux sites sur un liaison. Un chemin d'un site source vers un site cible témoigne que l'état du site source influence potentiellement la capacité à modifier l'état du site cible. En 1.14b, le graphe du flot d'information est décomposé en composantes fortement connexes. Chaque composante fortement connexe est représentée par un cercle. Les composantes fortement connexes terminales sont dessinées en rouge alors que les autres le sont en noir.*

cible. La notion de flot d'information dans les modèles écrits en Kappa sera formalisée en section 1.4.4.

Réduction de la combinatoire

Une fois annotée par une sur-approximation de son flot d'information, il est possible d'identifier des portions de configurations d'espèces biochimiques dont le comportement est indépendant.

Ceci repose sur une décomposition en composantes fortement connexes des graphes formés par le flot d'information entre les sites d'interactions des configurations de chaque espèce biochimique. Il faut ainsi regrouper sur chaque graphe toute paire de sites tels qu'il existe un chemin de l'un vers l'autre et réciproquement. La décomposition du flot d'information dessinée en figure 1.14a est donnée en figure 1.14b. Deux sortes de composantes fortement connexes sont ainsi considérées. Celles en rouge sont dîtes terminales car il n'est pas possible d'en sortir en suivi le flot d'information. Les autres sont représentées en noir. Pour obtenir une portion de la configuration d'un dimère dont le comportement ne dépend pas des autres sites d'interactions de cette configuration, il suffit de com-

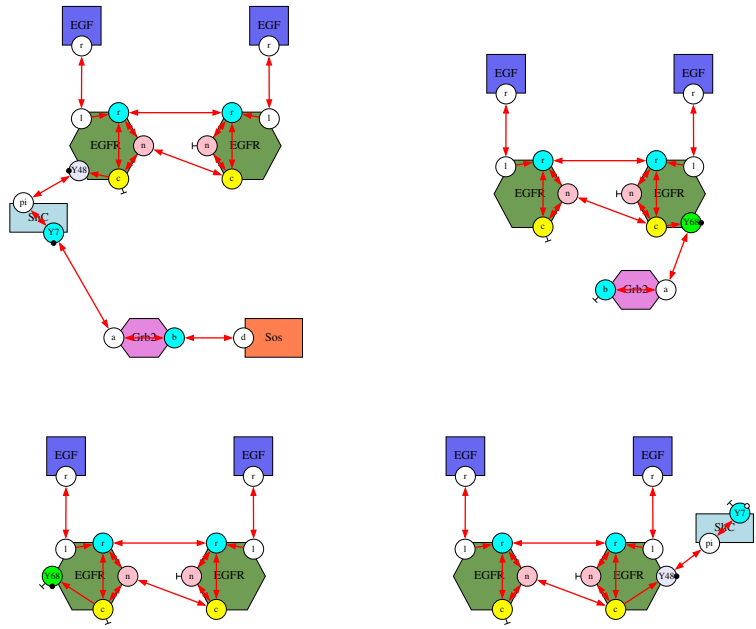


FIGURE 1.15 – *Chaque composante fortement connexe terminale est complétée des sites qui ont une influence sur elle. Chacune donne lieu à une portion de dimère qui se comporte de manière indépendante.*

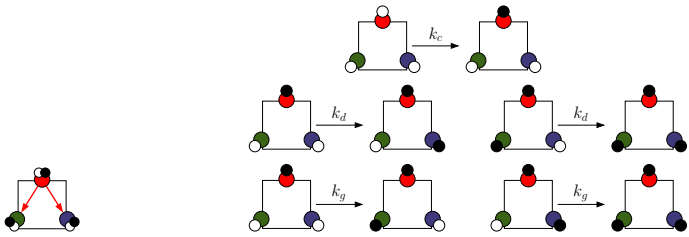
pléter chaque composante fortement connexe terminale en suivant le flot d'information en arrière. Ceci donne quatre portions de configuration d'espèces biochimiques, chacune correspondant à l'état d'avancement d'une occurrence du site *Y48* ou *Y68* dans le recrutement d'une occurrence de la protéine *Sos*, ce qui était le but.

Il semble ainsi possible d'exploiter l'absence de flot d'information pour détecter les corrélations inutiles, et ainsi découper les configurations d'espèces biochimiques en portions plus petites qui auront un comportement autonome.

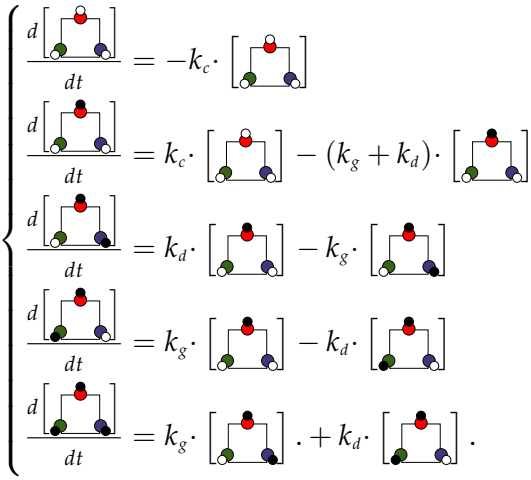
1.4.2 Exemple jouet

Un exemple jouet sera utile pour mieux comprendre comment tout ceci fonctionne en pratique. Celui-ci implique une seule sorte de protéine, munie de trois sites. Un site du haut qui sera représenté en rouge, un site gauche en vert et un site droit en bleu. Chaque site pourra être phosphorylé ou non. Il n'y a pas de liaisons dans ce modèle.

Des hypothèses sont faites en ce qui concerne comment l'état de chaque site influence la modification de l'état des autres sites. Dans chaque



(a) Carte de contacts annotée. (b) Règles-réactions.



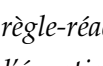
(c) Système d'équations différentielles induit.

FIGURE 1.16 – *Un exemple jouet. Une protéine contient trois sites qui peuvent être phosphorylés ou non. En 1.16a, la carte de contacts annotée indique que l'état du site du haut, en rouge, peut influencer la modification de l'état du site gauche, en vert, et l'état du site droit, en bleu. Par contre, l'état du site gauche n'a pas d'influence sur la capacité à modifier l'état du site droit, et réciproquement, l'état du site droit n'a pas d'influence sur la capacité à modifier l'état du site gauche. Ceci se traduit au niveau des règles d'interactions (voir en 1.16b). Une fois le site du haut phosphorylé (par la première réaction). Le site gauche peut se faire phosphoryler avec la constante de réaction, k_g, que le site droit soit déjà phosphorylé ou non. De même, le site droit peut se faire phosphoryler avec la constante de réaction, k_d, que le site gauche soit déjà phosphorylé ou non. Ce modèle ne contient que des étapes de phosphorylation pour garder le système différentiel de taille raisonnable. En appliquant le principe de la loi d'action de masse, ces règles-réactions induisent le système différentiel donné en 1.16c.*

occurrence de la protéine, l'état du site rouge contrôle à la fois l'évolution de l'état du site vert et celle du site bleu, comme indiqué dans la carte de contacts annotée en figure 1.16a. Par contre, l'état du site vert n'a pas d'incidence sur l'évolution de l'état du site bleu, et réciproquement, l'état du site bleu n'a pas d'incidence sur l'évolution de l'état du site vert.

Pour permettre le calcul à la main de toutes les dérivées, seul un ensemble minimal de règles-réactions sera considéré. Elles sont appelées règles-réactions car ce sont des règles Kappa dans lesquelles l'état de tous les sites des protéines qui interagissent est spécifié. Elles peuvent donc être vues comme des réactions entre des espèces biochimiques. Ces règles-réactions sont dessinées en figure 1.16b. La première spécifie que le site du haut, en rouge, peut se faire phosphoryler quand les deux autres sont déphosphorylés. La constant de cette règle-réaction est k_c. Une fois phosphorylé, le site gauche, en vert, et le site droit, en bleu, peuvent se faire phosphoryler à leur tour. Le fait que la phosphorylation au préalable du site du haut soit nécessaire à cela justifie le flot d'information du site du haut vers les deux autres sites. Sur la deuxième ligne de règles-réactions, la constante de phosphorylation du site gauche, en vert, ne dépend pas du fait que le site droit, en bleu, soit déjà phosphorylé ou non. Il n'y a donc pas de flot d'information du site bleu vers le site vert. De même, sur la troisième ligne de règles-réactions, la constante de phosphorylation du site droit, en bleu, ne dépend pas du fait que le site gauche, en vert, soit déjà phosphorylé ou non. Il n'y a donc pas de flot d'information entre l'état du site vert vers l'état du site bleu.

Le comportement de ce modèle est défini par l'application de la *loi d'action de masse*. Chaque réaction (ou ici règle-réaction) induit une contribution au système d'équations différentielles. L'activité de la réaction s'exprime comme le produit de la constante de la réaction et de la quantité des réactifs (qui apparaissent dans le membre gauche de la règle). Chaque réactif est alors consommé proportionnellement à l'activité de la réaction correspondante alors que chaque produit est ajouté dans la même quantité.

Exemple 1.4.1. *La première règle-réaction a pour activité* $k_c \cdot$  *. Elle induit deux termes dans le système d'équations différentielles :*

$$\begin{cases} \dfrac{d\left[\quad\right]}{dt} \stackrel{+}{=} -k_c \cdot \left[\quad\right] \\ \dfrac{d\left[\quad\right]}{dt} \stackrel{+}{=} k_c \cdot \left[\quad\right] \end{cases}$$

Le système différentiel complet est donné en figure 1.16c.

Il existe plusieurs conventions pour tenir compte des symétries éventuelles quand une réaction contient plusieurs occurrences d'un même réactif [10]. Elles prennent alors la forme d'un facteur correctif à appliquer à l'activité des réactions. Ce n'est le cas d'aucune réaction de cet exemple.

Il est possible d'exploiter l'absence de flot d'information du site vert vers le site bleu, et du site bleu vers le site vert. Ceci revient à découper chaque occurrence de protéines en deux portions. Comme l'état du site rouge contrôle à la fois l'évolution de l'état du site vert et celle du site bleu, ces deux portions devront se chevaucher sur le site rouge. Ainsi, les portions gauches de la protéine documenteront l'état du site rouge et du site vert alors que les portions droites documenteront l'état du site rouge et du site bleu. Le flot d'information entre les sites de ces portions de protéines est donné en 1.17a et 1.17b.

