

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Contexte et motivations . . . . .	1
1.2	Les langages de modélisation de systèmes d'interactions moléculaires . . . . .	1
1.3	Le langage Kappa . . . . .	2
1.4	Interprétation abstraite . . . . .	4
1.5	L'écosystème Kappa . . . . .	5
1.5.1	Analyses statiques . . . . .	5
1.5.2	Analyses causales . . . . .	6
1.5.3	Réduction de modèles . . . . .	9
1.6	Contributions . . . . .	10
<b>2</b>	<b>Graphes à sites</b>	<b>13</b>
2.1	Signature . . . . .	13
2.2	Complexes biochimiques . . . . .	14
2.3	Motifs . . . . .	15
2.4	Plongements entre motifs . . . . .	16
<b>3</b>	<b>Réécriture de graphes à sites</b>	<b>19</b>
3.1	Règles d'interaction . . . . .	19
3.2	Réactions induites par une règle d'interaction . . . . .	21
3.3	Réseaux de réactions sous-jacents . . . . .	21
<b>4</b>	<b>Analyse des motifs accessibles</b>	<b>25</b>
4.1	Accessibilité dans un réseau réactionnel . . . . .	25
4.2	Abstraction d'un ensemble d'états . . . . .	27
4.3	Transferts de point-fixes . . . . .	30
4.4	Analyse par ensembles de motifs orthogonaux . . . . .	31
4.4.1	Ensembles de motifs orthogonaux . . . . .	33
4.4.2	Pas de calcul abstraits . . . . .	33
4.4.3	Post-traitement et visualisation des résultats . . . . .	35
4.5	Étude de performance et utilisation concrète . . . . .	37
<b>5</b>	<b>Flot d'information dans la sémantique stochastique d'un modèle Kappa</b>	<b>41</b>
5.1	Système stochastique sous-jacent à un modèle Kappa . . . . .	41
5.2	Cas d'études . . . . .	41
5.3	Sémantique stochastique d'un modèle de réécritures de graphes à sites . . . . .	41
5.4	Cas d'étude . . . . .	41
5.4.1	Un exemple d'indépendance entre deux liaisons . . . . .	42
5.4.2	Un exemple avec une déliaison inconditionnelle . . . . .	43
5.4.3	An example of distant control . . . . .	48
5.5	Conclusion . . . . .	50

<b>6</b>	<b>Symétrie dans les graphes à sites</b>	<b>53</b>
6.1	Introduction . . . . .	54
6.1.1	Un modèle avec des sites équivalents . . . . .	54
6.1.2	sémantique . . . . .	56
6.1.3	Symétries et propriétés comportementales . . . . .	56
6.1.4	Symetries and model reduction . . . . .	66
6.2	Group actions over rule-based models . . . . .	66
6.2.1	Group actions over site graphs . . . . .	66
6.2.2	Induced group actions . . . . .	74
6.3	symetries in rule-based models . . . . .	101
6.3.1	symetries among a site graph . . . . .	102
6.3.2	symetries among a set of rules . . . . .	104
6.4	Permutations of sites in site graphs . . . . .	106
6.4.1	Formalisation . . . . .	106
6.4.2	Retour sur le cas d'étude . . . . .	112
<b>7</b>	<b>Conclusion</b>	<b>115</b>
7.1	Bibliography . . . . .	116

Analyses des motifs accessibles dans les modèles Kappa Jérôme FERET

<sup>1</sup>Département d'informatique de l'ÉNS, ÉNS, CNRS, Université PSL, Paris, France

<sup>2</sup>INRIA, Paris, France

# Chapter 1

## Introduction

### 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'interaction, est influencé par des compétitions pour des ressources communes. De plus, les occurrences de protéines peuvent former une grande diversité de complexes biochimiques différents. La concurrence entre des interactions à différentes échelles de temps génère des boucles de rétro-actions non linéaires qui contrôlent l'abondance de ces complexes biochimiques. Enfin, ces systèmes font intervenir des interactions entre de très petites molécules, comme des ions ou des ligands et des complexes 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 dans leur ensemble 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.

### 1.2 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 (Feinberg) ou les réseaux de Petri classiques (Heiner et Koch 2004), 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 complexes biomoléculaires.

Des langages de plus haut niveau, inspirés des différents paradigmes de programmation, tels que les tableaux d'états à messages (Damm et Harel 2001), les automates communicants (Plateau 1985), les algèbres de processus (Regev et collab. 2001 ; Ciocchetta et Hillston 2009), les langages orientés objet (Dematté et collab. 2008), les réseaux de Petri colorés (Gao et collab. 2011) et la réécriture de graphes à sites (Danos et Laneve 2004 ; Faeder

et collab. 2005 ; Andrei et Kirchner 2008 ; John et collab. 2011a), exploitent le fait que les interactions dépendent généralement de conditions locales sur les configurations des occurrences de protéines au sein des complexes 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 pour décrire cette interaction.

Il est important de distinguer les approches basées sur les agents de celles basées sur les règles de réécriture. Dans les approches basées sur les agents, chaque entité, que ce soit un processus (Ciocchetta et Hillston 2009) ou un objet (Dematté et collab. 2008), 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  $\pi$ -calcul (Regev et collab. 2001). Cependant, en général, les approches basées sur les agents donnent lieu à des systèmes de processus à états finis (Kahramanogullari et Cardelli 2013). Ceci permet d’étudier leur comportement à l’aide d’outils de vérification symbolique de modèles comme PRISM (Kwiatkowska et collab. 2011).

Lorsque les occurrences des protéines admettent trop de configurations différentes ou lorsque leurs capacités d’interaction 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.

Dans les approches fondées sur les règles, les modèles sont définis par des règles d’interaction. 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 toutes les capacités d’interaction 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’interaction des occurrences des protéines, elles encouragent à une modélisation sans *a priori* où les interactions émergent des règles lors de la conception du modèle.

Le calcul des ambients (Cardelli et Gordon 1998, 2000), des bioambients (Regev et collab. 2004) et celui des membranes (Cardelli 2004) sont un peu particuliers. Ils permettent de décrire des boîtes ou des compartiments, qui peuvent être arbitrairement imbriqués au sein d’une arborescence, alors que des agents, contenus dans les boîtes dans le cas des ambients, ou dans leurs parois dans le cas des membranes, permettent à ces compartiments de se déplacer ou de se fusionner. Les capacités d’interaction des agents peuvent alors dépendre de leur localisation dans la hiérarchie des compartiments. La calcul projectif des membranes (Danos et Pradalier 2004) représente plus fidèlement la disposition des compartiments au sein d’une cellule, en rendant la description de l’état du système indépendante du choix de la racine de l’arborescence des compartiments.

## 1.3 Le langage Kappa

Les langages de réécriture de graphes à sites (Danos et Laneve 2004 ; Faeder et collab. 2005 ; Andrei et Kirchner 2008 ; John et collab. 2011a) permettent de représenter de manière transparente les réseaux d’interactions entre des occurrences de protéines grâce à leur syntaxe qui est fortement inspirée de la chimie.

Dans Kappa, chaque complexe biochimique est représenté par un graphe à sites. Un exemple de graphe à sites est donné en Fig. 1.1(a). Dans un graphe à sites, des nœuds qui représentent des occurrences de protéines, sont associés à une liste de sites d’interaction. 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’interaction. Dans Kappa, la structure tri-dimensionnelle des occurrences de protéines n’est pas représentée explicitement. En revanche, les conditions pour qu’un site d’interaction 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 Fig. 1.1(b) est dessinée une règle pour la formation de dimers. Deux récepteurs (*EGFR*) qui sont tous deux liés à des ligands (*EGF*) peuvent se lier entre eux pour former un dimer. En Fig. 1.1(c) 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

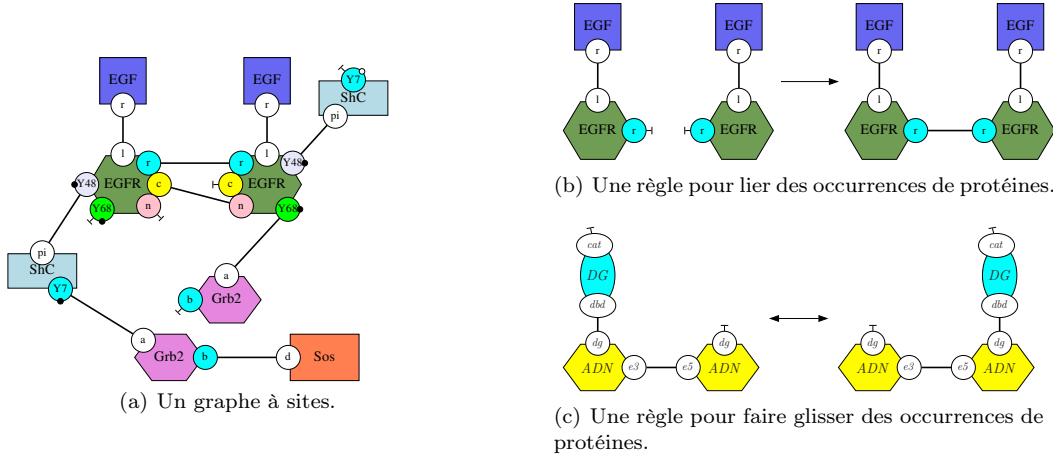


Figure 1.1: En 1.1(a) est dessiné un graphe à site. Il s'agit d'un complexe biochimique composé de deux occurrences du ligand (*EGF*), de deux occurrences du récepteur membranaire (*EGFR*), d'une occurrence de la protéine d'échafaudage (*Shc*), de deux occurrences de la protéine de transport (*Grb2*) et d'une occurrence de la protéine *Sos*. En 1.1(b) est donné un exemple de règle de liaison. Deux occurrences du récepteurs membranaires (*EGFR*), lorsqu'ils sont tous 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 1.1(c) 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.

les deux sens, le long d'un brin d'ADN (Köhler et collab. 2014).

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 de temps 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 (Gillespie 1977)). 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 (Danos et collab. 2007b). 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 (Boutillier et collab. 2017).

Le langage Kappa souffre de plusieurs limites. Par exemple, dans Kappa, les sites d'interaction 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 efficace, 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 (Faeder et collab. 2005) et mØd (Andersen et collab. 2016), soit indirectement en utilisant un codage sous forme d'hyperliens comme c'est possible dans le langage React(C) (John et collab. 2011a). 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'interaction dans les complexes biochimiques afin de restreindre l'ensemble des événements possibles à ceux qui satisfont ces contraintes (Danos et collab. 2015). 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 le même complexe 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 (Stewart 2010) permet d'utiliser ce procédé de manière transparente. Par ailleurs, le langage ML (Helms et collab. 2017) 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 (Damgaard et collab. 2012) 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 sont toutefois nécessaires pour calculer les propriétés du comportement collectif des populations de protéines.

## 1.4 Interprétation abstraite

L'interprétation abstraite a été introduite il y a maintenant 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 a été utilisée non seulement pour comparer différentes méthodes et algorithmes d'analyse statique (Cousot 2002), mais aussi pour développer des analyseurs statiques qui peuvent calculer automatiquement les propriétés sur le comportement des programmes (Blanchet et collab. 2003 ; Fähndrich et Logozzo 2010). 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  $\text{lfp } \mathbb{F}$  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 être représenté en langage mathématique sous 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  $D^\#$  de propriétés d'intérêt. C'est le domaine abstrait. Cet ensemble est ordonné par un ordre partiel  $\sqsubseteq$ . Chaque élément  $a^\#$  de ce domaine abstrait représente intentionnellement l'ensemble des éléments concrets qui satisfont cette propriété. Cet ensemble est noté  $\gamma(a^\#)$ . La fonction  $\gamma$ , ainsi définie, est croissante (si  $a^\# \sqsubseteq b^\#$ , alors  $\gamma(a^\#) \subseteq \gamma(b^\#)$ ). Ainsi, l'ordre  $\sqsubseteq$  représente le niveau d'information.

Un élément abstrait  $a^\#$  est dit être une abstraction d'un ensemble  $a$  d'éléments concrets, si et seulement si  $a$  est un sous-ensemble de l'ensemble  $\gamma(a^\#)$ . 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  $\mathbb{F}^\#$  opérant sur le domaine abstrait  $D^\#$  et telle que  $[\alpha \circ \mathbb{F} \circ \gamma](a^\#) \sqsubseteq \mathbb{F}^\#(a^\#)$  pour

chaque élément abstrait  $a^\sharp \in D^\sharp$ , admet un plus petit point fixe (pour l'ordre  $\sqsubseteq$ ) noté  $\text{lfp } \mathbb{F}^\sharp$ . 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 naturellement été inspirées par les analyses de flot d'information (Bodei et collab. 1998 ; Feret 2000) et de dénombrement (Nielson et Nielson 2000 ; Feret 2001) dans le  $\pi$ -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 (Gori et Levi 2006 ; Bodei et collab. 2015). 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 complexes biochimiques n'est pas représentée explicitement (Abou-Jaoudé et collab. 2015, 2016). 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 (Paulevé et collab. 2011), puis plus généralement pour les réseaux asynchrones discrets booléens ou multivalués (Folschette et collab. 2013). Dans les modèles réseaux booléens ou multivalués, l'interprétation abstraite a également été utilisée pour calculer une approximation des ensembles constituant des trappes (Cook et collab. 2011 ; Klarner et collab. 2015), 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 (John et collab. 2013 ; Allart et collab. 2019).

L'interprétation abstraite peut servir à la calibration d'un modèle (Kolcák et collab. 2019), 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 (Grosu et collab. 2011).

## 1.5 L'écosystème Kappa

Plusieurs outils pour analyser et manipuler des modèles Kappa sont présentés ici.

### 1.5.1 Analyses statiques

Un outil d'analyse statique (Boutillier et collab. 2018a), basé sur le cadre de l'interprétation abstraite, permet de calculer automatiquement certaines propriétés des modèles. Le but est d'améliorer la confiance dans les règles qui constituent le modèle. Il s'agit de retrouver des propriétés d'intérêt que le modélisateur pouvait, ou non, avoir en tête lors de la conception de son modèle ou bien de trouver des erreurs dans la modélisation.

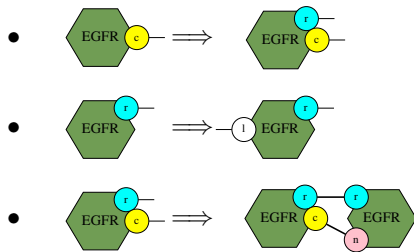
Cette analyse utilise un ensemble de motifs d'intérêt. Parmi ces motifs, l'analyse prouve que certains ne peuvent apparaître dans aucuns états potentiels du modèle. Les autres sont déclarés potentiellement accessibles : soit ils le sont effectivement, soit c'est une conséquence de la sur-approximation de l'analyse.

Les motifs d'intérêt permettent de poser les questions intéressantes sur la structure biochimique des occurrences des complexes lors de l'exécution du modèle. Actuellement, l'analyse pose trois types de questions :

Existe-t-il une relation entre l'état de plusieurs sites dans les occurrences d'une protéine ? Lorsque deux occurrences de protéines sont liées entre-elles, existe-t-il une relation entre l'état de leurs sites respectifs ? Est-ce qu'une occurrence d'une protéine peut être doublement liée à une autre occurrence d'une protéine, est-ce qu'une occurrence de protéines peut être liée à des occurrences différentes d'un même type de protéines ? La première catégorie est une analyse relationnelle classique. Elle permet, par exemple, de détecter si un site ne peut être lié sans qu'un autre ne le soit ou de détecter si un site ne peut être lié sans être phosphorylé. La seconde est utile quand des sites fictifs permettent d'encoder la localisation des occurrences de protéines, il est alors possible de vérifier, chaque fois que deux occurrences de protéines sont liées, si elles se situent nécessairement dans un même compartiment. Enfin, la troisième analyse la formation de doubles liaisons entre les occurrences de protéines. Le choix exact des questions posées par l'analyseur est fixé automatiquement suite à une inspection statique des règles du modèle.

Le résultat final de l'analyse d'accessibilité est présenté à l'utilisateur sous deux formes. D'une part, les règles dont le membre gauche est en contradiction avec les motifs qui ont été prouvés inaccessibles par l'analyse sont mentionnées à l'utilisateur. D'autre part, les propriétés intéressantes sur la structure des complexes biochimiques sont listées sous la forme de lemmes de raffinement.

Par exemple, les trois lemmes suivants :



informent l'utilisateur que (pour le premier) dans une occurrence du récepteur membranaire, le site *c* ne peut être lié sans que le site *r* ne le soit également, que (pour le second) le site *r* ne peut être lié sans que le site *l* ne le soit aussi, et que (pour le troisième) quand une occurrence du récepteur membranaire a ses sites *r* et *c* tous deux liés, ils sont nécessairement liés tous deux à une même occurrence du récepteur membranaire.

Un lemme de raffinement est ainsi présenté comme une implication entre un motif et une liste de motifs. Ici, les listes de motifs sont toutes réduites à un élément. Il faut interpréter une telle implication par le fait que toute occurrence du membre gauche de l'implication dans un état accessible peut se raffiner dans au moins un des motifs du membre droit.

Lorsque l'utilisateur obtient des propriétés auxquelles il ne s'attend pas, il doit retourner à son modèle pour comprendre l'origine du problème. Les erreurs typographiques sont assez courantes. Il arrive aussi souvent que certaines parties du modèle manquent, il faut aller les compléter ou les remplacer par des règles fictives si l'information n'est pas disponible dans la littérature. Il se peut aussi que l'état initial du modèle ait été mal choisi. Enfin, les erreurs peuvent aussi être dues à des relations causales complexes. L'analyse statique peut alors être complétée par l'analyse causale pour comprendre comment les configurations inattendues se produisent.

### 1.5.2 Analyses causales

La causalité est un outil très utile pour comprendre le comportement individuel des occurrences de protéines dans un modèle Kappa. Son but est d'étudier en quoi certains événements ont été nécessaires pour que d'autres événements aient pu avoir lieu.

Une trace causale est alors un ensemble d'événements dont certaines paires sont ordonnées par une relation de causalité. Celle-ci indique si l'application d'un événement a rendu possible l'application d'un autre. Un exemple de trace causale est donné en Fig. 1.2. Il s'agit de l'ensemble des événements pour qu'une occurrence du récepteur membranaire recrute une occurrence de la protéine *Sos* par le biais de son site *Y68*. Il faut tout d'abord activer deux occurrences du récepteur membranaire en les liant à des occurrences du ligand *EGF*. Les deux occurrences du récepteur peuvent alors établir une liaison symétrique, puis une liaison asymétrique ce qui permet de différencier une des deux occurrences du récepteur membranaire. Le site *Y68* de cette occurrence peut alors être phosphorylé pour qu'il puisse se lier à une occurrence de la protéine de transport *Grb2*. Indépendamment, cette occurrence de la protéine de transport peut s'être liée à une occurrence de la protéine *Sos*.



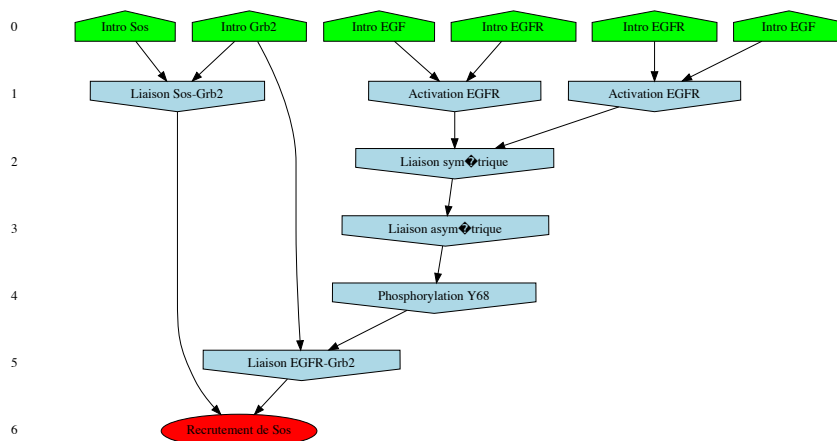


Figure 1.2: Une des deux traces causales pour le recrutement d’une occurrence de la protéine *Sos* par une occurrence du récepteur membranaire. Les nœuds verts représentent l’introduction des occurrences de protéines, les nœuds bleus représentent l’application des règles, le nœud rouge représente le but à observer. Les arcs décrivent les relations causales entre les événements.

Dans cette trace, tous les événements sont nécessaires, mais d’autres *scenarii* peuvent exister. Par exemple, les occurrences du récepteur membranaire peuvent recruter une occurrence de la protéine *Sos* par le biais du site Y48, ce qui donne lieu à une autre trace causale. Une trace causale décrit, en fait, un ensemble d’événements qui sont nécessaires dans un scénario potentiel.

Les traces causales sont obtenues en partant des résultats de la simulation, en relevant toutes les occurrences d’un événement d’intérêt. Pour chaque occurrence, une trace est extraite en collectant les événements nécessaires à cette occurrence ou récursivement à tout autre événement lui-même nécessaire (Danos et collab. 2007a). Les événements sont ensuite organisés sous la forme d’un graphe acyclique orienté grâce à la transformation de Mazurkiewicz (Mazurkiewicz 1984). Cette transformation exploite le fait que certains événements, causalement indépendants, commutent. Un moteur de recherche opérationnelle est ensuite utilisé pour retirer de cette trace causale les événements qui peuvent l’être. Une description de cette approche dans un formalisme catégorique est décrit dans cette publication (Danos et collab. 2012).

Les traces causales donnent une vision des voies de signalisation qui privilégie l’acquisition du signal. Dans un modèle, toutes les interactions sont en général réversibles, ce qui est nécessaire pour que l’occurrence d’une kinase, par exemple, puisse agir sur plusieurs occurrences de sa protéine cible à tour de rôle. Cet aspect, gestion de ressources, n’est pas du tout décrit dans les traces causales. Les traces causales ne peuvent donc pas remplacer les règles d’un modèle. Il s’agit juste d’un outil pour comprendre comment un objectif peut être atteint, mais qui ne permet pas à lui seul de définir le comportement collectif du modèle.

Les traces causales dépendent fortement de la syntaxe du langage. En effet, la syntaxe définit quelles préconditions peuvent être utilisées dans les règles, ce qui a une incidence sur le fait que deux événements puissent être vus ou non, comme indépendants causalement. Aussi, le fait que Kappa utilise de la réécriture hors contexte où seuls les sites qui ont une importance dans une interaction ont besoin d’être mentionnés, permet d’avoir plus d’événements qui commutent. Chaque trace causale peut alors résumer un plus grand nombre de traces classiques.

En Fig. 1.3 est considéré l’exemple d’une sorte de protéines avec deux sites de phosphorylation. Chaque site peut être phosphorylé indépendamment de l’état de l’autre site, ce qui se traduit en Kappa par les deux règles données en Fig. 1.3(a). Ces règles peuvent être appliquées dans n’importe quel ordre. Il y a donc une seule trace causale pour obtenir une occurrence de protéines doublement phosphorylée. Cette trace est dessinée en Fig. 1.3(e). Dans un réseau réactionnel, les complexes sont nommés et leur structure biochimique ne peut pas être utilisée. Il faut donc quatre réactions pour simuler ces deux règles Kappa. Or, chacune de ces réactions spécifie exactement quel réactif elle utilise, ce qui empêche les réactions de commuter. Il y a alors deux traces causales différentes selon que le site de droite ou de gauche ait été phosphorylé en premier.

Pour conclure sur la causalité, il est important de remarquer que les traces causales s’appuient sur une

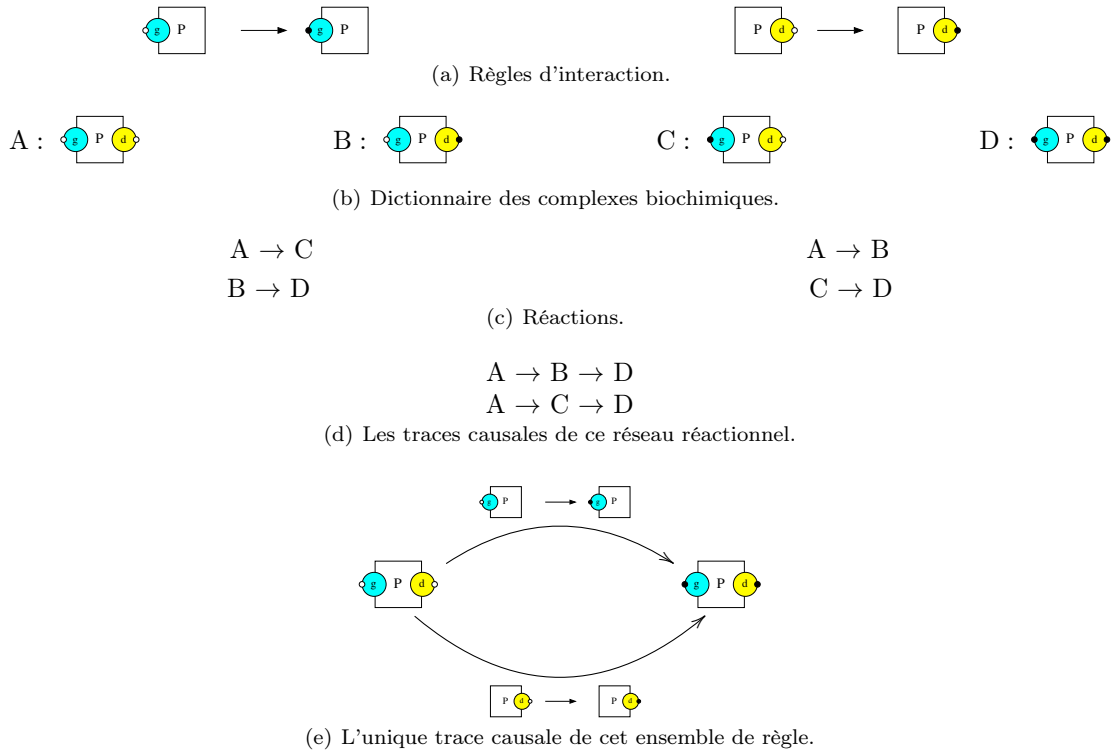


Figure 1.3: En 1.3(a), un modèle formé de deux règles d'interaction. En 1.3(b) l'ensemble des toutes les sortes de complexes biochimiques accessibles à partir d'une occurrence de la protéine entièrement non phosphorylée, un nom est donné à chaque sorte de complexe biochimique. En Fig. 1.3(c), le réseau réactionnel sous-jacent. Contrairement aux règles d'interaction, les réactions testent l'intégralité de l'état de l'occurrence de la protéine. Ainsi, les réactions qui phosphorylent les deux sites ne commutent pas. Il y a donc deux traces causales, selon que le site droit ou gauche ait été phosphorylé en premier avec les réactions (Fig. 1.3(d)). En Kappa, les règles de phosphorylation d'un site s'appliquent quelque soit l'état de l'autre site. Ainsi les traces causales ne distinguent pas quel site est phosphorylé en premier. Il n'y a alors qu'un seul type de trace causale (Fig. 1.3(e)).

vision positive de la causalité. Ce n'est en général pas suffisant pour comprendre le comportement des voies de signalisation intracellulaires. En effet, il y a souvent dans ces voies des événements qui ne sont certes pas nécessaires mais qui rendent d'autres événements plus probables. C'est le cas d'une interaction qui stabiliserait une structure instable pour lui laisser le temps de réaliser une certaine interaction. D'un point de vue logique, la stabilisation de la structure n'est pas requise. Mais il est improbable que sans elle, l'autre interaction puisse avoir lieu. Ces effets cinétiques sont capturés par les notions de causalité contre-factuelles (Halpern et Pearl 2013), dont l'adaptation à Kappa (Laurent et collab. 2018) ouvre des pistes de recherches pleines de promesses.