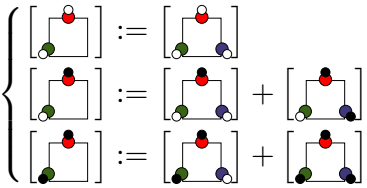
Pour utiliser les différentes configurations de portions de la protéine comme des variables, il leur faut un sens formel. Celui-ci provient du principe de dualité entre leurs significations intensionnelle et extensionnelle. De manière intensionnelle, la configuration d'une portion de la protéine peut être comprise comme le sous-graphe d'une configuration complète de la protéine. De manière extensionnelle, la configuration d'une portion de la protéine peut être vue comme le multi-ensemble de toutes les configurations complètes de la protéine qui contiennent cette portion, multipliées par le nombre de plongement de cette portion dans chaque configuration complète. Sous cette angle, la quantité la configuration d'une portion de la protéine peut se définir comme la combinaison linéaire des quantités des configurations complètes de la protéine qui contiennent cette portion (pondérée par le nombre d'occurrences). Ceci permet d'obtenir les définitions en figures 1.17c et 1.17d. En particulier, la quantité d'une portion de protéine dont aucun site n'est phosphorylé est égal à la quantité des protéines entièrement déphosphorylées (puisque le site rouge doit être phosphorylé en premier). La quantité des autres portions est obtenue en sommant la quantité des configurations obtenues selon l'état du site manquant.

Les équations qui décrivent l'évolution des quantités décrites en figures 1.17c et 1.17d peuvent ensuite être dérivées. Elles sont données en figures 1.17e et 1.17f et peuvent être vérifier analytiquement à partir des équations en figure 1.16c.

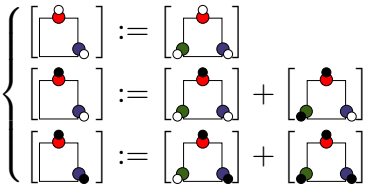
Réduire un système d'équations différentielles à 5 variables en un système à 6 variables n'est guère impressionnant. Toutefois, en regardant de plus près, les 5 variables du système initial correspondent à 1 variable pour la configuration avec le site rouge déphosphorylé et à 2×2 variables pour les autres configurations, selon que le site vert soit phosphorylé ou non, et selon que le site bleu soit phosphorylé ou non. Dans le modèle

(a) Carte de contacts pour la portion gauche de la protéine.

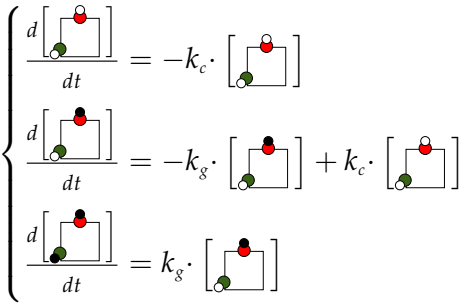
(b) Carte de contacts pour la portion droit de la protéine.



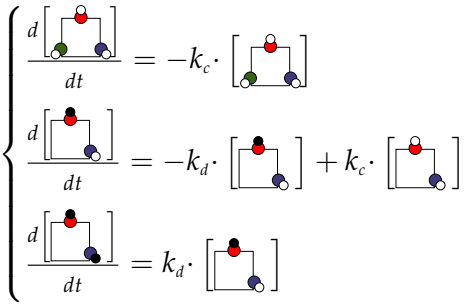
(c) Fonction d'abstraction function pour la portion gauche de la protéine.



(d) Fonction d'abstraction function pour la portion droite de la protéine.



(e) Système d'équations différentielles pour la portion gauche de la protéine.



(f) Système d'équations différentielles pour la portion droite de la protéine.

FIGURE 1.17 – *Réduction de la sémantique différentielle de l'exemple jouet (voir en figure 1.16). En utilisant le fait que l'état du site gauche de la protéine n'influence pas l'évolution de l'état du site droit et que réciproquement l'état du site droit n'influence pas l'état du site gauche, il est possible de découper la protéine en deux parties. La quantité d'un motif est définie comme la combinaison linéaire de la quantité des configurations de la protéine compatible avec ce motif en 1.17c et 1.17d. En 1.17e et 1.17f sont exprimées les dérivées de la quantité des différents motifs en fonction de la quantité de ces motifs.*

réduit, la variable pour la configurations avec le site rouge déphosphorylé est représentée de manière redondante par deux variables du système réduit (leur valeur restera donc égale au cours de l'éxecution du système réduit). Les variables pour les autres configurations correspondent à $2 + 2$ variables, car il y a deux côtés et pour chaque côté le site vert ou bleu peut être phosphorylé ou non. La réduction du modèle a donc remplacé une multiplication par une somme, ce qui aura un impact important lorsque le nombre de combinaisons possibles sera plus grand.

Il est dont possible d'exploiter l'absence de flot d'information entre des sites des occurrences de protéines, afin de détecter des corrélations entre les états de plusieurs sites qui peuvent être ignorées. Cela revient à découper les configurations des espèces biochimiques en configuration de portions d'espèces biochimiques, et ainsi casser la combinatoire du système différentiel sous-jacent. Bien entendu, cette abstraction perd de l'information. Il n'est plus possible d'exprimer les corrélations qui ont été oubliées. Par contre, l'évolution de la quantité des configurations de portions d'espèces biochimiques qui auront été gardées se décrit de manière autonome, c'est à dire uniquement à partir de la quantité des configurations de portions d'espèces biochimiques elles-mêmes.

1.4.3 Réduction de systèmes d'équations différentielles

En section 1.4.3 est proposé un cadre formel pour réduire la dimension des systèmes d'équations différentielles ordinaires.

Les systèmes d'équations différentielles décrivent des quantités qui évoluent au cours du temps selon des contraintes sur la valeur de leurs dérivées. Une réduction de modèle consiste à trouver des changements de variables pour lesquels l'évolution de nouvelles quantités, appelées observables, peut se décrire de manière autonome. Ainsi la dérivée des observables doit pouvoir s'exprimer uniquement à partir de la valeur des observables.

Plus formellement, un système d'équations différentielles est donné par un ensemble fini de variables, $\mathcal{V}$, qui représentent ici la quantité de chacune des configurations des espèces biochimiques et par une fonction, $\mathbb{F}$, de l'ensemble des fonctions réelles positives ou nulles sur l'ensemble des variables $\mathcal{V}$ vers l'ensemble des fonctions réelles sur l'ensemble des variables $\mathcal{V}$. La fonction $\mathbb{F}$ est supposée dérivable et sa dérivée est continue. Une fonction réelle positive ou nulle sur l'ensemble $\mathcal{V}$ est appelée un état potentiel. En effet, un état associe à chaque variable, la quantité de l'espèce biochimique correspondante, qui ne peut pas être négative. Une fonction réelle sur l'ensemble $\mathcal{V}$ est appelée un incrément. Une telle fonction définit

comment la quantité de chaque espèce biochimique doit être augmentée ou diminuée sur un instant infinitésimal dt.

La sémantique d'un système d'équations différentielles se définit comme la fonction qui, à un état initial X_0, associe la solution maximale, $X_{X_0}(T)$ de l'équation suivante :

$$X_{X_0}(T) = X_0 + \int_{t=0}^{T} \mathbb{F}(X_{X_0}(t)) \cdot dt.$$

définie sur l'intervalle de temps $[0, T^{max}_{X_0}[$. Comme la fonction $\mathbb{F}$ est différentiable et de dérivée continue, cette équation définit un problème de Cauchy-Lipschitz [56]. Elle a donc une unique solution maximale. Ici l'adjectif 'maximal' signifie que la valeur du paramètre $T^{max}_{X_0}$ doit être prise la plus grande possible dans l'ensemble $\mathbb{R}^+ \cup \{+\infty\}$. Il y a donc deux possibilités. Soit la solution de cette équation contient une asymptote verticale auquel cas $T^{max}_{X_0}$ est la première date à laquelle la valeur d'une variable de l'ensemble $\mathcal{V}$ diverge. Soit la solution est définie sur tout $\mathbb{R}^+$ et dans ce cas $T^{max}_{X_0}$ vaut $+\infty$.

Le comportement des systèmes réactionnels dont toutes les réactions de création (c'est à dire qui contiennent plus de produits que de réactifs) sont d'arité 0 ou 1 (c'est à dire sans réactif ou avec un seul réactif avec le coefficient stœchiométrique 1) ne diverge pas. De ce fait, leur sémantique est définie sur $\mathbb{R}^+$ quel que soit leur état initial. En particulier, seul le comportement des systèmes ouverts (zvec introduction externe de composants) est susceptible de diverger.

Réduire un système d'équations différentielles ordinaires consiste à changer de perspective en trouvant un ensemble de quantités d'intérêt, appelées les observables, dont l'évolution peut s'exprimer de manière autonome. Cela signifie que la dérivée des observables est entièrement définie par leurs valeurs. Pour formaliser ces notions, il faut d'une part relier la valeur des observables aux variables du système d'équations différentielles initial et d'autre part définir la fonction qui décrit l'évolution temporelle de ces valeurs.

Formellement, une réduction de modèle est définie par la donnée d'un ensemble fini d'*observables*, $\mathcal{V}^\sharp$, une fonction d'abstraction, ψ, qui associe à chaque état du système initial un état du nouveau système (c'est à dire une fonction de l'ensemble des fonctions positives ou nulles sur l'ensemble des observables $\mathcal{V}^\sharp$), et d'une fonction $\mathbb{F}^\sharp$ de l'ensemble des fonctions positives ou nulles sur l'ensemble des observables $\mathcal{V}^\sharp$ vers l'ensemble des fonctions réelles sur l'ensemble de observables $\mathcal{V}^\sharp$.

Des hypothèses supplémentaires sont requises pour garantir la correction de la réduction de modèle.

1. La fonction d'abstraction ψ doit être choisie linéaire à coefficients positifs. Elle doit de plus préserver les suites qui divergent, ce qui signifie que pour toute suite divergente de fonctions positives ou nulles $(\rho_i)_{i\in\mathbb{N}}$ sur l'ensemble des variables initiales $\mathcal{V}$, la suite $(\psi(\rho_i))_{i\in\mathbb{N}}$ diverge également.
2. Les fonctions d'abstraction ψ, la fonction de dynamique concrète $\mathbb{F}$ et la fonction de dynamique abstraite $\mathbb{F}^\sharp$ sont reliées par le diagramme commutatif suivant :

$$\begin{array}{ccc} (\mathcal{V} \to \mathbb{R}^+) & \xrightarrow{\mathbb{F}} & (\mathcal{V} \to \mathbb{R}) \\ \psi \downarrow & & \downarrow \psi \\ (\mathcal{V}^\sharp \to \mathbb{R}^+) & \xrightarrow{\mathbb{F}^\sharp} & (\mathcal{V}^\sharp \to \mathbb{R}) \end{array}$$

   ce qui signifie que $\psi \circ \mathbb{F} = \mathbb{F}^\sharp \circ \psi$.

Prendre la fonction d'abstraction ψ linéaire permet de la faire commuter avec la somme et l'intégrale utilisées pour définir la sémantique des systèmes différentiels. La prendre à coefficients positifs assure que l'image d'une fonction positive ou nulle est une fonction positive ou nulle, et donc que l'image d'un état concret sera un état abstrait. L'intérêt de l'hypothèse de préservation de la divergence des suites d'états sera expliqué plus tard. On peut toutefois remarquer que cette hypothèse revient à supposer que toute variable concrete apparait au moins une fois avec un coefficient non nul dans la combinaison linéaire associée à au moins un des observables. Enfin le diagramme commutatif permet à la fonction d'abstraction et à la fonction de dynamique concrete de commuter, au prix de transformer la dynamique concrete — sur les variables du système initiale — en dynamique abstraite — sur les observables. Dit autrement, partant d'un état concret, le même résultat est obtenu en calculant la dérivée du système dans cet état puis en calculant l'abstraction de cette dérivée ou en calculant l'abstraction de l'état d'abord puis en calculant la dérivée du système réduit dans l'état abstrait obtenu.

Il est maintenant possible d'appliquer la fonction d'abstraction ψ à gauche et à droite de l'égalité de Cauchy-Lipschitz, qui avait servi à définir la sémantique du système initial.

Ainsi, pour X_0 un état initial (une fonction positive ou nulle sur l'ensemble $\mathcal{V}$) et T un instant de l'intervalle $[0, T^{max}_{X_0}[$, l'équation suivante est vérifiée :

$$\psi(X_{X_0}(T)) = \psi\left(X_0 + \int_{t=0}^{T} \mathbb{F}(X_{X_0}(t))\cdot dt\right).$$

Par linéarité de la fonction d'abstraction ψ, elle se distribue sur l'addition, ce qui donne l'équation suivante :

$$\psi(X_{X_0}(T)) = \psi(X_0) + \psi\left(\int_{t=0}^{T} \mathbb{F}(X_{X_0}(t))\cdot dt\right).$$

Toujour par linéarité, elle commute avec l'intégrale, ce qui donne l'équation suivante :

$$\psi(X_{X_0}(T)) = \psi(X_0) + \int_{t=0}^{T} \psi(\mathbb{F}(X_{X_0}(t)))\cdot dt.$$

Enfin, en utilisant le fait que $[\psi \circ \mathbb{F}] = [\mathbb{F}^\sharp \circ \psi]$, il vient l'équation suivante :

$$\psi(X_{X_0}(T)) = \psi(X_0) + \int_{t=0}^{T} \mathbb{F}^\sharp(\psi(X_{X_0}(t)))\cdot dt.$$

De ce fait, la fonction qui à tout instant T dans l'intervalle $[0, T^{max}_{X_0}[$ associe l'état abstrait $\psi(X_{X_0}(T))$ est solution de l'équation différentielle suivante :

$$Y_{Y_0}(T) = Y_0 + \int_{t=0}^{T} Y_{Y_0}(t)\cdot dt,$$

pour $Y_0 = \psi(X_0)$.

C'est en fait une solution maximale de cette équation. C'est en effet clair lorsque $T = +\infty$. Dans le cas contraire, l'expression $X_{X_0}(T)$ diverge quand T tend vers $T^{max}_{X_0}$. Puis comme la fonction ψ préserve la divergence des suites et qu'elle est continue, l'expression $\psi(X_{X_0}(T))$ diverge aussi quand T tend vers $T^{max}_{X_0}$. Ainsi la fonction qui à tout instant T dans l'intervalle $[0, T^{max}_{X_0}[$ associe l'état abstrait $\psi(X_{X_0}(T))$ ne peut pas être prolongée.

La sémantique d'un système différentiel et la réduction de celle-ci a été formalisé. Il s'agit maintenant de définir la sémantique différentielle d'un système Kappa et de montrer comment trouver un ensemble d'observables dont l'évolution temporelle peut se décrire de manière autonome.

1.4.4 Application à Kappa

Il reste maintenant à spécialiser ce cadre générique pour réduire la sémantique différentielle des systèmes de règles du langage Kappa.

Sémantique différentielle

Comme il a été vu en section 1.2.7, un ensemble de règles Kappa engendre un réseau de réactions biochimiques. Chaque réaction étant alors constituée de deux n-uplets, l'un formé d'espèces appelées les réactifs de

la réaction et l'autre d'espèces appelées les produits de la réaction. Pour définir la sémantique différentielle d'un ensemble de règles, il faut de plus associer à chacune de ces règles un constante d'interaction. Chaque réaction hérite alors de la constante de sa règle originale (ou de la combinaison linéire des constantes de ses règles initiales si elle peut être obtenue de différentes manières).

Comme expliqué en section 1.4.2, le comportement d'un réseau réactionnel est défini par l'application de la loi d'action de masse. La contribution de chaque réaction au système d'équations différentielles est définie de la manière suivante. Chaque réactif est alors consommé proportionnellement à l'activité de la réaction correspondante alors que chaque produit est ajouté dans la même quantité, l'activité de la règle s'exprimant comme le produit de la constante de la règle et de la quantité des réactifs (qui apparaissent dans le membre gauche de la règle).

Il existe plusieurs conventions pour tenir compte des éventuelles symétries dans les règles de réécriture et dans les réactions. Des précisions sur ce sujet peuvent être trouvées dans cette publication [10]. Nous négligeons ici la question en considérant que les taux des règles d'interactions et des réactions sont prises tels quels sans facteur correctif. La régle de trois permet de tenir compte des facteurs correctifs pour les autres conventions. Par ailleurs, toutes les réactions ne sont pas prises en compte. En effet, seules les réactions qui ont le même nombre de réactifs que le nombre de composante connexe dans le membre gauche de la règle qui les a engendrées, sont prises en compte. Intuitivement, la sémantique différentielle représente le comportement du système discret dans un milieu homogène et parfaitement fluide dont le volume tend vers 0. Aussi, la chance de tirer au sort deux parties d'une même occurrence de configuration d'espèces biochimique pour plonger plusieurs composantes connexes du membre gauche d'une règle d'interaction tend vers 0.

Inférence du flot d'information

En Kappa, une approximation supérieure du flot d'information entre les sites des configurations d'espèces biochimiques peut être obtenue en inspectant chaque règle d'interactions. En effet, il ne peut y avoir un flot d'information d'un premier site vers un second site dans la configuration d'une espèce biochimique que s'il existe existe un règle qui transforme l'état du second site selon des conditions sur l'état du premier site.

De ce fait, il est possible d'annoter la configuration d'une espèce biochimique par une approximation supérieure de son flot d'information en calculant toutes les instances des composantes connexes du membre

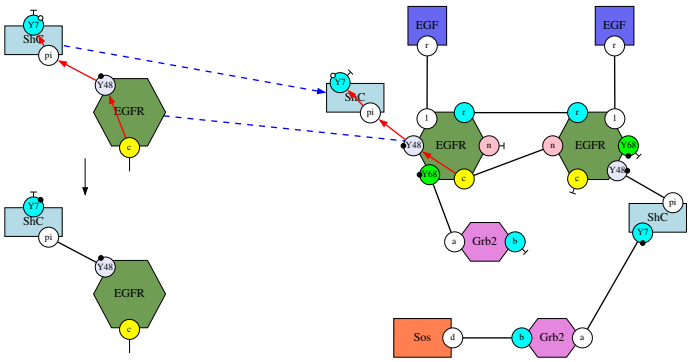


FIGURE 1.18 – *Flot d'information induit par une règle d'interactions sur la configuration d'une espèce biochimique. Pour tout plongement d'une composante connexe du membre gauche de la règle (ici dessiné en haut) vers la configuration de l'espèce biochimique et tout chemin allant d'un site, ici le site c de l'occurrence du récepteur EGFR, et un site dont l'état est modifié par la règle, ici le site Y7 de l'occurrence de la protéine d'échafaudage ShC, le chemin est reporté sur la configuration de l'espèce biochimique selon le plongement.*

gauche de chaque règle d'interactions du modèle et en reportant le long des plongements correspondants chaque chemin constitué d'étapes entre deux sites d'une même occurence de protéines ou entre deux sites sur un même lien et qui se termine dans un site dont l'état est modifié par la règle. Ainsi, si deux sites de la configuration de l'espèce biochimique sont l'image par un plongement de deux sites reliés par une étape de ce chemin, un arc de flot d'information est placé entre ces deux sites dans la configuration de l'espèce biochimique. Cette construction est illustrée en figure 1.18.

Il n'est cependant pas concevable d'annoter ainsi toutes les configurations d'espèces biochimiques, car celles-ci sont trop nombreuses en général. Une telle approche ne passerait pas à l'échelle de modèles d'interactions biochimiques même de taille modeste. Il est possible au contraire de résumer l'annotation de toutes les configurations d'espèces biochimiques sur la carte de contacts. Il faut se souvenir que la carte de contacts est fournie une abstraction de l'ensemble de toutes les configurations des espèces biochimiques d'un modèle en repliant entre eux les occurrences de chaque type de protéines. La carte de contacts peut être calculée directement par inspection des états initiaux et des règles du modèle, ou par analyse statique, quitte à introduire des états fictifs. Comme tout motif se projète de manière unique sur la carte de contacts en envoyant toutes les occurrences d'une même protéine sur l'unique occurrence de cette protéine dans le carte de contacts, cette projection permet de répertorier le flot d'information induit par une

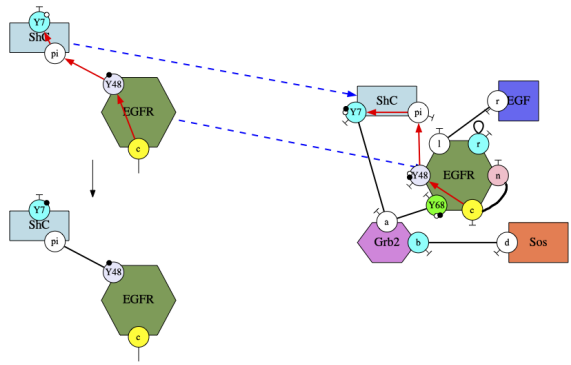


FIGURE 1.19 – *Flot d'information induit par une règle sur la carte de contacts. Pour tout chemin allant d'un site, ici le site c de l'occurrence du récepteur EGFR, et un site dont l'état est modifié par la règle, ici le site Y7 de l'occurrence de la protéine d'échafaudage ShC, le chemin est reporté sur la carte de contacts en suivant l'unique projection du membre gauche de la règle sur la carte de contacts.*

règle directement sur la carte de contacts. Ainsi tout arc qui apparaît dans un chemin constitué d'étapes entre deux sites d'une même occurence de protéines ou entre deux sites sur un même lien, partant d'un site dans un composante connexe dans le membre gauche d'une règle et finissant sur un site dont l'état est modifié par la règle de réécriture, est reporté sur la carte de contacts comme un flot d'information potentiel entre l'image de ces deux sites. Cette construction est illustrée en figure 1.19.