### 1.5.3 Réduction de modèles

La réduction de modèles consiste à simplifier un modèle en ajustant le grain d'observation. Les réductions de modèles peuvent se formaliser comme des transformations de graphes (Gay et collab. 2014), des transformations tropicales (Radulescu et collab. 2015), des bisimulations (Camporesi et collab. 2010 ; Cardelli et collab. 2015), ou, tout simplement, des changements de variables (Feret et collab. 2009). Elles peuvent être classées selon la classe de propriétés qu'elles préservent.

Des outils de réduction exacte permettent de simplifier à la fois les systèmes d'équations différentielles (Feret et collab. 2009 ; Danos et collab. 2010) et les systèmes stochastiques (Feret et collab. 2013) qui sont décrits en Kappa. Ces algorithmes trouvent automatiquement des changements de variables par inspection statique des règles initiales des modèles et dérivent des modèles réduits en conséquence. La preuve de correction de ces algorithmes est faite par interprétation abstraite : le modèle réduit définit la projection exacte, par le changement de variables découvert par l'analyse, du comportement transitoire du modèle avant réduction. L'ensemble des complexes biochimiques, le changement de variables et la description extensionnelle du modèle avant réduction ne sont jamais représentés explicitement, ce qui permet à la méthode de passer à l'échelle.

Les outils de réduction de modèles pour Kappa combinent deux types d'abstraction : le premier exploite les symétries potentielles au sein des sites d'interaction des occurrences des protéines du modèle, alors que le second identifie parmi les corrélations éventuelles entre les états des sites des occurrences des protéines, celles qui n'ont aucun impact sur leur comportement collectif. Les symétries sont décrites comme des actions de groupes qui préservent l'ensemble des règles de réécriture qui constituent un modèle (Camporesi et collab. 2010 ; Feret 2015). Elles induisent une relation d'équivalence entre les complexes biochimiques qui, elle-même, définit une relation de bisimulation sur les différents états du modèle. Les états en relation seront regroupés en un seul dans le modèle réduit. Intuitivement, cette analyse détecte quels sites ont exactement les mêmes capacités d'interaction et ignore la différence entre ces sites : dans le modèle réduit, la configuration d'une occurrence d'une protéine est définie par le nombre de sites dans un certain état en faisant abstraction de quels sites précis sont dans cet état. Ceci engendre une réduction d'un facteur exponentiel : par exemple, pour un type de protéines avec  $n$  sites symétriques pouvant chacun prendre deux états différents, la réduction permet de passer de  $2^n$  configurations potentielles à seulement  $(n + 1)$ .

La deuxième approche se base sur l'analyse du flot d'information entre les différents sites d'interaction des complexes biochimiques. Cela permet de comprendre quelles corrélations entre l'état des différents sites peuvent avoir une influence sur le comportement global du système et de passer les autres sous silence. Une approximation qualitative du flot d'information est calculée en répertoriant, au sein des règles de réécriture, tous les chemins entre les sites dont l'état est testé (ceux qui apparaissent dans le membre gauche d'une règle) et les sites dont l'état est modifié (ceux qui apparaissent dans le membre droit de cette règle avec un état différent de celui du membre gauche) (voir en Fig. 1.4(a)). Chaque motif est alors annoté en regroupant le flot d'information présent dans chacune des règles qui peut s'y appliquer. Les motifs intéressants sont ceux pour lesquels il existe un site d'interaction qui est accessible par tous les autres en suivant cette annotation. Par exemple, le complexe biochimique dessiné en Fig. 1.1(a) contient les quatre motifs d'intérêt donnés en Fig. 1.4(b) avec leur annotation. Dans ce modèle, les motifs d'intérêt sont exactement ceux qui décrivent l'état d'un seul site  $Y48$  ou  $Y68$ . Ainsi la corrélation entre l'état des différents sites  $Y48$  et  $Y68$  n'est plus représentée dans le modèle réduit. D'un point de vue combinatoire, ceci permet de passer de  $m^2 \cdot n^2$  complexes biochimiques à  $m + n$  motifs d'intérêt (où  $m$  et  $n$  représentent respectivement le nombre de configurations différentes pour la partie du complexe liée aux sites  $Y48$  et  $Y68$ ).

Sur un modèle plus complet (Blinov et collab. 2006 ; Schoeberl et collab. 2002 ; Brightman et Fell 2000 ; Danos et collab. 2007a), cet outil permet de passer de  $10^{20}$  complexes biochimiques à 175,000 motifs d'intérêt, en moins de 10 minutes.

Des méthodes approchées utilisent des formes tronquées de développement formels de la sémantique stochastique (Gillespie 2009), alors que les méthodes de tropicalisation exploitent la séparation entre les échelles de

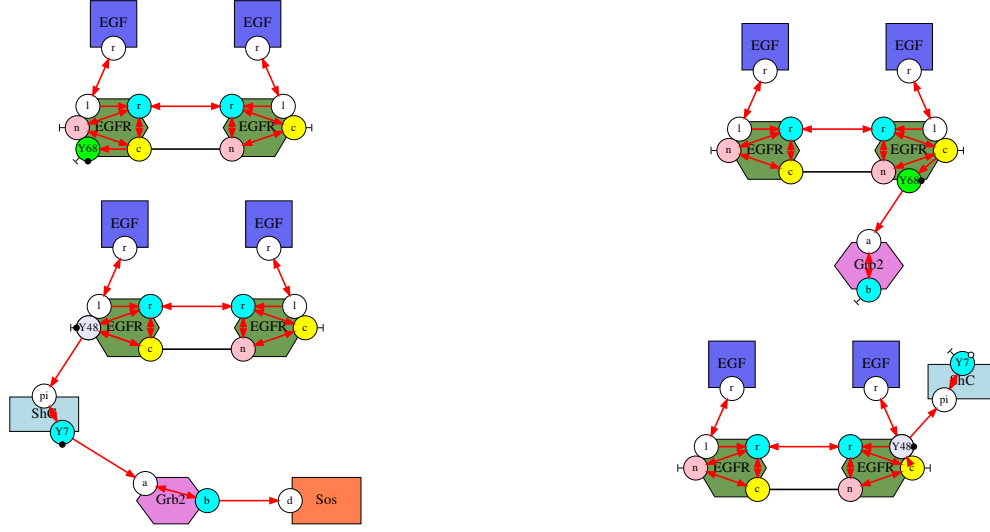
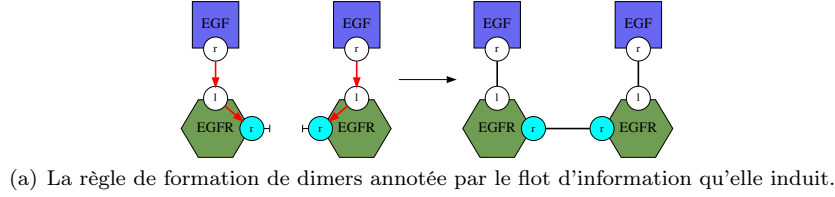


Figure 1.4: En 1.4(a), chaque chemin entre un site dont l'état est testé et un site dont l'état est modifié dans une composante connexe du membre gauche d'une règle induit un flot d'information. Ici, la capacité de lier le site  $r$  d'une occurrence du récepteur dépend du fait que cette occurrence soit liée à une occurrence du ligand. En 1.4(b) sont représentés les quatre motifs d'intérêt qui apparaissent dans le complexe biochimique dessiné en Fig. 1.1(a). Ils sont tous quatre annotés par une relation qui spécifie comment l'information se propage – ou s'est propagée – à travers leurs différents sites d'interaction (cette relation est obtenue en recopiant le flot d'information des règles compatibles avec ces motifs). Ils contiennent chacun un site accessible par tous les autres en suivant cette relation.

temps et de concentration (Radulescu et collab. 2012). Ces méthodes ne procurent pas de bornes d'erreur explicites. Par ailleurs, elles nécessitent une description extensionnelle des réseaux réactionnels sous-jacents.

Des méthodes exactes opèrent de manière analytique pour extraire des relations d'équivalence entre les complexes biochimiques de la description explicite des réseaux réactionnels (Cardelli et collab. 2015) ou même directement sur des systèmes d'équations différentielles (Cardelli et collab. 2019). Elles permettent de calculer la meilleure bisimulation en avant, parmi celles qui sont basées sur un partitionnement des variables, et quelles variables prennent toujours la même valeur. La notion de symétries développée pour Kappa est plus restrictive car elle se concentre sur les bisimulations qui correspondent à un certain groupe de transformations. En revanche, elle permet de détecter des relations de proportionnalité entre variables. Par ailleurs, elle ne nécessite de représenter, ni les réseaux réactionnels, ni les systèmes différentiels sous-jacents, évitant ainsi un calcul dont la durée est souvent prohibitive (Camporesi et collab. 2017).

La réduction de modèles basée sur l'étude du flot d'information est à la fois une généralisation et une formalisation d'approches systématiques existantes (Borisov et collab. 2005 ; Conzelmann et collab. 2006). L'utilisation d'un langage formel et l'interprétation abstraite de sa sémantique a permis d'établir formellement la correction de ces approches.

## 1.6 Contributions

Le reste de ce document décrit le langage Kappa (Danos et Laneve 2003, 2004) sous forme graphique, ainsi que l'analyse statique qui permet de détecter quels motifs peuvent se former lors de l'exécution des modèles (Danos

et collab. 2008 ; Feret et Lý 2018).

En particulier, la notion de graphe à sites, qui représente l'état des systèmes modélisés, est introduite Chap. 2, alors que celle de règle de réécriture est décrite Chap. 3. 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. Par ailleurs, ce chapitre traite uniquement d'une analyse du comportement qualitatif des modèles, l'aspect quantitatif, les constantes cinétiques, ne sont pas abordées.

L'analyse statique, qui est introduite Chap. 4 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 (Sect. 4.4.1) 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 (Feret et Lý 2018). Cette analyse est implantée dans l'analyseur statique KaSa (Boutillier et collab. 2018a) et le choix des arbres de décisions, qui paramétrise l'analyse, est fait automatiquement par une pré-analyse. Le chapitre se conclut Sect. 7 et quelques perspectives sont données. La description du langage et de l'analyse reste volontairement assez haut niveau. Une formalisation complète et rigoureuse pour le langage complet est disponible dans les différents articles scientifiques qui sont cités dans le corps du texte.



## Chapter 2

# Graphes à sites

La section présente décrit 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 la section 3 pour décrire, grâce à des règles de réécriture, l'évolution de l'état de ces systèmes.

### 2.1 Signature

En Kappa, 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. Elle peut être représentée graphiquement par une *carte de contacts*, comme celle dessinée en Fig. 2.1. 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'interaction 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 ou comme le fait d'être méthylé – ou non. 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'interaction peuvent également porter un *état de liaison* : les sites qui portent le symbole  $\neg$  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 2.1.1** En Fig. 2.1 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 (Blinov et collab. 2004a). 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

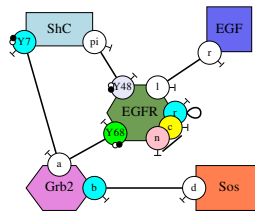


Figure 2.1: Une carte de contacts. Elle définit la signature d'un modèle en donnant la liste de toutes les sortes de protéines, leurs différents sites d'interaction, les différents états internes que peuvent prendre ces sites et les différentes liaisons potentielles entre ces sites.

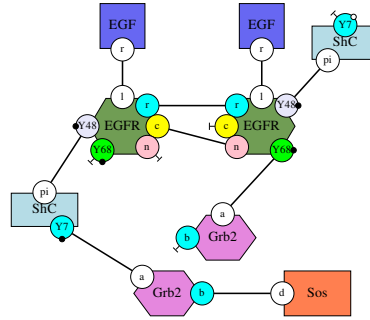


Figure 2.2: Un complexe biochimique. Il contient plusieurs occurrences de protéines. Chaque occurrence documente l'ensemble de ses sites d'interaction. Les sites qui peuvent porter un état interne en ont un. Par ailleurs, les sites sont soit libres, soit liés deux à deux.

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* a 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  $\neg$ ) 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*.

## 2.2 Complexes biochimiques

Les modèles Kappa décrivent l'évolution d'une soupe de complexes biochimiques. Un complexe biochimique est formé de plusieurs occurrences de protéines. Chaque occurrence d'une protéine est associée à un ensemble de sites d'interaction. 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 un complexe biochimique. De plus, un site ne peut pas être lié simultanément à deux sites. Un complexe biochimique forme un graphe connexe, ce qui signifie qu'il est possible de passer de n'importe quelle occurrence de protéines à n'importe quelle autre, en suivant zéro, un ou plusieurs liens.

**Exemple 2.2.1** En Fig. 2.2 est donné un exemple de complexe biochimique. Ce complexe est formé 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*. Chaque occurrence du récepteur membranaire est liée au site *r* d'une occurrence du ligand par son site *l*. Les occurrences de récepteur forment un dimer grâce à une double liaison, une liaison symétrique par leurs sites *r* respectifs et une liaison asymétrique entre le site *c* de l'un et le site *n* de l'autre. L'occurrence du récepteur membranaire dont le site *c* est lié a son site *Y68* phosphorylé et libre, alors que son site *Y48* est



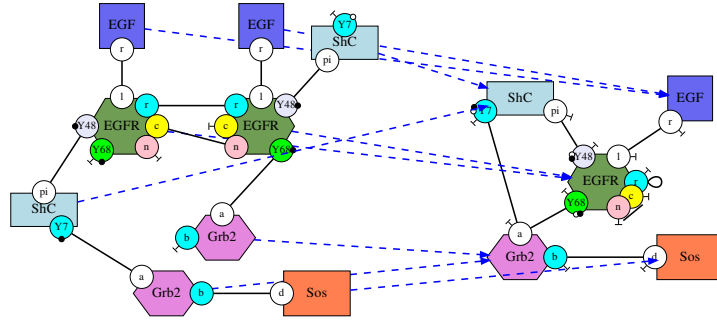


Figure 2.3: L'unique projection entre le complexe biochimique de la Fig. 2.2 et la carte de contacts de la Fig. 2.1. Cette projection est obtenue en associant chaque occurrence de protéines de l'espèce biochimique à l'unique sorte de protéines correspondante dans la carte de contacts.

phosphorylé et lié au site  $\pi$  d'une occurrence de la protéine d'échafaudage. Le site Y7 de cette occurrence de la protéine d'échafaudage est phosphorylé et lié au site a d'une occurrence de la protéine de transport dont le site b est lié au site d d'une occurrence de la protéine Sos. L'autre occurrence du récepteur a son site Y48 phosphorylé et lié au site  $\pi$  de l'autre occurrence de la protéine d'échafaudage. Le site Y7 de cette occurrence de la protéine d'échafaudage n'est ni phosphorylé, ni lié à un autre site. Enfin, le site Y68 de cette seconde occurrence du récepteur membranaire est lié au site a de l'autre occurrence de la protéine de transport. Celle-ci a son site b libre.

La signature d'un modèle restreint l'ensemble des complexes biochimiques de ce modèle. Tous les complexes biochimiques qui sont corrects du point de vue de la syntaxe ne sont ainsi pas adéquats. Ce rôle est assuré par la carte de contacts, qui d'une part, donne la liste de tous les sites d'interaction 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 un complexe 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'un complexe biochimique 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 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 complexe biochimique se projette sur la carte de contacts : ainsi la fonction qui associe à chaque nœud d'un complexe biochimique 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 tous les complexes biochimiques du modèle et chaque nœud de la carte de contacts résume toutes les configurations possibles des protéines du type correspondant.

**Exemple 2.2.2** En Fig. 2.3 est représentée la projection entre le complexe biochimique dessiné dans la Fig. 2.2 et la carte de contacts donnée en Fig. 2.1. Cette projection montre que ce complexe biochimique est compatible avec cette carte de contacts.

## 2.3 Motifs

L'évolution des complexes biochimiques est décrite par des règles de réécriture. Celles-ci définissent à la fois les conditions qui doivent être réalisées pour qu'une interaction donnée 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 qui permet donc de spécifier sous quelles conditions une interaction peut avoir lieu.

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 contiguë de complexe biochimique. De ce fait, il peut comporter zéro, une ou plusieurs occurrences de chaque sorte de protéines. Chaque occurrence de protéines

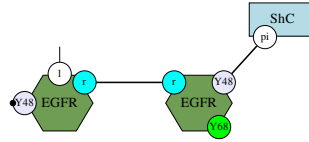


Figure 2.4: Un motif connexe. Il contient plusieurs occurrences de protéines. Chaque occurrence de protéines documente un sous ensemble de ses sites d'interaction. Chaque site peut éventuellement porter un état interne et éventuellement un état de liaison (en conformité avec la signature du modèle, donnée en Fig. 2.1). 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.

est associée à un ensemble de sites d'interaction. 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 (différent de lui-même donc). L'état de liaison d'un site peut également ne pas être spécifié.

**Exemple 2.3.1** En Fig. 2.4 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. L'occurrence de la protéine d'échafaudage mentionne uniquement son site  $\pi$ . Celui-ci est lié au site Y48 d'une des deux occurrences du récepteur membranaire. L'état d'activation de ce dernier site n'est pas précisé. Cette occurrence du récepteur membranaire mentionne également son site Y68, sans en préciser ni l'état interne ni l'état de liaison, et son site  $r$ , lui-même lié au site  $r$  de l'autre occurrence du récepteur membranaire. Cet autre occurrence mentionne également son site Y48 qui est phosphorylé mais dont l'état de liaison n'est pas spécifié et son site  $l$  qui est lié à un site qui n'est pas précisé.

Comme c'était le cas pour les complexes 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'interaction 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 complexes biochimiques, il doit être possible de projeter le motif sur la carte de contacts. Cela veut dire que la fonction qui associe à chaque nœud d'un motif l'unique nœud de la même sorte dans la carte de contacts est un homomorphisme.

## 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 complexes 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 les complexes biochimiques. Un complexe biochimique ne peut se plonger dans aucun autre motif connexe.

**Exemple 2.4.1** Deux exemples de plongements sont donnés en Fig. 2.5. Ce sont les seuls plongements entre ce motif et ce complexe biochimique. Dans le premier (voir en Fig. 2.5(a)) l'unique occurrence de la protéine d'échafaudage du motif est associée à l'occurrence de la protéine d'échafaudage du complexe biochimique dont le site Y7 est libre. L'occurrence du récepteur membranaire qui est liée à l'occurrence de la protéine

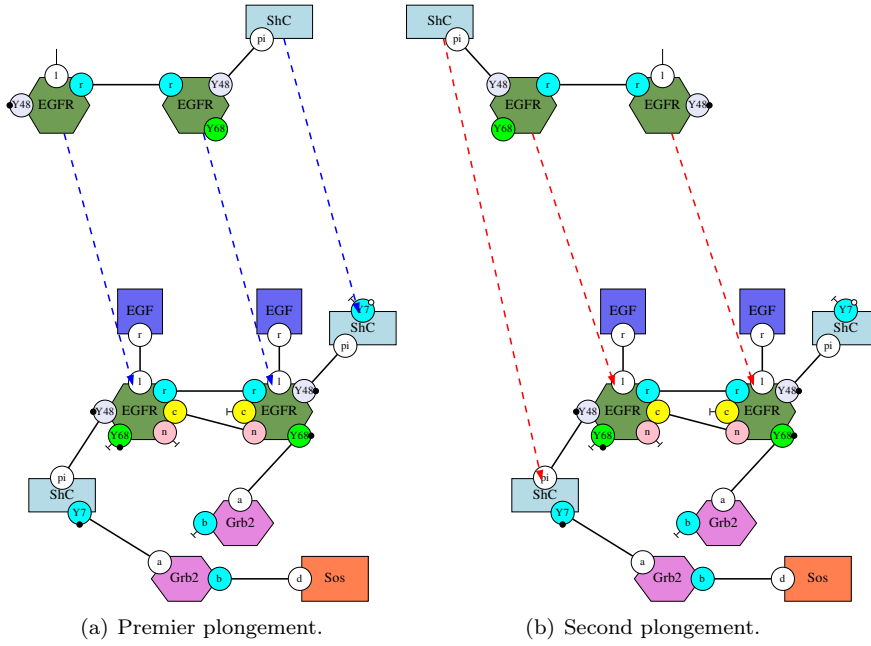


Figure 2.5: Deux plongements entre le motif donné dans la Fig. 2.4 et le complexe biochimique donné dans la Fig. 2.2. En 2.5(a) l'occurrence de la protéine d'échafaudage est associée à l'occurrence de la protéine d'échafaudage dont le site Y7 est libre. En 2.5(b) l'occurrence de la protéine d'échafaudage est associée à l'occurrence de la protéine d'échafaudage dont le site Y7 est lié.

d'échafaudage du motif est associée à l'occurrence du récepteur membranaire qui est liée à l'occurrence de la protéine d'échafaudage dont le site Y7 est libre. Enfin, l'autre occurrence du récepteur membranaire du motif est associée à l'autre occurrence du récepteur membranaire du complexe biochimique. Il est possible de remarquer que le site l de cette dernière occurrence du récepteur est lié dans le motif, sans que le site partenaire ne soit précisé, alors qu'il est explicitement lié au site r d'une occurrence du ligand dans le complexe biochimique. Dans le second plongement (voir en Fig. 2.5(b)) l'occurrence de la protéine d'échafaudage du motif est associée à l'occurrence de la protéine d'échafaudage du complexe biochimique dont le site Y7 est lié. L'occurrence du récepteur membranaire qui est liée à l'occurrence de la protéine d'échafaudage du motif est associée à l'occurrence du récepteur membranaire qui est liée à l'occurrence de la protéine d'échafaudage dont le site Y7 est lié. Enfin, l'autre occurrence du récepteur membranaire du motif est associée à l'autre occurrence du récepteur membranaire du complexe biochimique.

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 (Danos et collab. 2010 ; Petrov et collab. 2012).



## Chapter 3

# Réécriture de graphes à sites

Les motifs vont permettre de spécifier l'évolution potentielle de l'état des systèmes modélisés en Kappa, grâce à des règles de réécriture. C'est l'objet de cette section.

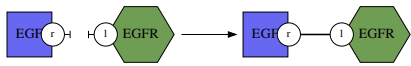
Afin de simplifier la présentation, seul un fragment du langage Kappa est présenté. En particulier, les règles de réécriture qui sont introduites dans cette section n'engendrent pas d'effets de bord. Un effet de bord est une transformation à l'extérieur du membre gauche des règles. Les effets de bords 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. Cela a permis de présenter tous les différents concepts de la syntaxe et de la sémantique de Kappa sous forme graphique.

### 3.1 Règles d'interaction

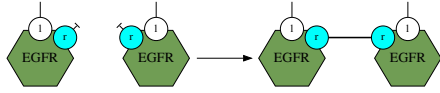
Les complexes biochimiques peuvent se transformer en appliquant des règles d'interaction. Une règle d'interaction est définie par une paire de motifs, qui contiennent exactement les mêmes sortes de protéines. Le premier motif spécifie quelles conditions locales doivent être réalisées pour permettre à 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'interaction.

**Exemple 3.1.1** *Des exemples de règles d'interaction sont données en Fig. 3.1. 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'interaction 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. La première étape est l'activation d'une occurrence du récepteur membranaire par une occurrence du ligand (voir en Fig. 3.1(a)). En se liant à une occurrence du ligand, une occurrence du récepteur change de conformation et peut alors établir une liaison symétrique avec une autre occurrence du récepteur qui doit pour cela être elle-même activée (voir en Fig. 3.1(c)). Comme seules les occurrences du ligand peuvent se lier aux sites l des occurrences du récepteur, il n'est pas nécessaire de mentionner les occurrences du ligand dans la règle. Il suffit d'écrire que les sites l des deux occurrences du récepteur doivent être liés sans préciser à quels sites. Après cette étape, les deux occurrences du récepteur tiennent le même rôle. Pour les distinguer, une liaison asymétrique peut alors s'établir (voir en Fig. 3.1(e)) entre le site c d'une des deux occurrences et le site n de l'autre occurrence. Le site Y68 de l'occurrence du récepteur qui est liée par son site c peut alors se faire phosphoryler par l'autre occurrence du récepteur membranaire (voir en Fig. 3.1(g)). Cela change la conformation de cette occurrence du récepteur membranaire et lui permet de se lier à une occurrence de la protéine de transport (voir en Fig. 3.1(i)). Indépendamment, les occurrences de la protéine de transport peuvent se lier aux occurrences de la protéine cible (voir en Fig. 3.1(k)).*

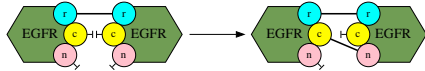
Chacune de ces interactions est réversible. Cependant les interactions inverses ne peuvent s'effectuer que sous certaines conditions. Ces interactions sont décrites dans la colonne de droite. Les liaisons symétriques entre les occurrences du récepteur membranaire capturent les occurrences du ligand qui ne peuvent alors pas se libérer (voir en Fig. 3.1(b)). Les liaisons asymétriques empêchent les liaisons symétriques de se briser (voir en Fig. 3.1(d)). Les liaisons asymétriques peuvent se briser sans condition (voir en Fig. 3.1(f)). La phosphorylation



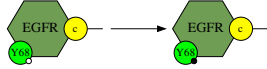
(a) Activation d'une occurrence du récepteur.



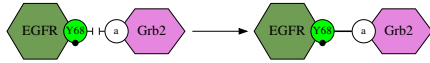
(c) Liaison symétrique



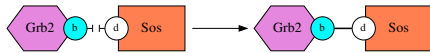
(e) Liaison asymétrique.



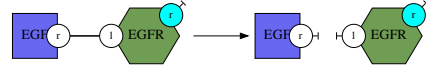
(g) Phosphorylation d'une occurrence du récepteur.



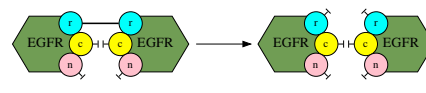
(i) Recrutement d'une occurrence de la protéine transporteur.