Le cas des composantes connexes qui ne sont pas modifiées dans une règle est particulier. Il n'y a en effet aucun chemin de flot d'information dans celles-ci. Pour garantir qu'il sera possible d'exprimer la quantité des ces motifs dans le système réduit, il faut considérer qu'au moins un site d'interraction, au choix, pour chacune de ces composantes connexes est modifié, et donc reporter tous les chemins partant des autres sites et terminant sur ce site dans la carte de contacts annotée.

Ainsi, il est possible de résumer de le flot d'information induit par les règles entre les différents sites des configurations des espèces biochimiques sur la carte de contacts. Il n'est pas nécessaire d'énumérer les différentes configurations des espèces biochimiques. Il suffit en effet d'identifier les chemins dont chaque étape relie soit deux sites d'une même occurrence de protéines, soit deux sites qui partagent un lien, et qui se terminent sur un site qui est modifié par la règle, et de répertorier ce chemin sur la carte de contacts.

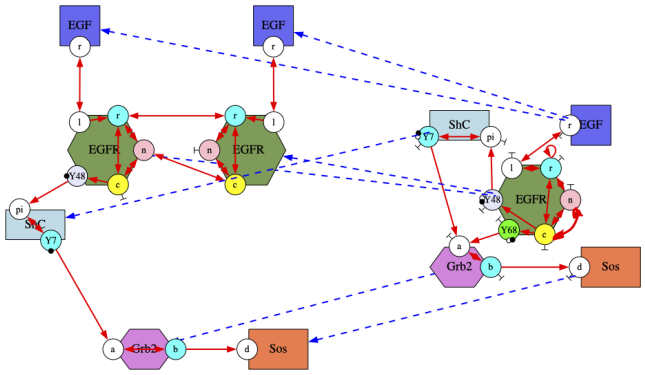
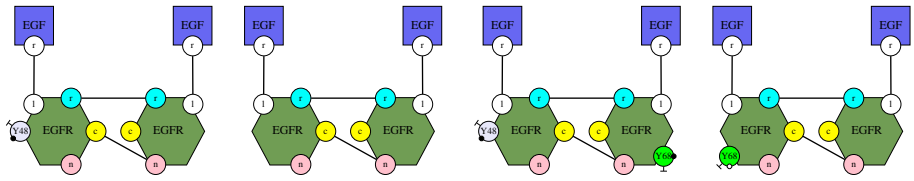


FIGURE 1.20 – *Annotation d'un motif par le flot d'information décrit dans la carte de contacts. Le motif se plonge de manière unique dans la carte de contacts en envoyant chaque occurrences d'agent sur l'unique occurrence de cet agent dans le carte de contacts. Deux sites du motifs sont reliés par un arc de flot d'information si et seulement si ils sont sur le même agent ou sur une même liaison et si leur image dans la carte de contacts est lié par un arc de flot d'information.*

Pré-fragments et fragments

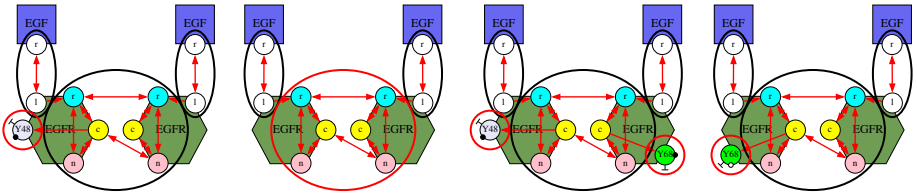
Réciproquement, l'approximation supérieure du flot d'information sur la carte de contacts permet de reconstruire le flot d'information sur n'importe quel motif. Il suffit pour cela de considérer l'image inverse de la projection qui envoie ce motif sur la carte de contacts. Ainsi, il y aura un arc de flot d'information entre deux sites d'un motif si et seulement si il y a un arc de flot d'information sur l'image de ces deux sites par sa projection sur la carte de contacts. Cette construction est illustrée en figure 1.20.

L'annotation d'un motif par son flot d'information permet de vérifier si d'une part il ne comporte pas de sites dont la corrélation entre les états ne présente pas d'intérêt vis à vis du comportement du système modélisé, et si d'autre part il contient tous les sites d'interactions susceptibles d'influencer son comportement. Un motif sera gardé comme observable lorsque ces conditions seront toutes deux réalisées. Formellement, la première condition est satisfaite si le graphe formé par l'annotation du motif ne contient qu'une composante fortement connexe terminale. Dans ce cas, le motif sera appelé un *pré-fragment*. La seconde condition est réalisée lorsqu'un pré-fragment ne se plonge pas dans un autre pré-fragment. C'est à dire, s'il est impossible de compléter le motif, sans briser la condition que le graphe formé par l'annotation de ce motif par son flot d'information n'admet qu'une seule composante fortement connexe terminale. Dans ce cas, le motif sera appelé un *fragment*.



(a) Premier motif. (b) Deuxième motif. (c) Troisième motif. (d) Quatrième motif.

FIGURE 1.21 – *Parmi ces quatre motifs, lesquels sont des fragments ?*



(a) Premier motif. (b) Deuxième motif. (c) Troisième motif. (d) Quatrième motif.

FIGURE 1.22 – *Pour décider lesquels de ce motifs sont des fragments, la première étape est de reporter l'annotation du flot d'information entre les sites d'interactions de la carte de contacts sur ces motifs et de décomposer le graphe obtenu en composantes fortement connexes.*

Exemple 1.4.2. *En figure 1.21 sont dessinés quatre motifs. Parmi ceux-ci, lesquels sont des fragments. Pour répondre à cet question, il faut annoter ces motifs par le flot d'information répertorié sur la carte de contacts. Seuls les motifs pour lesquels le graphe formé par l'annotation du flot d'information n'a qu'une composante fortement connexe terminale sont des pré-fragments.*

Les motifs annotés avec le flot d'information ainsi obtenu sont représentés en figure 1.22, ainsi que les compostantes fortement connexes du graphe de leur flot d'information. Il apparaît que celui du troisième motif a deux composantes fortement connexes terminales. Ce n'est donc pas un pré-fragment. En revanche, les trois autres motifs le sont bien.

Le deuxième motif se plonge dans le premier motif. Ce n'est dont pas un fragment. Par contre, le premier et le quatrième motif sont des fragments. En effet, si on leur ajoute une occurrence de site, ce sera forcément soit une occurrence du site Y48 soit une occurrence du site Y68. Dans les deux cas, cela ajoutera une composante fortement connexe terminal dans le graphe du flot d'information du motif ainsi obtenu. De ce fait, ces motifs ne se plongent pas dans des fragments, autres qu'eux-même.

Ainsi seuls les deux motifs représentés dans les figures 1.21a et 1.21d sont des fragments.

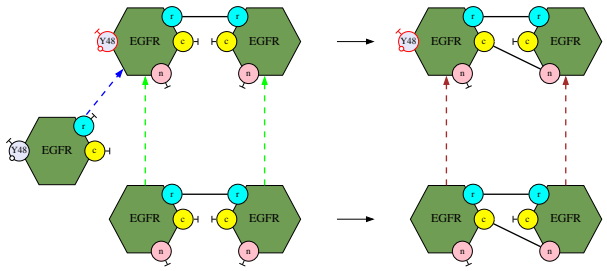
Sémantique différentielle réduite

Il reste à montrer que l'ensemble des fragments d'un modèle peut être pris comme ensemble des observables de la sémantique différentielle réduite. En d'autres termes, il faut montrer qu'il est possible d'exprimer la quantité de chaque fragment crée et détruite sur un temps infinitésimal en fonction de la quantité des autres fragments.

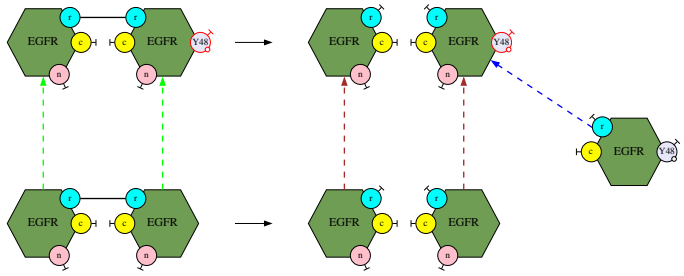
Raffinements orthogonaux Avant tout, il est important de remarquer que la quantité de chaque pré-fragment s'exprime comme une combinaison linéaire de la quantité des fragments. Cette propriété repose sur l'utilisation d'arbres de décision pour raffiner étape par étape un pré-fragment en un ensemble de pré-fragments orthogonaux plus précis [29, 64, 46]. Par définition, un pré-fragment qui n'est pas un fragment se plonge dans un pré-fragment plus grand. Ce dernier contient donc un site qui n'est pas présent dans le pré-fragment initial. Le pré-fragment initial peut alors remplacé par l'ensemble des pré-fragments obtenus en ajoutant ce site pour tous les états possibles de ce sites. L'arbre de décision est ainsi construit par induction jusqu'à ce que les feuilles de l'arbre soient toutes des fragments. Pour chaque nœud, la quantité du motif est égale à la somme des quantités des motifs sur ses feuilles, ce qui garantit que la quantité du pré-fragment initial est égal à la somme des quantités des fragments sur les feuilles de l'arbre. À noter que certains fragments peuvent apparaître plusieurs fois dans les feuilles de l'arbre, ce qui peut donc donner des coefficients différents de 1 dans la combinaison linéaire ainsi obtenue.

Spécialisation d'une règle à la consommation ou la production d'un pré-fragment La seconde étape consiste à exprimer la consommation et la production de la quantité de chaque fragment sur un temps infinitésimal, en fonction de la quantité des autres fragments. Cette partie présente un résultat plus général qui exprime la quantité consommée et produite pour chaque pré-fragment en fonction de la quantité des autres pré-fragments. Ceci répond bien à la question puisque la partie précédente a permis de traduire la quantité de chaque pré-fragment comme une combinaison linéaire de la quantité des fragments du modèle.

Cette approche repose sur une spécialisation des règles du modèle à la consommation ou à la production d'une occurrence d'un pré-fragment à une position donnée. En effet, chaque chevauchement potentiel entre un pré-fragment et le membre gauche ou droit d'une règle permet de raffiner celle-ci en ajoutant des deux côtés de cette règle toute information présente dans le pré-fragment qui ne serait pas dans le membre en question. Cette



(a) Exemple de raffinement à gauche.



(b) Exemple de raffinement à droite.