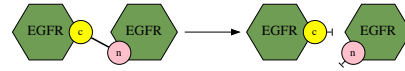
(k) Liaison d'une occurrence de la protéine transporteur à une occurrence de la protéine cible



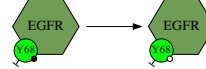
(b) Désactivation d'une occurrence du récepteur.



(d) Déliaison du lien symétrique.



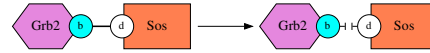
(f) Déliaison du lien asymétrique.



(h) Déphosphorylation d'une occurrence du récepteur.



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



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

Figure 3.1: Règles d'interaction impliquées dans le recrutement d'une occurrence de la protéine cible par la voie de signalisation courte (sans passer par la protéine d'échafaudage).

du site Y68 d'une occurrence du récepteur est bloquée quand ce site est lié (voir en Fig. 3.1(h)). Les liaisons entre les occurrences du récepteur et les occurrences de la protéine de transport d'une part, et celles entre les occurrences de la protéine de transport et celles de la protéine cible d'autre part, peuvent se défaire sans condition (voir en Figs. 3.1(j) et 3.1(l)).

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.

## 3.2 Réactions induites par une règle d'interaction

Comme signalé précédemment, le membre gauche d'une règle d'interaction 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'interaction qui ne peut plus être raffinée (sans ajouter de nouvelles composantes connexes) est alors appelée une règle-réaction (Harmer et collab. 2010).

**Exemple 3.2.1** En Fig. 3.2 est montré un exemple de deux raffinements d'une même règle d'interaction en deux règles-réactions. La règle d'interaction est celle qui permet de casser, en l'absence de lien asymétrique, le lien symétrique entre deux occurrences du récepteur membranaire (voir en Fig. 3.1(d)).

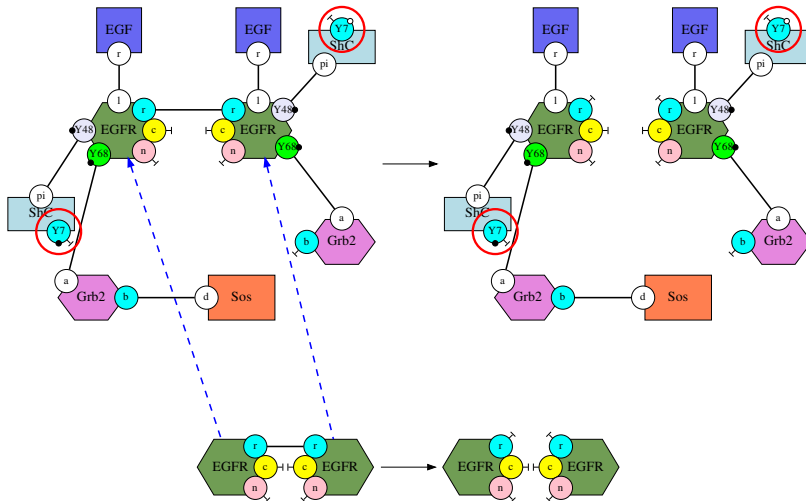
1. Dans le premier raffinement (voir en Fig. 3.2(a)), la règle est appliquée à un dimer dont la première occurrence du récepteur est liée par son site Y48 à une occurrence de la protéine d'échafaudage dont le site Y7 est libre et phosphorylé et par son site Y68 à une occurrence de la protéine de transport elle-même liée à une occurrence de la protéine cible. La deuxième occurrence du récepteur de ce dimer est liée par son site Y48 à une occurrence de la protéine d'échafaudage dont le site Y7 est libre et non-phosphorylé et par son site Y68 à une occurrence de la protéine de transport dont le site b est libre.
2. Dans le second (voir en Fig. 3.2(b)), les deux occurrences de la protéine d'échafaudage ont été interverties. Ainsi, la règle est appliquée à un dimer dont la première occurrence du récepteur est liée par son site Y48 à une occurrence de la protéine d'échafaudage dont le site Y7 est libre et non-phosphorylé et par son site Y68 à une occurrence de la protéine de transport elle-même liée à une occurrence de la protéine cible. La seconde occurrence du récepteur de ce dimer est liée par son site Y48 à une occurrence de la protéine d'échafaudage dont le site Y7 est libre et phosphorylé et par son site Y68 à une occurrence de la protéine de transport dont le site b est libre.

Bien que les deux complexes biochimiques qui apparaissent dans les membres gauches de ces deux règles-réactions soient formés exactement des mêmes occurrences de protéines et dans les mêmes configurations, puisque seul l'agencement entre ces occurrences change, il apparaît que les deux règles-réactions obtenues ne produisent pas les mêmes complexes biochimiques. Ceci justifie pleinement le choix, dans Kappa, de représenter la topologie des liens entre les occurrences de protéines. Sans celle-ci il est impossible de décrire fidèlement la séparation des occurrences de récepteurs, tout en respectant la distribution des différentes occurrences de protéines et de leurs configurations dans chacun des complexes biochimiques résultant de cette séparation. Par exemple, dans le langage BCS (Ded et collab. 2016), les complexes biochimiques sont représentés par l'ensemble des occurrences de protéines qui les constituent, ainsi que leurs configurations, mais sans préciser la topologie des liens entre ces occurrences de protéines. Aussi il est impossible de représenter fidèlement la règle de déliaison qui est dessinée en Fig. 3.1(d) dans ce langage.

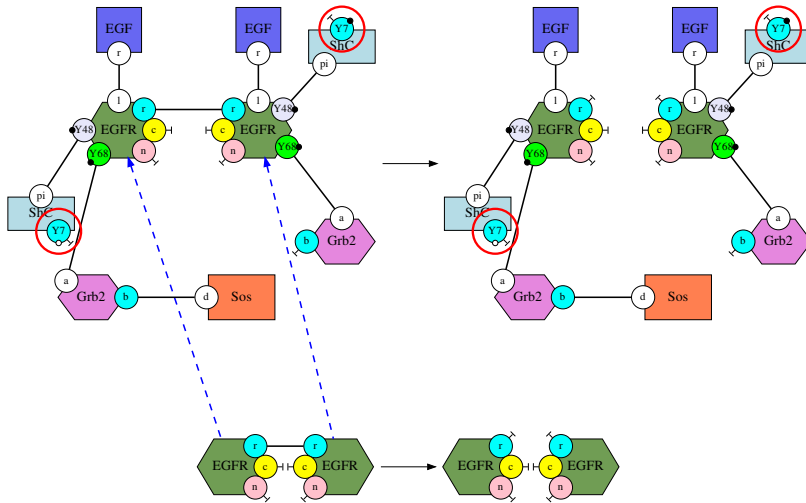
## 3.3 Réseaux de réactions sous-jacents

Un ensemble de règles peut alors être traduit en un ensemble – éventuellement infini – de règles-réactions en remplaçant chaque règle d'interaction par l'ensemble des règles-réactions qui peuvent être obtenues comme raffinement de ces règles.

Ensuite, quitte à nommer les différents complexes biochimiques qui peuvent intervenir dans les règles-réactions ainsi obtenues, nous pouvons assimiler ces règles-réactions à un réseau de réactions (éventuellement



(a) Premier raffinement.



(b) Second raffinement.

Figure 3.2: Deux exemples de raffinements d'une même règle d'interaction en deux règles-réactions. Les différences entre ces deux raffinements sont mises en valeur par des cercles rouges (les deux occurrences de la protéine *Shc* ont été échangées). Dans les deux cas, la règle-réaction est obtenue en ajoutant dans le membre gauche et dans le membre droit de la règle d'interaction exactement la même information sur le contexte d'application de la règle.



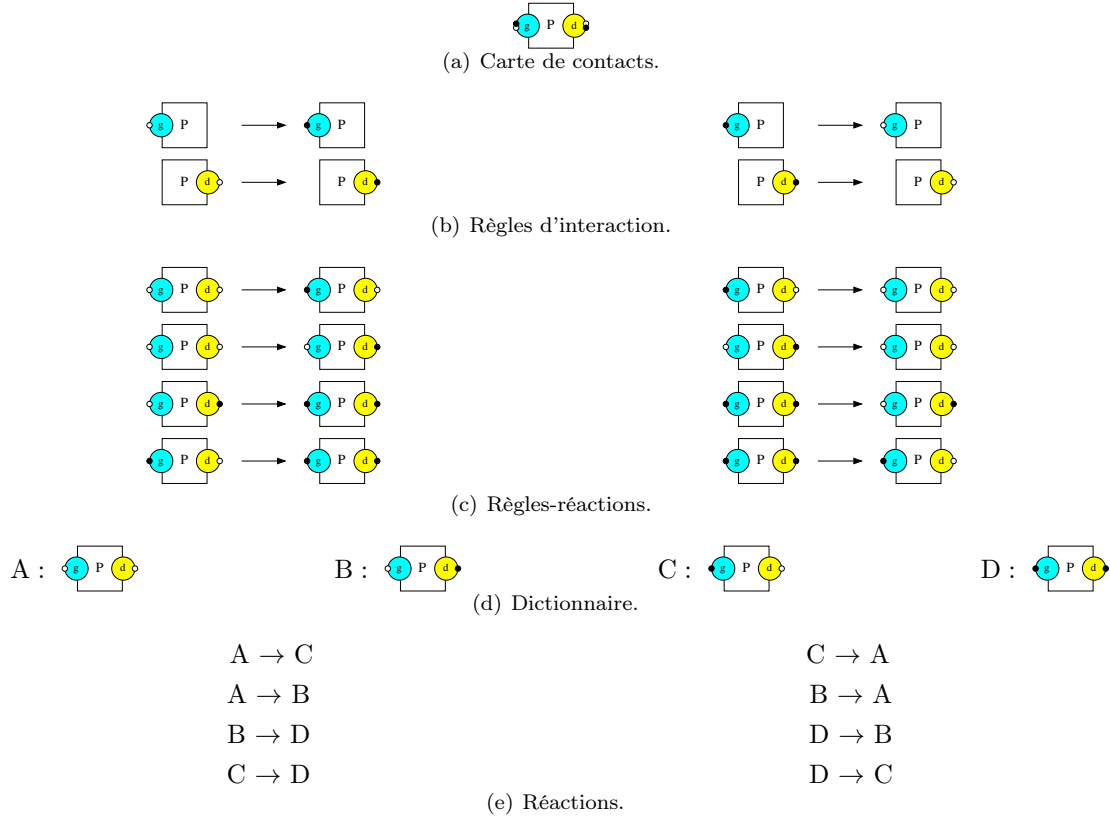


Figure 3.3: Un modèle formé d'une carte de contacts et de quatres règles d'interaction et sa traduction sous forme réseau réactionnel.

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.

**Exemple 3.3.1** Pour conclure cette section, nous détaillons la génération d'un réseau de réactions à partir d'un ensemble jouet de règles Kappa. Nous considérons un modèle avec une seule sorte de protéines qui admet deux sites,  $g$  et  $d$ , chacun pouvant être phosphorylé ou non. La signature du modèle est donnée par la carte de contacts qui est dessinée en Fig. 3.3(a). La phosphorylation et la déphosphorylation de chaque site dans une occurrence de protéines peut se faire indépendamment de l'état de l'autre site, ce qui est formalisé dans les quatre règles données en Fig. 3.3(b). Ainsi ni les règles de phosphorylation, ni celles de déphosphorylation d'un site, ne mentionnent l'état de phosphorylation de l'autre site.

Pour obtenir les règles-réactions associées à ce modèle jouet, il suffit d'explicitier dans quel contexte local les interactions peuvent se produire. Ainsi chaque règle Kappa donne ici lieu à deux règles-réactions selon que le site qui n'est pas mentionné dans la règle initiale est phosphorylé ou non. Ces règles-réactions sont données en Fig. 3.3(c).

La prochaine étape est de nommer les différents complexes biochimiques qui interviennent dans les règles-réactions ainsi obtenues. Une occurrence de protéines dont aucun site n'est phosphorylé, est appelée  $A$ , une occurrence de protéines dont seul le site  $d$  est phosphorylé, est appelée  $B$ , une occurrence de protéines dont seul le site  $g$  est phosphorylé, est appelée  $C$  et une occurrence de protéines dont les deux sites sont phosphorylés, est appelée  $D$ . Les réactions données en Fig. 3.3(e) sont obtenues en remplaçant chaque occurrence de complexe biochimique par son nom dans les règles-réactions.

Le choix d'une sémantique en terme de réseaux de réactions a été fait pour simplifier la présentation. C'était ainsi que le langage BNGL avait été implanté initialement (Blinov et collab. 2004a). 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 (Danos et collab. 2008 ; Feret et collab. 2013), soit dans un cadre catégorique (Danos et collab. 2012 ; Feret 2015). La première méthode est plus opérationnelle alors que la seconde abstrait au contraire beaucoup de détails. Il faut cependant noter que les cadres catégoriques usuels de la réécriture de graphes, que ce soit par push-out simples (Löwe 1993), push-out doubles (Corradini et collab. 1997) ou sesqui-pushout (Corradini et collab. 2006)) ne représentent pas fidèlement les effets de bord avec la définition usuelle des plongements entre graphes à sites. Deux approches connues permettent d’y remédier. Il est possible soit de changer la définition des plongements (Danos et collab. 2012 ; Feret 2015), soit d’enrichir les objets de la catégorie par des contraintes (Behr et Krivine 2019).

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 (Danos et collab. 2007b ; Boutillier et collab. 2017).