FIGURE 1.23 – *Deux raffinements de règles d'interactions. En 1.23a, la règle de formation de liaison asymétrique dans les occurrences de dimères (voir la figure 1.7e page 16) est raffinée en faisant se chevaucher un motif avec son membre gauche. En 1.23b, la règle pour briser la liaison symétrique dans les occurrences de dimères (voir la figure 1.7d page 16) est raffinée en faisant, cette fois ci se chevaucher ce motif avec le membre droit de la règle. Dans les deux cas, la partie commune au motif et au membre de la règle contient un site dont l'état est modifié par la règle d'interaction. De plus, l'information présente dans le motif qui manque dans le membre de la règle est ajoutée (en rouge) aux deux membres de la règle pour spécialiser celle-ci à la consommation ou à la production de ce motif par la règle.*

construction avait déjà été utilisée page 25 pour détecter parmi un ensemble de motifs d'intérêts certains qui ne sont pas accessibles dans un modèle.

Plus formellement, un chevauchement entre deux motifs se caractérise par un troisième motif et deux plongements des deux premiers motifs vers ce troisième motif, de sorte qu'aucune information qui ne serait présente ni dans le premier motif, ni dans le second ne soit présente dans le troisième. En d'autre terme, le troisième motif doit être minimal. Dans la théorie des catégories, ceci correspond à une *somme amalgamée*.

Exemple 1.4.3. *En figure 1.23 sont donnés des exemples de raffinements de règles.*

Le premier est une spécialisation de la règle de formation du lien asymétrique dans les dimères (voir en figure 1.7e page 16) à la consommation des occurrences de

la protéine EGFR dont les sites d'interactions r et c sont libres et le site Y48 libre et non phorphorylé (l'état des sites l, n et Y68 ne sont pas mentionnés) en première position du membre gauche de la règle. Il faut pour cela considérer le chevauchement obtenu en fusionnant l'occurrence de la protéine EGFR du motif avec la première occurrence de cette protéine dans le membre gauche de la règle (un autre chevauchement pourrait être obtenu de la même manière en l'identifiant à la seconde occurrence de cette protéine). La seule information qui est présente dans le motif sans l'être dans le membre gauche de la règle est que le site Y48 doit être libre et non phosphorylé. Le raffinement de la règle est donc obtenu en ajoutant ces informations à gauche et à droite de la règle. Le résultat est donc une spécialisation de la règle de liaison asymétrique au cas où la première occurrence de la protéine EGFR a son site Y68 libre et phosphorylé.

Le second raffinement est une spécialisation de la règle pour briser un lien symétrique dans les dimères (voir en figure 1.7d page 16) à la production des occurrences du même motif en deuxième position du membre droit de la règle. Il résulte du chevauchement obtenu en fusionnant l'occurrence de la protéine EGFR du motif avec la seconde occurrence de cette protéine dans le membre droit de la règle. La seule information qui est présente dans le motif sans l'être dans le membre gauche de la règle est que le site Y48 doit être libre et non phosphorylé. Le raffinement de la règle est donc obtenu en ajoutant ces informations à gauche et à droite de la règle. Le résultat est donc une spécialisation de la règle pour briser la liaison symétrique dans les occurrences de dimères dans le cas où la deuxième occurrence de la protéine EGFR a son site Y68 libre et phosphorylé.

Termes de consommation et de production d'un motif

Les raffinements d'une règle par un motif permettent d'exprimer la quantité de ce motif consommée et produite sur un temps infinitésimal par cette règle en fonction de la quantité des autres motifs. De plus lorsque ce motif est un pré-fragment et que le chevauchement entre le pré-fragment et le membre de la règle, qui a induit le raffinement, contient un site modifié par la règle sur sa partie commune, alors par construction du flot d'information, les composantes connexes de la règle raffinée sont toutes des pré-fragments. Comme la quantité de chaque pré-fragment s'exprime comme un combinaison linéaire de la quantité des fragments, ceci permet d'exprimer la consommation et la production de chaque fragment sur un temps infinitésimal en fonction de la quantité des autres fragments.

Par ailleurs, comme la sémantique différentielle ne prends en compte que les régles-réactions qui ont le même nombre de composantes connexes dans leur membre gauche que la règle dont elles sont issues, seuls les

raffinements qui préservent le nombre de composantes connexes dans le membre gauche ont une contribution dans le système différentiel réduit.

Ainsi, pour chaque chevauchement entre un pré-fragment et le membre gauche d'une règle tels qu'un site modifié appartienne à la partie commune et que les membres gauches de la règle initiale et de la règle raffinée aient le même nombre de composantes connexes, si l'on note k la constante de la règle, *aut* le nombre de plongement entre le pré-fragment et lui-même, et $C'_1,\ldots,C'_n$ les pré-fragments qui constituent le membre gauche de la règle raffinée, la consommation de ce pré-fragment sur un temps infinitésimal est donnée par l'expression suivante :

$$\frac{k\cdot\Pi_{i=1}^{n}\left[C_i\right]}{aut}.$$

De la même manière, pour chaque chevauchement entre un pré-fragment et le membre droit d'une règle tels qu'un site modifié appartienne à la partie commune et que les membres gauches de la règle initiale et de la règle raffinée aient le même nombre de composante connexe, si l'on note k la constante de la règle, *aut* le nombre de plongement entre le pré-fragment et lui-même, et $C'_1,\ldots,C'_n$ les pré-fragments qui constituent le membre gauche de la règle raffinée, la consommation de ce pré-fragment sur un temps infinitésimal est donnée par l'expression suivante :

$$\frac{k\cdot\Pi_{i=1}^{n}\left[C_i\right]}{aut}.$$

Ainsi le principe de la loi d'action de masse opère directement sur les quantité de motifs, ce qui permet d'exprimer directement un système d'équations différentielles ordinaires réduit. Appliqué à des fragments, il offre, par construction, une abstraction exacte du système d'équations différentielles initial. Il est important de noter que ce qui a été réalisé est en fait une factorisation de l'expression de la dérivée des quantités de fragments. En effet, cette dérivée peut s'exprimer en sommant l'activité de toutes les règles-réactions dans lesquels ce fragment est produit ou consommé. Chaque activité est alors le produit d'une constante et de quantités de configurations d'espèces biochimiques. En exprimant cette dérivée par des concentrations de fragments, des configurations d'espèces biochimiques ont été regroupées facteur par facteur pour retrouver des motifs, transformant ainsi des sommes de produits en produits de sommes. La préservation des termes de l'expression initiale est une preuve purement combinatoire qui constitue le chapitre 8.3 de [14].

Par ailleurs, un fragment peut apparaître dans un réaction sans être consommé, ni produit. Il est en général impossible d'exprimer la quantité correspondante en fonction de la quantité des fragments. Cela n'a pas d'incidence car cela conduit à des contributions négatives et positives qui s'annule parfaitement dans le système d'équations différentielles initial.

Exemple 1.4.4. *Pour continuer l'exemple 1.4.3, il est possible d'exprimer le terme de consommation et de production liés aux raffinements représentés en figure 1.23. Ainsi le raffinement gauche en figure 1.23a engendre le contribution suivante :*

$$\frac{d\left[\begin{array}{c}\text{r}\\ \text{c}\ \text{EGFR}\ \text{Y48}\end{array}\right]}{dt} = k\cdot\left[\begin{array}{c}\text{r}\quad\text{r}\\ \text{Y48}\ \text{EGFR}\ \text{c}\quad\text{c}\ \text{EGFR}\\ \text{n}\quad\quad\text{n}\end{array}\right],$$

où k est la constante de la règle de formation de la liaison asymétrique dans les dimères.

De même, le raffinement droit en figure 1.23b engendre le contribution suivante :

$$\frac{d\left[\begin{array}{c}\text{r}\\ \text{c}\ \text{EGFR}\ \text{Y48}\end{array}\right]}{dt} = k_d\cdot\left[\begin{array}{c}\text{r}\quad\text{r}\\ \text{Y48}\ \text{EGFR}\ \text{c}\quad\text{c}\ \text{EGFR}\\ \text{n}\quad\quad\text{n}\end{array}\right]$$

où k_d est la constante de la règle pour briser les liaisons symétriques dans les dimères.

Ces deux contributions sont à doubler. En effet, on obtient exactement les mêmes termes en tenant compte du chevauchement entre le même motif et la seconde occurrence de la protéine EGFR dans le membre gauche de la règle et de celui entre ce motif et la première occurrence de la protéine EGFR dans le membre droit de la règle.

1.4.5 Pour aller plus loin

Dans cette partie, une méthode de réduction de la sémantique différentielle des modèles Kappa a été présentée. Elle se base sur une analyse du flot d'information entre les différents sites d'interactions des configurations des espèces biochimiques pour identifier des corrélations entre l'état de certains sites qui sont inutiles pour décrire le comportement temporel du modèle. Ceci permet d'identifier des motifs d'intérêt, appelés fragments, dont l'évolution peut être décrit uniquement en fonction de la quantité des fragments dans le système, et donc sans connaître la quantité de chaque configuration d'espèces biochimiques. Ainsi, il est possible de générer automatiquement un système réduit d'équations différentielles

décrivant l'évolution de la quantité de ces fragments au cours du temps. Il suffit pour cela de spécialiser les règles à la consommation ou à la production des fragments aux différentes positions compatibles sur la règle. Ceci évite d'avoir à énumérer les réactions du modèle ou l'ensemble des configurations d'espèces biochimiques. Cette méthode a réussi à exploiter la structure en cascade du flot d'information pour réduire la dimension de modèles à large échelle de voies de signalisation intracellulaire.

La présentation de la méthode a été simplifiée en ignorant les règles avec effets de bords. Ces règles permettent de détruire des occurrences de protéines sans en décrire entièrement l'état ou de briser une liaison sur un site sans spécifier à quel site ce site est lié. Ceci pose deux difficultés techniques. Premièrement la spécialisation des règles pour la production d'un fragment à une certaine position peut donner lieu à plusieurs raffinements de la règle en fonction des effets de bord à effectuer. Ensuite, la description précise de ces effets de bord impose d'étendre la syntaxe pour quantifier existentiellement sur la présence de liens non spécifiés entre des agents. Toutefois, il a été montré dans [16] que la quantité de ces motifs étendus pouvait s'exprimer sous forme de somme alternée de motifs classiques grâce au développement de la formule du binôme.

La méthode présentée se repose sur une approximation uniforme du flot d'information. En effet, le flot d'information est résumé sur la carte de contacts qui ne comporte qu'une occurrence de chaque type des différentes protéines. Une réduction de modèle plus compacte peut parfois être obtenue en utilisant une approximation non uniforme du flot d'information [17, 14]. Le calcul des fragments est alors un peu plus compliqué. Il faut notamment appliquer un opérateur de clôture sur l'approximation du flot d'information afin de garantir que l'annotation du membre gauche de l'application d'une règle spécialisée à la production d'un fragment est toujours plus riche que celle de son membre droit. Cette propriété est toujours assurée avec une approximation uniforme du flot d'information puisque chaque type de protéines n'apparaît qu'une seule fois sur la carte de contacts.

Les analyses non-uniforme offrent une hiérarchie d'approximations pour réduire la sémantique différentielle des modèles écrits en Kappa. Il n'existe pour l'instant pas de méthodes pour savoir quels contextes doivent être distingués et ainsi, choisir le meilleur compromis entre précision et complexité de l'analyse.

Des approches analytiques peuvent trouver de meilleurs changements de variables pour réduire la sémantique différentielle des réseaux réactionnels. Toutefois, comme l'espace des changements de variables est vaste, elles se restreignent le plus souvent à l'exploration d'un sous-ensemble de celui-ci, comme celui des bisimulations qui sont induites par une relation

d'équivalence entre les différentes configurations d'espèces biochimiques du modèle [19, 20]. L'approche présentée ici présente deux avantages. Comme elle s'appuie sur des propriétés structurelles, elle n'a pas besoin de la représentation explicite du réseau réactionnel ou de sa sémantique différentielle initiale. Par ailleurs, les changements de variables qu'elle produit ne sont en général pas exprimables comme une bisimulation induite par une relation d'équivalence entre les différentes configurations d'espèces biochimiques du modèle. Par contre, cette méthode n'offre aucune garantie d'optimalité sur le sous-ensemble des changements de variables considéré. Par ailleurs, les réductions de modèles par les approches analytiques ne sont valables que pour un jeu de constantes de réactions données, alors que la méthode proposée dans cette partie donne une réduction de modèles qui est valable quelque soit les constantes des règles d'interactions. En contre-partie, cette méthode ne peut pas exploiter les relations numériques éventuelles entre les contantes des règles d'interactions.

Les méthodes de réduction exactes sont par définition limitées par le critère de correction qui est trop exigeant. De plus, elles n'offrent aucune marge de manœuvre sur le choix des observables. Une alternative est de considérer des réductions numériquement approchées. En fixant à la main un ensemble d'observables, il est possible d'abstraire l'état du système par une hyper-boîte qui encadre la valeur de chaque observable entre les bornes d'un intervalle. Le modèle réduit consiste alors en une équation différentielle sur les bornes des différents intervalles, ou d'un point de vue géométrique sur les coordonnées des hyper-faces de l'hyper-boîte. Ces équations sont obtenues en majorant la dérivée de chaque observable au voisinage de l'hyper-face supérieure correspondante et en minorant la dérivée de chaque observable au voisinage de l'hyper-face inférieure correspondante. Cette approche a été utilisée pour retranscrire dans un cadre formel des méthodes de troncation [69] et des méthodes de tropicalisation [5] tout en fournissant, à chaque instant, un encadrement correct de la valeur des observables, ce qui va bien au delà des résultats de convergence asymptotiques habituels.

1.5 Conclusion

Le langage Kappa a été présenté, ainsi que deux applications des méthodes formelles pour les modèles écrits dans ce langage.

La première est une analyse statique qui permet de détecter parmi un ensemble de motifs d'intérêt lesquels peuvent potentiellement apparaître dans des configurations d'espèces biochimiqes dans les traces d'exécution

d'un modèle. Du point de vue de l'utilisateur, cette analyse permet de trouver – ou de retrouver – des propriétés structurelles sur les différentes configurations des occurrences des protéines au sein des configurations des espèces biochimiques : elle détecte quelles sont les relations entre l'état des sites des occurrences d'une protéine (Est-ce que tel site peut être lié sans que tel autre le soit ? Est-ce que ce site peut être lié sans être phosphorylé ?) ; elle permet de vérifier si deux occurrences de protéines liées entre-elles sont, oui ou non, nécessairement localisées au même endroit au sein d'une hiérarchie statique de compartiments ; elle analyse si une occurrence de protéines peut être doublement liée à une autre ou si elle peut être liée à deux occurrences différentes de protéines. En plus, de permettre la détection de règles mortes, qui ne pourront jamais être appliquées dans le modèle, le résultat est présenté graphiquement sous la forme de lemmes de raffinement, le rendant compréhensible et facilement utilisable pour des analyses ultérieures. Il est ensuite possible de se concentrer sur le comportement des occurrences d'une protéine en particulier et d'obtenir un système de transitions pour décrire leurs changements potentiels de configuration.

Cette analyse passe à l'échelle de grands modèles. Cependant, pour ceux-ci, le temps de calcul reste trop important pour permettre une analyse interactive et sans latence pendant l'écriture même des modèles. Une formulation du calcul du plus petit point fixe abstrait sous forme de résolution de clauses de Horn pourrait donner lieu à une analyse incrémentale. Celle-ci permettrait de mettre à jour très rapidement le résultat de l'analyse lorsque des règles sont retirées, ajoutées ou modifiées dans un modèle. Par ailleurs, une collaboration étroite avec les modélisateurs est toujours nécessaire pour identifier des nouvelles familles de propriétés d'intérêt. Un autre axe de recherche est l'intégration de l'analyse statique dans des cycles de modélisations automatiques. En effet, les méthodes de fouille de la littérature basées sur l'intelligence artificielle et le traitement automatique des langages naturels pourront bénéficier de l'analyse statique d'une part pour évaluer le bien fondé d'une étape de raffinement de modèle et d'autre part pour orienter les méthodes automatiques dans leur recherche de nouvelles règles.

En ce qui concerne la modélisation en Kappa, il est important de considérer non pas un réseau d'interactions biomoléculaires dans son individualité, mais une famille de réseaux d'interactions pouvant représenter un système dans différents contextes cellulaires et ses évolutions potentielles. Les travaux sur la plate-forme de modélisation Kami vont dans ce sens [51]. Il est aussi important de proposer des méthodes pour assister le modélisateur dans la construction de modèles, afin d'agglomérer des

informations partielles sur les interactions biomoléculaires en les raffinant progressivement. Une approche inspirée des approches déductives, qui assimile le processus de modélisation à une recherche de preuves assistée par ordinateur, est très prometteuse [55]. Dans ce contexte, une analyse statique le plus tôt possible dans la chaîne de modélisation doit être développée pour aider au mieux le modélisateur dans sa tâche.

La seconde application des méthodes formelles porte sur la réduction exacte de la sémantique différentielle des modèles. Cette approche se base sur le flot d'information entre les différents sites des configurations des espèces biochimiques du modèles, afin d'en identifier des fragments dont le comportement différentiel peut s'exprimer uniquement en fonction de la quantité des différents fragments dans le modèle. La loi d'action de masse, qui permet d'exprimer l'évolution de la quantité de chaque configuration d'espèces biochimiques, s'applique alors directement pour exprimer l'évolution des quantités des différentes configurations de fragments. Ceci permet de générer un système différentiel réduit directement sans avoir à énumérer les réactions biochimiques du système initial ou les différentes configurations de ses espèces biochimiques. Le système d'équations différentielles initials et le système réduit sont liées par la relation formelle suivante : les solutions du système réduit sont la projection exacte des solutions du système initial par une fonction affine.

Les modèles sont de plus en plus grands, que ce soit en nombre de configurations d'espèces biochimiques différentes ou en nombre d'instances de ces configurations. Évaluer leur comportement est primordial, mais difficile. Les méthodes exactes de réduction de modèles sont utiles, mais limitées, pour ce type de modèles. Il est important de développer des méthodes numériques approchées pour les sémantiques différentielles des modèles qui permettront de trouver un encadrement garanti de l'évolution de la quantité de chaque motif d'intérêt au cours du temps, sous la forme de paires de fonctions, elles-mêmes définies comme la solution d'un système différentiel. Des travaux préliminaires ont permis d'intégrer dans un cadre formel des méthodes de troncation de développement formel [69] ou des méthodes de tropicalisation [5], tout en fournissant des bornes évoluant au cours de l'exécution des modèles sur les erreurs numériques accumulées. Il devrait également être possible de définir une version quantitative de l'analyse de flot d'information entre sites des protéines, afin de négliger les petits flots d'information, au prix d'une perte de précision dans les modèles réduits.

Cette approche devra aussi être s'appliquée au cas de la sémantique stochastique des modèles Kappa. Celle-ci décrit le comportement d'un modèle comme une distribution de traces d'exécution. Un cadre formel

pour l'exécution numériquement approchée des modèles permettra d'interfacer les sémantiques différentielles et stochastiques de Kappa pour concevoir une sémantique hybride, plus adaptée à la description des interactions entre des occurrences d'espèces biochimiques géants rares et des occurrences de petites espèces présentes en très grand nombre.

Bibliographie

[1] W. ABOU-JAOUDÉ, D. THIEFFRY et J. FERET : Formal derivation of qualitative dynamical models from biochemical networks. *Biosystems*, 149: 70–112, 2016. URL `https://doi.org/10.1016/j.biosystems.2016.09.001`.

[2] E. ALLART, J. NIEHREN et C. VERSARI : Computing difference abstractions of metabolic networks under kinetic constraints. *Actes de* L. BORTOLUSSI et G. SANGUINETTI, éds : *Computational Methods in Systems Biology - 17th International Conference, CMSB 2019, Trieste, Italy, September 18-20, 2019, Proceedings*, vol. 11773 de *Lecture Notes in Computer Science*, pp. 266–285. Springer, 2019. URL `https://doi.org/10.1007/978-3-030-31304-3_14`.

[3] J. L. ANDERSEN, C. FLAMM, D. MERKLE et P. F. STADLER : A software package for chemically inspired graph transformation. *Actes de* R. ECHAHED et M. MINAS, éds : *Graph Transformation - 9th International Conference, ICGT 2016, in Memory of Hartmut Ehrig, Held as Part of STAF 2016, Vienna, Austria, July 5-6, 2016, Proceedings*, vol. 9761 de *Lecture Notes in Computer Science*, pp. 73–88. Springer, 2016. URL `https://doi.org/10.1007/978-3-319-40530-8_5`.

[4] O. ANDREI et H. KIRCHNER : A rewriting calculus for multigraphs with ports. *Electr. Notes Theor. Comput. Sci.*, 219:67–82, 2008. URL `https://doi.org/10.1016/j.entcs.2008.10.035`.

[5] A. BEICA, J. FERET et T. PETROV : Tropical abstraction of biochemical reaction networks with guarantees. *Electr. Notes Theor. Comput. Sci.*

[6] B. BLANCHET, P. COUSOT, R. COUSOT, J. FERET, L. MAUBORGNE, A. MINÉ, D. MONNIAUX et X. RIVAL : A static analyzer for large safety-critical software. *Actes de* R. CYTRON et R. GUPTA, éds : *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation 2003, San Diego, California, USA, June 9-11, 2003*, pp. 196–207, 2003.

[7] M. L. BLINOV, J. R. FAEDER, B. GOLDSTEIN et W. S. HLAVACEK : Bionetgen : software for rule-based modeling of signal transduction

based on the interactions of molecular domains. *Bioinformatics*, 20(17), 2004.