## Chapter 4

# Analyse des motifs accessibles

Si la carte de contacts (e.g. voir en Fig. 2.1 à la page 13) donne un aperçu rapide de toutes les interactions potentielles entre les différents sites des occurrences des protéines dans un modèle, elle n'est en générale pas suffisante pour décrire précisément la structure de ses complexes biochimiques. En effet, l'état des différents sites d'interaction d'un complexe biochimique est souvent contraint par des invariants structuraux. 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 d'un même complexe 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 complexes 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, afin 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 (Boutillier et collab. 2018a) et intégrée dans la plate-forme de modélisation en ligne dédiée au langage Kappa (Boutillier et collab. 2018b). Ceci permet d'assister le modélisateur pendant l'écriture du modèle en lui fournissant les contraintes structurelles qui sont vérifiées par les complexes biochimiques et en l'avertissant de la présence de règles mortes, après chaque ajout ou modification d'une règle d'interaction.

### 4.1 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 3.3 page 21, 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  $\mathbb{N}$

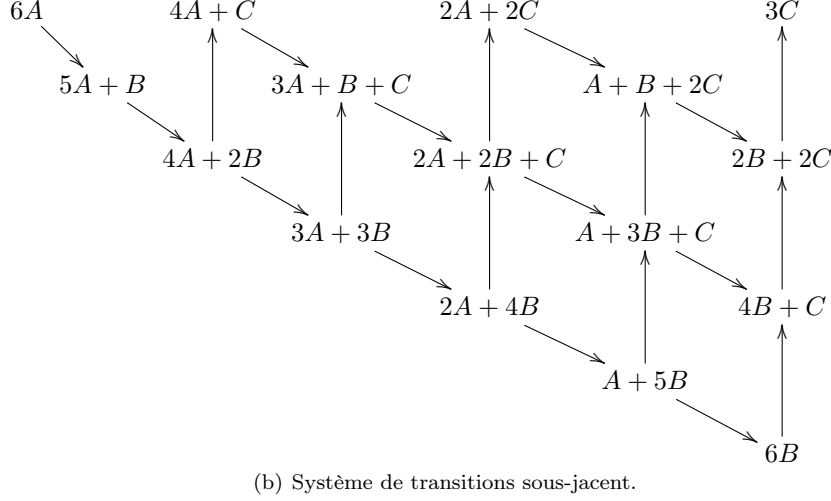
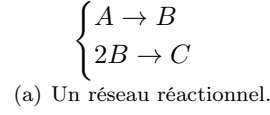


Figure 4.1: Un réseau réactionnel et sa sémantique. En 4.1(a) un réseau réactionnel formé de deux réactions. La première permet de transformer une occurrence de l'espèce biochimique  $A$  en une occurrence de l'espèce biochimique  $B$ , la seconde permet de transformer deux occurrences de l'espèce biochimique  $B$  en une occurrence de l'espèce biochimique  $C$ . La restriction de l'ensemble de toutes les transitions possibles aux états qui sont atteignables à partir d'un état initial formé de six occurrences de la protéine  $A$  est dessinée en 4.1(b) sous la forme d'un système de transitions.

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  $q \rightarrow q'$ .

**Exemple 4.1.1** *Un système de transitions est donné comme exemple en Fig. 4.1(b). Il correspond à la restriction du système de transitions associé aux réactions qui sont données en Fig. 4.1(a) aux états accessibles à partir d'un état initial formé de six occurrences de l'espèce biochimique  $A$ . Dans ce réseau, la somme entre le nombre d'occurrences de  $A$ , de  $B$  et deux fois celui de  $C$  est toujours égal à la quantité initiale de  $A$ . En effet, cette quantité n'est modifiée par l'application d'aucune des réactions du réseau.*

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 :

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

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* (Tarski 1955) 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  $\text{lfp } \mathbb{F}$ , 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.

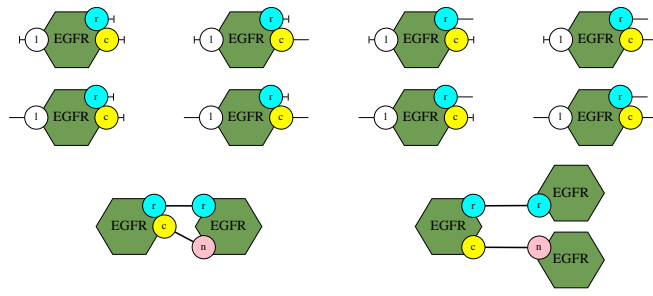


Figure 4.2: Un ensemble de motifs d'intérêt pour l'analyse des complexes biochimiques accessibles dans le modèle des premières interactions qui interviennent dans l'acquisition du facteur de croissance de l'épiderme. Les huit premiers motifs permettent de s'intéresser aux relations potentielles entre l'état de liaison des sites  $l$ ,  $r$  et  $c$  dans les occurrences du récepteur membranaire. Ces 8 motifs correspondent exactement à chaque combinaison possible pour l'état de ces 3 sites, chacun de ces sites pouvant être libre ou lié. Les deux derniers motifs permettent de distinguer deux occurrences du récepteur liées par une double liaison d'une chaîne d'au moins trois occurrences du récepteur.

## 4.2 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 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 4.2.1** *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 Fig. 4.2. 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) (Danos et collab. 2008). Elles permettent d'abstraire un ensemble de complexes 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 dimers dans ce modèle fait intervenir des doubles liaisons. Il est légitime de se demander s'il est possible de former des chaînes 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 chaîne 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 dit 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'apparaît dans un complexe biochimique présent 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 nombreux sont les complexes 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  $\gamma_{\mathcal{P}}(X^{\#})$ . 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*.

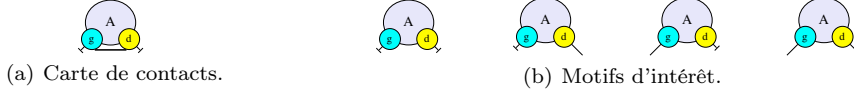
Réciproquement, étant donné un ensemble d'états  $X$ , l'ensemble des éléments de l'ensemble  $\mathcal{P}$  qui apparaissent dans au moins un complexe biochimique d'un état élément de l'ensemble  $X$  sera noté  $\alpha_{\mathcal{P}}(X)$ . La fonction  $\alpha_{\mathcal{P}}(X)$  est croissante également. 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* (Cousot et Cousot 1977 ; Cousot 1999).

**Exemple 4.2.2** En Fig. 4.3 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 Fig. 4.3(a). 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 Fig. 4.3(b) 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  $A$ .

**Exemple 4.2.3** Des exemples de meilleures approximations sont donnés en Fig. 4.3(c). Par abus de langage, nous appelons la meilleure approximation d'un complexe biochimique, la meilleure approximation de l'ensemble formé d'un seul état lui même formé de ce seul complexe. Le modèle admet deux types de complexes biochimiques. Les occurrences de la protéine  $A$  peuvent former des chaines d'occurrences de protéines liées successivement par leur site  $d$  et  $g$ , laissant le site  $g$  de la première occurrence de la protéine  $A$  et le site  $d$  de la dernière occurrence de la protéine  $A$  libres. Les occurrences de la protéine  $A$  peuvent aussi former des anneaux en reliant le premier et le dernier sites d'une chaine d'occurrences de protéines. La meilleure approximation d'une chaine d'occurrences de la protéine  $A$  dépend de la taille de cette chaine. Ainsi, la meilleure approximation d'une chaine réduite à une occurrence de la protéine  $A$  est l'ensemble qui contient uniquement la vue locale dont les deux sites sont libres ; la meilleure approximation d'une chaine formée d'exactly deux occurrences de la protéine  $A$  est l'ensemble qui contient deux vues locales : l'une avec le site  $g$  libre et le site  $d$  lié, l'autre avec le site  $g$  lié et le site  $d$  libre ; enfin la meilleure approximation des chaines d'occurrences de la protéine  $A$  de longueur au moins égale à 3 contient également la vue locale dont les deux sites  $g$  et  $d$  sont liés. Par contre, la meilleure approximation d'un anneau d'occurrences de la protéine  $A$  est toujours l'ensemble formé uniquement de la vue locale dont les deux sites  $g$  et  $d$  sont liées, et ce, quelle que soit la longueur de cette anneau. La fonction qui associe à chaque ensemble de complexe biochimique sa meilleure approximation est distributive. Cela signifie que la meilleure approximation d'un ensemble de complexes biochimiques est l'union de la meilleure approximation des singletons correspondants.

**Exemple 4.2.4** Des exemples de concrétisations sont donnés en Fig. 4.3(d). Par définition, la concrétisation de l'ensemble formé uniquement de la vue locale dans laquelle les deux sites sont libres, est l'ensemble de tous les états qui ne contiennent pas d'autres vues locales. Il s'agit donc des états qui ne contiennent que le complexe biochimique composé d'exactly une occurrence de la protéine  $A$ . Par la même démarche, la concrétisation de l'ensemble formé uniquement de la vue locale dont les deux sites sont liés est l'ensemble de tous les états qui ne contiennent que des anneaux d'occurrences de la protéine  $A$  (quitte à utiliser cette vue locale plusieurs fois). Par contre, tout complexe contenant une vue locale avec exactement un site libre, doit contenir également une vue locale avec l'autre site, libre. De ce fait, la concrétisation d'un ensemble composé d'une seule vue locale avec un site libre et l'autre lié, est l'ensemble ne contenant que l'état vide (qui est noté  $\emptyset$ ). Si seules les deux vues locales où un site est lié et l'autre libre sont autorisées, seules des chaines de deux occurrences de la protéine  $A$  peuvent être construites. Enfin, si seule la vue locale avec les deux sites libres est interdite, il est possible de former n'importe quelle chaine d'occurrences de la protéine  $A$  de taille au moins égale à 2 et n'importe quel anneau d'occurrences de protéines (sans restriction de taille). La fonction de concrétisation n'est pas distributive (l'image de l'union de deux ensembles de vues locales peut être un sur-ensemble strict de l'union de leurs images).

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



- $\alpha_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\}$
  - $\alpha_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \text{---} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} , \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\}$
  - $\alpha_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \text{---} \text{A} \text{---} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} , \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} , \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\}$
  - $\alpha_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\}$
  - $\alpha_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \text{---} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\}$
- (c) Exemples de meilleures approximations.

- $\gamma_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \left\{ n \cdot \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \mid n \in \mathbb{N} \right\}$
- $\gamma_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \left\{ n_1 \cdot \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} + n_2 \cdot \begin{array}{c} \text{A} \text{---} \text{A} \\ \text{s} \text{---} \text{d} \end{array} + \dots \mid n_1, n_2, \dots \in \mathbb{N} \right\}$
- $\gamma_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \{\emptyset\}.$
- $\gamma_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \{\emptyset\}.$
- $\gamma_{\mathcal{P}} \left( \left\{ \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} , \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \right\} \right) = \left\{ n \cdot \begin{array}{c} \text{A} \text{---} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \mid n \in \mathbb{N} \right\}.$
- $\gamma_{\mathcal{P}} \left( \left( \begin{array}{c} \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} , \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \end{array} \right) \right) = \left\{ \begin{array}{l} n_2 \cdot \begin{array}{c} \text{A} \text{---} \text{A} \\ \text{s} \text{---} \text{d} \end{array} \\ + n_3 \cdot \begin{array}{c} \text{A} \text{---} \text{A} \text{---} \text{A} \\ \text{s} \text{---} \text{d} \end{array} + \dots \\ + n'_1 \cdot \begin{array}{c} \text{A} \\ \text{s} \text{---} \text{d} \end{array} + n'_2 \cdot \begin{array}{c} \text{A} \text{---} \text{A} \\ \text{s} \text{---} \text{d} \end{array} + \dots \end{array} \mid \begin{array}{l} n_2, n_3, \dots \in \mathbb{N} \\ n'_1, n'_2, \dots \in \mathbb{N} \end{array} \right\}.$

(d) Exemples de concrétisations

Figure 4.3: Un exemple jouet pour mieux comprendre le comportement des fonctions d'abstraction et de concrétisation. En 4.3(a), 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 4.3(b), 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. En 4.3(c) sont donnés des exemples de meilleure approximation d'ensemble d'états. Cela consiste à collecter les vues locales qui peuvent apparaître dans ces états. Réciproquement, En 4.3(d) sont donnés des exemples de concrétisations d'ensembles de vues locales. Ceci consiste à recomposer l'ensemble des états qui ne contiennent aucune occurrence des vues locales manquantes. Dans le cas particulier des vues locales, cela revient à prendre en compte tous les états composés uniquement des vues locales mises à disposition, sachant que chaque vue peut être utilisée zéro, une ou plusieurs fois.

$$\gamma_{\mathcal{P}} \left( \alpha_{\mathcal{P}} \left( \begin{array}{c} \text{A} \\ \text{---} \end{array} \right) \right) = \left\{ n_1 \cdot \begin{array}{c} \text{A} \\ \text{---} \end{array} + n_2 \cdot \begin{array}{c} \text{A} \quad \text{A} \\ \text{---} \end{array} + \dots \mid n_1, n_2, \dots \in \mathbb{N} \right\}$$

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

$$\alpha_{\mathcal{P}} \left( \gamma_{\mathcal{P}} \left( \begin{array}{c} \text{A} \\ \text{---} \end{array}, \begin{array}{c} \text{A} \\ \text{---} \end{array} \right) \right) = \left\{ \begin{array}{c} \text{A} \\ \text{---} \end{array} \right\}$$

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

Figure 4.4: Suite de l'exemple donné en Fig. 4.3. Un exemple d'application de la composée de fonctions  $\gamma_{\mathcal{P}} \circ \alpha_{\mathcal{P}}$  est montré en 4.4(a). Celui-ci montre que l'abstraction ne permet pas de distinguer des ensembles d'occurrences de la protéine A et ce quels que soient leurs tailles et leurs nombres. En 4.4(b) 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.