[8] C. BODEI, L. BRODO, R. GORI, D. HERMITH et F. LEVI : A global occurrence counting analysis for brane calculi. *Actes de* M. FALASCHI, éd. : *Logic-Based Program Synthesis and Transformation - 25th International Symposium, LOPSTR 2015, Siena, Italy, July 13-15, 2015. Revised Selected Papers*, vol. 9527 de *Lecture Notes in Computer Science*, pp. 179–200. Springer, 2015. URL https://doi.org/10.1007/978-3-319-27436-2_11.

[9] C. BODEI, P. DEGANO, F. NIELSON et H. R. NIELSON : Control flow analysis for the pi-calculus. *Actes de* D. SANGIORGI et R. de SIMONE, éds : *CONCUR '98 : Concurrency Theory, 9th International Conference, Nice, France, September 8-11, 1998, Proceedings*, vol. 1466 de *Lecture Notes in Computer Science*, pp. 84–98. Springer, 1998. URL https://doi.org/10.1007/BFb0055617.

[10] P. BOUTILLIER, F. CAMPORESI, J. COQUET, J. FERET, K. Q. LÝ, N. THÉRET et P. VIGNET : Kasa : A static analyzer for kappa. *Actes de* M. CESKA et D. SAFRÁNEK, éds : *Computational Methods in Systems Biology - 16th International Conference, CMSB 2018, Brno, Czech Republic, September 12-14, 2018, Proceedings*, vol. 11095 de *Lecture Notes in Computer Science*, pp. 285–291. Springer, 2018. URL https://doi.org/10.1007/978-3-319-99429-1_17.

[11] P. BOUTILLIER, T. EHRHARD et J. KRIVINE : Incremental update for graph rewriting. *Actes de* H. YANG, éd. : *Programming Languages and Systems - 26th European Symposium on Programming, ESOP 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, vol. 10201 de *Lecture Notes in Computer Science*, pp. 201–228. Springer, 2017. URL https://doi.org/10.1007/978-3-662-54434-1_8.

[12] P. BOUTILLIER, A. FAURE DE PEBEYRE et J. FERET : Proving the absence of unbounded polymers in rule-based models. *Actes de Nine International Workshop on Static Analysis and Systems Biology (SASB'18)*, ENTCS. elsevier. to appear.

[13] P. BOUTILLIER, M. MAASHA, X. LI, H. F. MEDINA-ABARCA, J. KRIVINE, J. FERET, I. CRISTESCU, A. G. FORBES et W. FONTANA : The kappa platform for rule-based modeling. *Bioinformatics*, 34(13):i583–i592, 2018. URL https://doi.org/10.1093/bioinformatics/bty272.

[14] F. CAMPORESI : *Formal and exact reduction for differential models of signalling pathways in rule-based languages*. Thèse de doctorat, Paris Sciences et Lettres Research University, January 2017.

[15] F. CAMPORESI et J. FERET : Formal reduction for rule-based models. *Actes de* M. W. MISLOVE et J. OUAKNINE, éds : *Twenty-seventh Conference on the Mathematical Foundations of Programming Semantics, MFPS 2011, Pittsburgh, PA, USA, May 25-28, 2011*, vol. 276 de *Electronic Notes in Theoretical Computer Science*, pp. 29–59. Elsevier, 2011. doi:10.1016/j.entcs.2011.09.014. URL `https://doi.org/10.1016/j.entcs.2011.09.014`.

[16] F. CAMPORESI et J. FERET : Using alternated sums to express the occurrence number of extended patterns in site-graphs. *Actes de* J. YANG et J. A. BACHMAN, éds : *SASB 2017 - The Eighth International Workshop on Static Analysis for Systems Biology*, Static Analysis and Systems Biology, p. 18, New York, United States, août 2017. Elsevier. URL `https://hal.inria.fr/hal-01613603`. To appear.

[17] F. CAMPORESI, J. FERET et J. HAYMAN : Context-sensitive flow analyses : A hierarchy of model reductions. *Actes de* A. GUPTA et T. A. HENZINGER, éds : *Computational Methods in Systems Biology - 11th International Conference, CMSB 2013, Klosterneuburg, Austria, September 22-24, 2013. Proceedings*, vol. 8130 de *Lecture Notes in Computer Science*, pp. 220–233. Springer, 2013. doi:10.1007/978-3-642-40708-6_17. URL `https://doi.org/10.1007/978-3-642-40708-6_17`.

[18] F. CAMPORESI, J. FERET, H. KOEPPL et T. PETROV : Combining model reductions. *Actes de MFPSXXVI : Postproceedings of the 26th Conference on the Mathematical Foundations of Programming Semantics*, vol. 265 de *Electronic Notes in Theoretical Computer Science*, pp. 73–96. Elsevier Science Publishers, 2010.

[19] L. CARDELLI, M. TRIBASTONE, M. TSCHAIKOWSKI et A. VANDIN : Forward and backward bisimulations for chemical reaction networks. *Actes de 26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, pp. 226–239, 2015.

[20] L. CARDELLI, M. TRIBASTONE, M. TSCHAIKOWSKI et A. VANDIN : Symbolic computation of differential equivalences. *Theor. Comput. Sci.*, 777:132–154, 2019. doi:10.1016/j.tcs.2019.03.018. URL `https://doi.org/10.1016/j.tcs.2019.03.018`.

[21] F. CIOCCHETTA et J. HILLSTON : Bio-PEPA : A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33 – 34):3065 – 3084, 2009. Concurrent Systems Biology : To Nadia Busi (1968–2007).

[22] B. COOK, J. FISHER, E. KREPSKA et N. PITERMAN : Proving stabilization of biological systems. *Actes de* R. JHALA et D. A. SCHMIDT,

éds : *Verification, Model Checking, and Abstract Interpretation - 12th International Conference, VMCAI 2011, Austin, TX, USA, January 23-25, 2011. Proceedings*, vol. 6538, pp. 134–149. Springer, 2011. URL https://doi.org/10.1007/978-3-642-18275-4_11.

[23] P. COUSOT : The calculational design of a generic abstract interpreter. *Dans* M. BROY et R. STEINBRÜGGEN, éds : *Calculational System Design*. NATO ASI Series F. IOS Press, Amsterdam, 1999.

[24] P. COUSOT : Constructive design of a hierarchy of semantics of a transition system by abstract interpretation. *Theoretical Computer Science*, 277(1–2):47–103, 2002.

[25] P. COUSOT et R. COUSOT : Abstract interpretation : A unified lattice model for static analysis of programs by construction or approximation of fixpoints. *Actes de* R. M. GRAHAM, M. A. HARRISON et R. SETHI, éds : *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*, pp. 238–252. ACM, 1977. URL https://doi.org/10.1145/512950.512973.

[26] P. COUSOT et R. COUSOT : Systematic design of program analysis frameworks. *Actes de* A. V. AHO, S. N. ZILLES et B. K. ROSEN, éds : *Conference Record of the Sixth Annual ACM Symposium on Principles of Programming Languages, San Antonio, Texas, USA, January 1979*, pp. 269–282. ACM Press, 1979. URL https://doi.org/10.1145/567752.567778.

[27] T. C. DAMGAARD, E. HØJSGAARD et J. KRIVINE : Formal cellular machinery. *Electr. Notes Theor. Comput. Sci.*, 284:55–74, 2012. doi:10.1016/j.entcs.2012.05.015. URL https://doi.org/10.1016/j.entcs.2012.05.015.

[28] V. DANOS, J. FERET, W. FONTANA, R. HARMER, J. HAYMAN, J. KRIVINE, C. D. THOMPSON-WALSH et G. WINSKEL : Graphs, rewriting and pathway reconstruction for rule-based models. *Actes de* D. D'SOUZA, T. KAVITHA et J. RADHAKRISHNAN, éds : *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, vol. 18 de *LIPIcs*, pp. 276–288. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012. URL https://doi.org/10.4230/LIPIcs.FSTTCS.2012.276.

[29] V. DANOS, J. FERET, W. FONTANA, R. HARMER et J. KRIVINE : Rule-based modelling, symmetries, refinements. *Actes de* J. FISHER, éd. : *Formal Methods in Systems Biology, First International Workshop, FMSB 2008, Cambridge, UK, June 4-5, 2008. Proceedings*, vol. 5054 de *Lecture Notes in Computer Science*, pp. 103–122. Springer, 2008. URL https://doi.org/10.1007/978-3-540-68413-8_8.

[30] V. DANOS, J. FERET, W. FONTANA, R. HARMER et J. KRIVINE : Abstracting the differential semantics of rule-based models : Exact and automated model reduction. *Actes de Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pp. 362–381. IEEE Computer Society, 2010. URL https://doi.org/10.1109/LICS.2010.44.

[31] V. DANOS, J. FERET, W. FONTANA et J. KRIVINE : Scalable simulation of cellular signaling networks. *Actes de* Z. SHAO, éd. : *Programming Languages and Systems, 5th Asian Symposium, APLAS 2007, Singapore, November 29-December 1, 2007, Proceedings*, vol. 4807 de *Lecture Notes in Computer Science*, pp. 139–157. Springer, 2007. URL https://doi.org/10.1007/978-3-540-76637-7_10.

[32] V. DANOS, J. FERET, W. FONTANA et J. KRIVINE : Abstract interpretation of cellular signalling networks. *Actes de* F. LOGOZZO, D. A. PELED et L. D. ZUCK, éds : *Verification, Model Checking, and Abstract Interpretation, 9th International Conference, VMCAI 2008, San Francisco, USA, January 7-9, 2008, Proceedings*, vol. 4905 de *Lecture Notes in Computer Science*, pp. 83–97. Springer, 2008. URL https://doi.org/10.1007/978-3-540-78163-9.

[33] V. DANOS, R. HONORATO-ZIMMER, S. JARAMILLO-RIVERI et S. STUCKI : Rigid geometric constraints for Kappa models. *Actes de SASB'12 : PostProceedings of the 3rd International Workshop on Static Analysis and Systems Biology*, vol. 313 de *ENTCS*, pp. 23–46. Elsevier, 2015.

[34] V. DANOS et C. LANEVE : Formal molecular biology. *Theoretical Computer Science*, 325(1):69 – 110, 2004. doi:http://dx.doi.org/10.1016/j.tcs.2004.03.065. URL http://www.sciencedirect.com/science/article/pii/S0304397504002336. Computational Systems Biology.

[35] L. DEMATTÉ, C. PRIAMI et A. ROMANEL : The blenx language : A tutorial. *Actes de* M. BERNARDO, P. DEGANO et G. ZAVATTARO, éds : *Formal Methods for Computational Systems Biology : 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008 Bertinoro, Italy, June 2-7, 2008 Advanced Lectures*, pp. 313–365, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. URL http://dx.doi.org/10.1007/978-3-540-68894-5$_$9.