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 4.2.5** 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, voir également en Fig. 4.3(d), 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 4.2.6** 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 des complexes 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.

### 4.3 Transferts de point-fixes

Le plus petit point fixe qui définit l'ensemble des 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  $X^{\#}$ , de considérer l'ensemble des états concrets  $\gamma_{\mathcal{P}}(X^{\#})$  qui vérifient la propriété  $X^{\#}$ , 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  $\mathbb{F}$  (Cousot et Cousot 1979). 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 (Cousot et Cousot 1979). 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  $\mathbb{F}^\sharp$  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  $\text{lfp } \mathbb{F} \subseteq \gamma_{\mathcal{P}}(\text{lfp } (\mathbb{F}^\sharp))$  est également satisfaite (Cousot et Cousot 1979).

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  $P$  apparaîtrait également dans le membre gauche de la règle-réaction et ne serait donc 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 apparaît dans un état dans la concrétisation de  $Y$ .

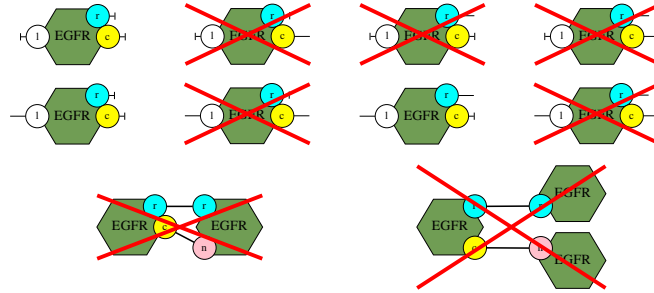
Ainsi, pour calculer les nouveaux motifs d'intérêt de l'ensemble  $[\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}](Y)$ , il suffit de calculer tous les *chevauchements* possibles entre un nouveau motif d'intérêt potentiel (dans  $\mathcal{P} \setminus Y$ ) et un membre droit d'une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

**Exemple 4.3.1** *Un exemple de cette construction est dessiné en Fig. 4.5. L'état abstrait actuel est donné en Fig. 4.5(a). Seules trois vues locales sont pour l'instant autorisées, celle avec aucun des sites  $l$ ,  $r$ ,  $c$  lié, celle avec seul le site  $l$  lié et celle avec seuls les sites  $l$  et  $r$  liés. Par ailleurs, ni les doubles liaisons, ni les chaînes d'au moins trois récepteurs ne sont autorisées à cet instant de l'analyse. La preuve que l'on peut construire une occurrence de protéines avec les trois sites  $l$ ,  $r$  et  $c$  liés, est donnée en Fig. 4.5(b). Il suffit d'identifier cette vue locale à l'occurrence gauche du récepteur dans le membre droit de la règle (les plongements en pointillés bleus et marrons envoient ces deux occurrences de protéines sur une même occurrence de protéines) qui permet d'établir une liaison asymétrique entre deux récepteurs membranaires (voir en Fig. 3.1(e)). Ceci est possible car les sites en commun dans ces deux occurrences de protéines sont dans un état compatible : en effet, la vue locale demande que ces sites soient liés, alors que le membre droit de la règle précise à quels sites ils le sont. La règle est alors spécialisée à la production de la nouvelle vue locale pour ce chevauchement particulier entre la vue locale et le membre droit de la règle. Cela consiste à ajouter dans les membres gauche et droit de la règle l'information que le site  $l$  est lié. Pour conclure, il suffit alors d'exhiber un plongement (ici dessiné en pointillés rouges) en le membre gauche de la règle ainsi raffinée et une espèce biochimique, en vérifiant que cette espèce biochimique ne contient aucune occurrence des motifs d'intérêt non encore autorisés. Le dimer avec uniquement une liaison symétrique et dans lequel les deux occurrences du récepteur membranaire sont toutes deux liées à des occurrences du ligand et les sites  $Y48$  et  $Y68$  non phosphorylés et libres rempli parfaitement ces conditions. Ainsi la vue locale dans laquelle les trois sites  $l$ ,  $r$  et  $c$  sont liés simultanément sera utilisable dès la prochaine itération de l'analyse.*

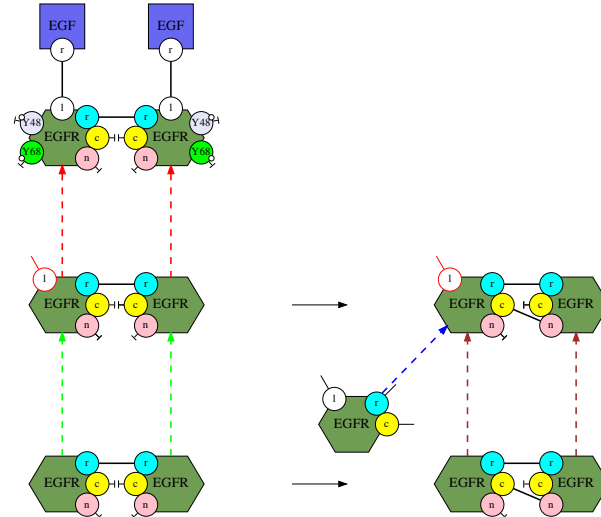
Lors du calcul de l'ensemble  $[\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}](Y)$ , l'étape la plus coûteuse en temps de calcul est de vérifier que les membres gauches des règles raffinées peuvent apparaître dans un état de l'ensemble  $\gamma_{\mathcal{P}}(Y)$ . La section suivante a pour but de réduire ce coût moyennant une approximation supplémentaire.

## 4.4 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



(a) Un état abstrait.



(b) La construction de la vue locale dans laquelle les trois sites sont liés peut être construite en une étape à partir de cet état abstrait.

Figure 4.5: Découverte d'un nouveau motif d'intérêt accessible dans le modèle des premières étapes de la voie de l'acquisition du facteur de croissance de d'épiderme (voir en Fig. 3.1 page 20). En 4.5(a), il est supposé qu'à ce moment de l'analyse, seules trois vues locales sont autorisées. Par ailleurs, il n'est permis de former ni des doubles liaisons entre récepteurs membranaires, ni des chaînes de trois récepteurs ou plus. En 4.5(b), la preuve que la vue locale doit être déclarée accessible à ce niveau d'abstraction est représentée sous forme de diagramme. Elle consiste à appliquer la règle de liaison asymétrique en identifiant la vue locale au récepteur de gauche dans le membre droit de la règle et en raffinant la règle en conséquence. Le membre gauche de la règle obtenue apparaît dans une espèce biochimique ne contenant aucun motif d'intérêt non encore découvert, ce qui conclut la preuve.

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  $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$ .

#### 4.4.1 Ensembles de motifs orthogonaux

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 (Feret et Lý 2018). 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 espèce biochimique, 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.

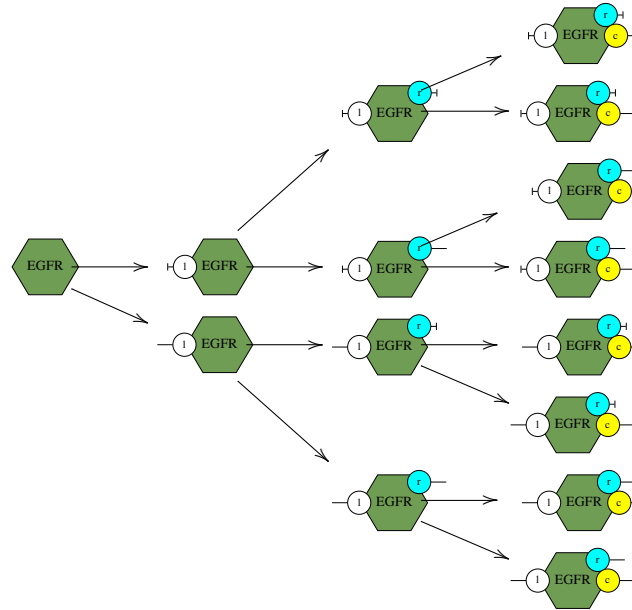
Le choix exact des ensembles de motifs orthogonaux repose sur une analyse préliminaire qui calcule, par inspection des règles du modèle, quelles questions intéressantes se posent. Trois catégories de questions sont considérées par défaut dans l'analyseur KaSa (mais il est possible de paramétrer l'analyse 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 (Danos et collab. 2008). La seconde permet de détecter des relations entre l'état des sites dans des occurrences de protéines qui partagent un lien (Feret et Lý 2018) dans le but d'analyser les déplacements de complexes biochimiques lorsque ceux-ci sont codés 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 (Feret et Lý 2018). Une quatrième sorte d'ensembles de motifs orthogonaux est en cours d'implantation. Elle se concentre sur la formation de complexes biochimiques cycliques : son but est de prouver l'absence de complexes biochimiques de taille non bornée (Boutillier et collab.).

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.

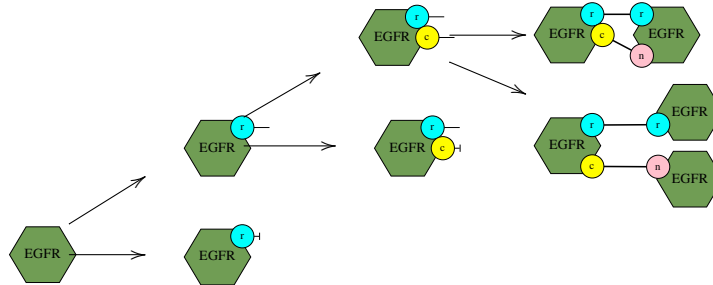
**Exemple 4.4.1** *L'ensemble des motifs d'intérêt introduit en Fig. 4.2 est inclus dans la réunion de deux ensembles de motifs orthogonaux. En effet, l'ensemble des vues locales peut être obtenu, en partant d'une occurrence de la protéine EGFR sans aucun site, en se demandant successivement si le site  $l$  est libre ou non, si le site  $r$  est libre ou non et si le site  $c$  est libre ou non. L'arbre de décision correspondant se trouve en Fig. 4.6(a). 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. L'arbre de décision ainsi obtenu est donné en Fig. 4.6(b).*

#### 4.4.2 Pas de calcul abstraits

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 se produit lors du test de la nouvelle condition utilisée pour prouver 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 (voir en Fig. 4.5). 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



(a) Ensemble de motifs orthogonaux pour les vues locales.



(b) Ensemble de motifs orthogonaux pour discuter des doubles liaisons.

Figure 4.6: Deux exemples d'ensemble de motifs orthogonaux. En 4.6(a), l'ensemble des vues locales (voir les huit premiers motifs en Fig. 4.2 page 27) sous forme d'arbre de décision. En 4.6(b), celui pour discuter de la présence potentielle de doubles liaisons entre des récepteurs et de la présence potentielle de trimers (voir les deux derniers motifs en Fig. 4.2).

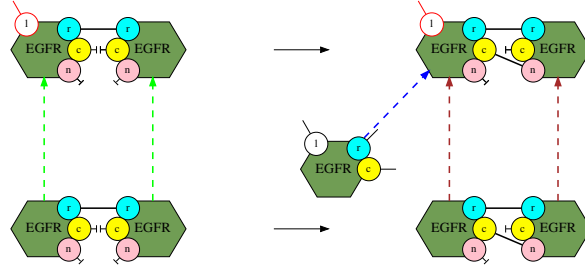


Figure 4.7: L'étape abstraite de la Fig. 4.5 revisitée avec la procédure de décision approchée. Au lieu de vérifier que chaque motif connexe du membre gauche de la règle raffinée se plonge dans un complexe biochimique dans la concrétisation de l'état abstrait, il suffit de s'assurer, pour chaque occurrence 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.

être compatible avec aucun des motifs de cet ensemble de motifs 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.

**Exemple 4.4.2** En Fig. 4.7, l'étape de calcul qui avait été décrite en Fig. 4.5 est rejouée en remplaçant le test de réalisabilité par cette procédure approchée. Au lieu de construire un plongement du membre gauche de la règle raffinée vers un complexe biochimique afin de vérifier qu'il ne contient pas de motif non encore autorisé, la nouvelle procédure se contente de vérifier pour chaque occurrence de protéines dans le membre gauche de la règle raffinée et pour chaque ensemble de motifs orthogonaux portant sur cette sorte de protéines si celui-ci contient un motif autorisé compatible avec cette occurrence. Dans ce cas, cela revient à vérifier qu'il existe bien une vue locale déjà autorisée dans laquelle les deux sites  $l$  et  $r$  sont liés, alors que le site  $c$  est libre et que le motif dans lequel le site  $r$  est lié et le site  $c$  est libre est autorisé dans le deuxième ensemble de motifs orthogonaux.