[36] J. R. FAEDER, M. L. BLINOV, B. GOLDSTEIN et W. S. HLAVACEK : Rule-based modeling of biochemical networks. *Complexity*, 10(4):22–41, 2005. doi:10.1002/cplx.20074. URL http://dx.doi.org/10.1002/cplx.20074.

[37] M. FÄHNDRICH et F. LOGOZZO : Static contract checking with abstract interpretation. *Actes de* B. BECKERT et C. MARCHÉ, éds : *Formal Verification of Object-Oriented Software - International Conference, FoVeOOS 2010, Paris, France, June 28-30, 2010, Revised Selected Papers*, vol. 6528 de *LNCS*, pp. 10–30. Springer, 2010.

[38] M. FEINBERG : Lectures on chemical reaction networks. Notes of lectures given at the Mathematics Research Centre, University of Wisconsin, in 1979.

[39] J. FERET : Confidentiality analysis of mobile systems. *Actes de* J. PALSBERG, éd. : *Static Analysis, 7th International Symposium, SAS 2000, Santa Barbara, CA, USA, June 29 - July 1, 2000, Proceedings*, vol. 1824 de *Lecture Notes in Computer Science*, pp. 135–154. Springer, 2000. URL https://doi.org/10.1007/978-3-540-45099-3_8.

[40] J. FERET : Occurrence counting analysis for the pi-calculus. *Electr. Notes Theor. Comput. Sci.*, 39(2):1–18, 2001. doi:10.1016/S1571-0661(05)01155-2. URL https://doi.org/10.1016/S1571-0661(05)01155-2.

[41] J. FERET : Reachability analysis of biological signalling pathways by abstract interpretation. *Actes de Proc. ICCMSE'07*, vol. 963 de *AIP*, 2007.

[42] J. FERET : An algebraic approach for inferring and using symmetries in rule-based models. *Electr. Notes Theor. Comput. Sci.*, 316:45–65, 2015. URL https://doi.org/10.1016/j.entcs.2015.06.010.

[43] J. FERET, V. DANOS, J. KRIVINE, R. HARMER et W. FONTANA : Internal coarse-graining of molecular systems. *PNAS*, 2009.

[44] J. FERET, H. KOEPPL et T. PETROV : Stochastic fragments : A framework for the exact reduction of the stochastic semantics of rule-based models. *Int. J. Software and Informatics*, 7(4):527–604, 2013. URL http://www.ijsi.org/ch/reader/view_abstract.aspx?file_no=i173.

[45] J. FERET et K. Q. LY : Local traces : An over-approximation of the behavior of the proteins in rule-based models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(4):1124–1137, July-Aug. 2018. doi:10.1109/TCBB.2018.2812195. URL doi.ieeecomputersociety.org/10.1109/TCBB.2018.2812195.

[46] J. FERET et K. Q. LÝ : Reachability analysis via orthogonal sets of patterns. *Electr. Notes Theor. Comput. Sci.*, 335:27–48, 2018. URL https://doi.org/10.1016/j.entcs.2018.03.007.

[47] M. FOLSCHETTE, L. PAULEVÉ, M. MAGNIN et O. F. ROUX : Under-approximation of reachability in multivalued asynchronous networks.

Electr. Notes Theor. Comput. Sci., 299:33–51, 2013. doi:10.1016/j.entcs.2013.11.004. URL https://doi.org/10.1016/j.entcs.2013.11.004.

[48] D. T. GILLESPIE : Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[49] R. GORI et F. LEVI : An analysis for proving temporal properties of biological systems. *Actes de* N. KOBAYASHI, éd. : *Programming Languages and Systems, 4th Asian Symposium, APLAS 2006, Sydney, Australia, November 8-10, 2006, Proceedings*, vol. 4279 de *Lecture Notes in Computer Science*, pp. 234–252. Springer, 2006. URL https://doi.org/10.1007/11924661_15.

[50] R. GROSU, G. BATT, F. H. FENTON, J. GLIMM, C. L. GUERNIC, S. A. SMOLKA et E. BARTOCCI : From cardiac cells to genetic regulatory networks. *Actes de* G. GOPALAKRISHNAN et S. QADEER, éds : *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, vol. 6806 de *Lecture Notes in Computer Science*, pp. 396–411. Springer, 2011. URL https://doi.org/10.1007/978-3-642-22110-1_31.

[51] R. HARMER, Y. L. CORNEC, S. LÉGARÉ et E. OSHURKO : Bio-curation for cellular signalling : The KAMI project. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 16(5):1562–1573, 2019. URL https://doi.org/10.1109/TCBB.2019.2906164.

[52] R. HARMER, V. DANOS, J. FERET, J. KRIVINE et W. FONTANA : Intrinsic information carriers in combinatorial dynamical systems. *Chaos*, 20, September 2010. URL http://link.aip.org/link/CHAOEH/v20/i3/p037108/s1.

[53] M. HEINER et I. KOCH : Petri net based model validation in systems biology. *Dans* J. CORTADELLA et W. REISIG, éds : *Applications and Theory of Petri Nets 2004 : 25th International Conference, ICATPN 2004, Bologna, Italy, June 21–25, 2004. Proceedings*, pp. 216–237. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. URL http://dx.doi.org/10.1007/978-3-540-27793-4$_$13.

[54] T. HELMS, T. WARNKE, C. MAUS et A. M. UHRMACHER : Semantics and efficient simulation algorithms of an expressive multilevel modeling language. *ACM Trans. Model. Comput. Simul.*, 27(2):8 :1–8 :25, 2017. doi:10.1145/2998499. URL https://doi.org/10.1145/2998499.

[55] A. HUSSON : *Logical foundations of a modelling assistant for molecular biology*. Thèse de doctorat, Université de Paris, France, 2019. URL http://ahusson.fr/husson_manuscript_electronic.pdf.

[56] E. L. INCE : *Ordinary Differential Equations*. Dover Publications, Inc., 1956.

[57] M. JOHN, C. LHOUSSAINE, J. NIEHREN et C. VERSARI : Biochemical reaction rules with constraints. *Actes de Programming Languages and Systems - 20th European Symposium on Programming, ESOP 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings*, vol. 6602 de *Lecture Notes in Computer Science*, pp. 338–357. Springer, 2011. URL http://dx.doi.org/10.1007/978-3-642-19718-5.

[58] M. JOHN, M. NEBUT et J. NIEHREN : Knockout prediction for reaction networks with partial kinetic information. *Actes de* R. GIACOBAZZI, J. BERDINE et I. MASTROENI, éds : *Verification, Model Checking, and Abstract Interpretation, 14th International Conference, VMCAI 2013, Rome, Italy, January 20-22, 2013. Proceedings*, vol. 7737 de *Lecture Notes in Computer Science*, pp. 355–374. Springer, 2013. URL https://doi.org/10.1007/978-3-642-35873-9_22.

[59] O. KAHRAMANOGULLARI et L. CARDELLI : An intuitive modelling interface for systems biology. *Int. J. Software and Informatics*, 7(4):655–674, 2013. URL http://www.ijsi.org/ch/reader/view_abstract.aspx?file_no=i177.

[60] H. KLARNER, A. BOCKMAYR et H. SIEBERT : Computing maximal and minimal trap spaces of boolean networks. *Natural Computing*, 14(4):535–544, 2015. doi:10.1007/s11047-015-9520-7. URL https://doi.org/10.1007/s11047-015-9520-7.

[61] A. KÖHLER, J. KRIVINE et J. VIDMAR : A rule-based model of base excision repair. *Actes de* P. MENDES, J. O. DADA et K. SMALLBONE, éds : *Computational Methods in Systems Biology - 12th International Conference, CMSB 2014, Manchester, UK, November 17-19, 2014, Proceedings*, vol. 8859 de *Lecture Notes in Computer Science*, pp. 173–195. Springer, 2014. URL https://doi.org/10.1007/978-3-319-12982-2_13.

[62] J. KOLCÁK, D. SAFRÁNEK, S. HAAR et L. PAULEVÉ : Parameter space abstraction and unfolding semantics of discrete regulatory networks. *Theor. Comput. Sci.*, 765:120–144, 2019. doi:10.1016/j.tcs.2018.03.009. URL https://doi.org/10.1016/j.tcs.2018.03.009.

[63] M. Z. KWIATKOWSKA, G. NORMAN et D. PARKER : PRISM 4.0 : Verification of probabilistic real-time systems. *Actes de* G. GOPALAKRISHNAN et S. QADEER, éds : *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, vol. 6806 de *Lecture Notes in Computer Science*, pp. 585–591. Springer, 2011. URL https://doi.org/10.1007/978-3-642-22110-1_47.

[64] E. MURPHY, V. DANOS, J. FERET, J. KRIVINE et R. HARMER : *Elements of Computational Systems Biology*, chap. Rule Based Modelling and Model Refinement. Wiley Book Series on Bioinformatics. John Wiley & Sons, Inc., 2010.

[65] H. R. NIELSON et F. NIELSON : Shape analysis for mobile ambients. *Actes de* M. N. WEGMAN et T. W. REPS, éds : *POPL 2000, Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Boston, Massachusetts, USA, January 19-21, 2000*, pp. 142–154. ACM, 2000. URL `https://doi.org/10.1145/325694.325711`.

[66] L. PAULEVÉ, M. MAGNIN et O. F. ROUX : Abstract interpretation of dynamics of biological regulatory networks. *Electr. Notes Theor. Comput. Sci.*, 272:43–56, 2011. URL `https://doi.org/10.1016/j.entcs.2011.04.004`.

[67] T. PETROV, J. FERET et H. KOEPPL : Reconstructing species-based dynamics from reduced stochastic rule-based models. *Actes de* O. ROSE et A. M. UHRMACHER, éds : *Winter Simulation Conference, WSC '12, Berlin, Germany, December 9-12, 2012*, pp. 225 :1–225 :15. WSC, 2012. URL `https://doi.org/10.1109/WSC.2012.6465241`.

[68] A. REGEV, W. SILVERMAN et E. SHAPIRO : Representation and simulation of biochemical processes using the pi-calculus process algebra. *Actes de* R. B. ALTMAN, A. K. DUNKER, L. HUNTER et T. E. KLEIN, éds : *Pacific Symposium on Biocomputing, Volume 6*, pp. 459–470, Singapore, 2001.

[69] K. C. SAINT-GERMAIN et J. FERET : Conservative numerical approximations of the differential semantics in biological rule- based models, 2016. Master thesis.

[70] D. STEWART : Spatial biomodelling, 2010. Master thesis, School of Informatics, University of Edinburgh.

[71] A. TARSKI : A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.*, 5(2), 1955.