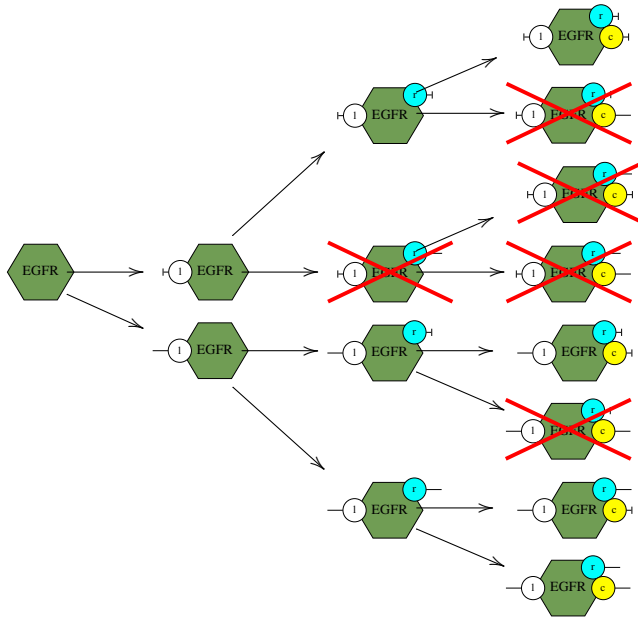
Outre le fait de ne pas vérifier l'existence d'un complexe biochimique qui pourrait compléter le collage obtenu entre les motifs connexes du membre gauche de la règle raffinée et les motifs déjà déclarés potentiellement accessibles par l'analyse, il est intéressant de remarquer que la procédure de décision 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 4.4.3** Le résultat de l'itération pour le modèle formé des règles qui avaient été décrites en Fig. 3.1 pour les ensembles de motifs orthogonaux qui avaient été introduits en Fig. 4.6, est donné en Fig. 4.8. 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 (voir en Fig. 4.8(a)), 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 (voir en Fig. 4.8(b)) montre qu'il est impossible de former des chaînes d'au moins trois récepteurs membranaires.

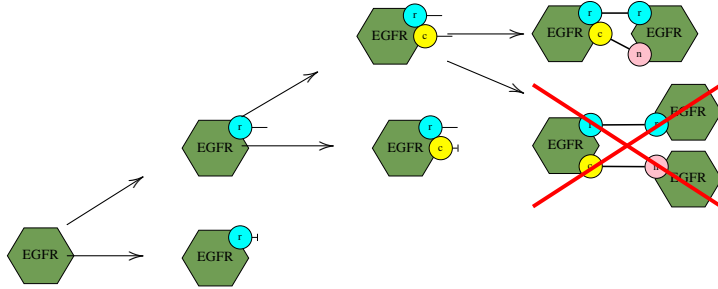
Il est important de rappeler que l'analyse ne donne qu'une sur-approximation des états accessibles. Ainsi, tout motif prouvé comme non accessible est bien inaccessible. Par contre, il n'y a aucune garantie qu'un motif non prouvé inaccessible puisse apparaître dans un état accessible depuis un des états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

#### 4.4.3 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,



(a) Analyse d'accessibilité pour l'ensemble des vues locales (voir en Fig. 4.6(a)).

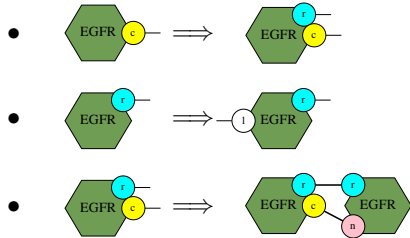


(b) Analyse d'accessibilité pour discuter de la présence éventuelle de trimers et de double liaisons.

Figure 4.8: Résultat de l'analyse pour les deux ensembles de motifs orthogonaux donnés en Fig. 4.6. Les motifs orthogonaux sont aux feuilles des arbres de décision. Ceux qui sont barrés en rouge n'apparaissent dans aucune exécution du modèle (pour n'importe quel état initial sans lien). Par construction de l'arbre de décision, les nœuds dont tous les enfants sont inaccessibles sont également inaccessibles et donc barrés eux-aussi.

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 un complexe biochimique accessible peut toujours se raffiner en l'un des motifs de la postcondition.

**Exemple 4.4.4** *Le résultat de l'analyse décrit en Fig. 4.8 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. 4.4.1). Les règles pour lesquelles ce n'est pas le cas sont reportées à l'utilisateur.

## 4.5 Étude de performance et utilisation concrète

Nous avons utilisé l'analyseur statique KaSa sur plusieurs modèles. Les résultats de ces analyses sont décrites en Fig. 4.9. Les onze premiers modèles sont des traductions directes des modèles qui sont fournis avec la distribution du logiciel BNGL (Blinov et collab. 2004a). Le modèle 'sos' décrit les premiers événements de l'acquisition du facteur de croissance de l'épiderme (il comprend entre autres les règles données en Fig. 3.1). Les modèles 'machine' et 'ensemble' sont deux versions de la voie de signalisation MAPK, publiées par Eric Deeds et Ryan Suderman (Suderman et Deeds 2013). Les versions du modèle 'korkut' concernent la voie de signalisation de la protéine Ras. Ce modèle a été conçu par John Bachman et Benjamin Gyori (Sorger lab) dans le cadre du projet DARPA Big Mechanism (Cohen 2015) en utilisant des outils de traitement automatique du langage naturel pour extraire des faits de la littérature, en assemblant ces faits en Kappa, puis en corrigeant manuellement le modèle obtenu (Gyori et collab. 2017). Le modèle 'tgf' s'intéresse lui à la matrice extra-cellulaire de la protéine  $\text{tgf-}\beta$ . Cinq versions de ce modèle ont été analysées. Elles ont été assemblées à la main d'après la littérature par Nathalie Théret et Jean Coquet, puis corrigées avec l'aide de l'analyseur KaSa. Enfin, plusieurs versions du modèle de la voie de signalisation de la protéine Wnt, écrites par Héctor F. Medina Abarca (Fontana Lab) dans le cadre du projet DARPA Big Mechanism (Cohen 2015) ont été analysées. Ce modèle a également été assemblé manuellement après lecture humaine de la littérature. Dans ce dernier modèle, le grand nombre de règles vient du fait que des scripts ont été utilisés pour raffiner des règles d'interaction génériques afin d'ajuster leur cinétique en fonction d'information contextuelle sur l'état des occurrences de protéines qui interagissent et de leurs voisines.

Pour chaque modèle et chaque version de ce modèle, nous avons reporté le nombre total de règles dans le modèle, ainsi que nombre d'implications découvertes par l'analyse portant sur des relations soit entre au moins deux sites, soit entre les états de liaison et d'activation d'un même site. Nous avons également donné le nombre de règles qui ont été trouvées mortes par l'analyseur statique (du fait de l'approximation, l'analyseur peut manquer des règles mortes, par contre, toute règle détectée morte l'est). Nous donnons également le temps de calcul total de l'analyse, ce qui montre que KaSa passe à l'échelle même sur des modèles comportant un grand nombre de règles d'interaction.

Les informations trouvées par l'analyse statique sont utiles. Une même démarche a été suivie pour améliorer la qualité de ces modèles, qu'ils soient écrits à la main ou assemblés automatiquement par fouille automatique de la littérature. 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

modèle	nombre de règles	nombre de contraintes inférées	nombre de règles mortes détectées	temps d'analyse (secondes)
repressilator	42	0	0	0.010
egfr_net	39	0	0	0.020
egfr_net_red	45	0	0	0.023
fceri_fyn	46	0	0	0.049
fceri_fyn_lig	48	0	0	0.049
fceri_fyn_trimer	362	0	36	0.511
fceri_fyn_gamma2	59	0	0	0.062
fceri_fyn_ji	36	0	0	0.035
fceri_fyn_ji_red	32	0	0	0.034
fceri_fyn_lyn_745	40	0	2	0.040
fceri_fyn_trimer	192	0	0	0.271
sos	20	0	0	0.024
machine	220	0	7	0.687
ensemble	233	0	0	0.652
korkut (2017/01/13)	3916	0	1610	12
korkut (2017/01/17)	12896	0	874	35
korkut (2017/02/06)	5750	0	884	78
TGF (V19)	97	0	10	0.352
TGF (V20)	99	0	10	0.424
TGF (V21)	211	0	0	0.966
TGF (2017/04/01)	235	0	0	0.826
TGF (2018/04/19)	292	0	0	0.917
BigWnt (2015/12/28)	356	0	1	3.55
BigWnt (2016/09/28)	1419	0	0	11
BigWnt (2017/03/22)	1486	0	12	14

Figure 4.9: Résultats expérimentaux (calculés sur un MacBook Pro avec une puce Intel Core i7-6567U (cadencée 3.3 GHz)). Pour chaque modèle et chaque version, le nombre de règles est donné, ainsi que le nombre de contraintes découvertes par l'analyse et le nombre de règles mortes trouvées (qui ne sont donc jamais utilisées dans le modèle). Le temps total de l'analyse est également fourni.

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* (Feret et Lý 2016 ; Feret et Ly 2018) 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.

**Exemple 4.5.1** *Le système de transitions qui représente les traces locales des occurrences du récepteur membranaire dans le modèles des premières interactions qui interviennent dans l'activation du facteur de croissance de l'épiderme, est dessiné en Fig. 4.10. La succession entre les différentes configurations possibles y est clairement décrite. Ainsi, partant d'un récepteur avec les sites  $l$ ,  $r$ ,  $c$  et  $n$  libres (le site  $n$  a été ajouté pour rendre l'exemple plus intéressant), le site  $l$  peut devenir lié en premier, ensuite le site  $r$  peut devenir lié, ensuite soit le site  $c$ , soit le site  $n$  peut devenir lié. Par contre, ces deux liaisons sont exclusives : les sites  $c$  et  $n$  d'une occurrence de protéines ne peuvent pas être liés tous les deux simultanément. Par ailleurs, toutes les liaisons peuvent se défaire dans l'ordre inverse de leur création.*



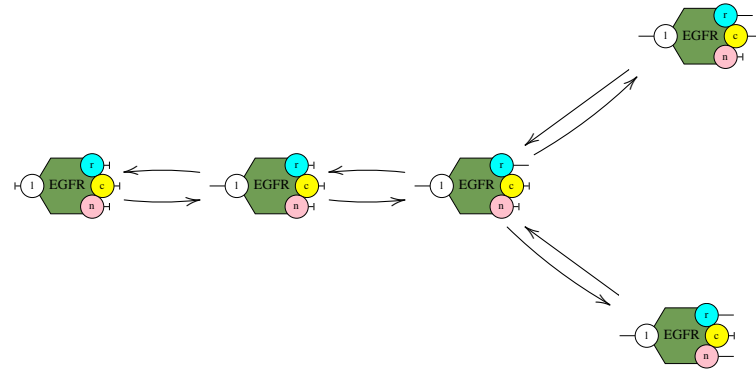


Figure 4.10: Le système de transitions pour les vues locales des occurrences du récepteur membranaire. L'état du site  $n$  a été ajouté pour rendre l'exemple plus intéressant. Ce système de transitions explique les étapes que traverse ces occurrences lorsque leurs sites deviennent liés. Les transitions en sens inverse, qui correspondent aux règles de libération des sites sont aussi représentées.



## Chapter 5

# Flot d'information dans la sémantique stochastique d'un modèle Kappa

### 5.1 Système stochastique sous-jacent à un modèle Kappa

### 5.2 Cas d'études

Les modèles d'interaction entre protéines souffrent d'une très grande complexité combinatoire. Les protéines peuvent se lier entre elles et modifier leurs états, ce qui conduit potentiellement à la formation de très grands complexes biochimiques. Ceci empêche toute description extensionnelle des systèmes dynamiques engendré par ces modèles, que ce soit dans un cadre différentiel ou stochastique.

Le chapitre précédent portait sur une approche qui permettait de réduire le nombre de variables nécessaire pour décrire la dynamique de ces modèles dans le cadre différentiel. En suivant le flot d'information potentiel entre les différentes parties des espèces biochimiques, elle consiste à identifier quelles corrélations entre états de sites d'interaction ont une importance sur la dynamique du système, puis à oublier les autres en découpant les espèces biochimiques en petite unité d'information, appelées fragments. Alors qu'elle donne des résultats prometteurs dans le cadre différentiel, le présent chapitre montre au contraire qu'il est beaucoup plus difficile de réduire la dimension des modèles stochastiques engendrés par un modèle de réécriture de graphes à sites.

Ce chapitre commence par la description formelle de la sémantique stochastique d'un modèle Kappa. En Sect. ?? sont fournis trois exemples pour illustrer en quoi l'approche décrite dans le chapitre ?? pour la réduction des systèmes différentiels engendrés ne peut pas s'appliquer directement pour réduire les systèmes stochastiques sous-jacents. En Sect. ?? ...

### 5.3 Sémantique stochastique d'un modèle de réécritures de graphes à sites

### 5.4 Cas d'étude

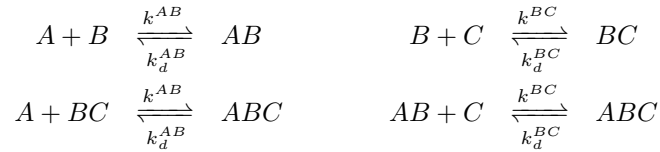
Trois cas d'études sont considérés. Le premier, qui est décrit en Sect. 5.4.1, fait intervenir une sorte de protéine pouvant se lier indépendamment à deux autres sortes de protéines. La première sorte de protéine peut alors être vue comme deux morceaux entièrement indépendant, ce qui conduit à une réduction du modèle correcte à la fois dans le cadre différentiel ou stochastique. Dans le second exemple, qui est donné en Sect. 5.4.2, il est montré que, par rapport à la réduction du système différentiel engendré par un modèle de réécriture de graphes à sites, des corrélations supplémentaires sont souvent nécessaires pour simuler fidèlement les systèmes stochastiques. Dans cette exemple, une règle de déliaison ne peut être simplifiée sans connaître précisément la distribution des différents états des deux protéines qu'elles séparent pour préserver la dynamique du système sous-jacent dans le cadre stochastique. Dans le cadre différentiel, cette information peut être oubliée tout en obtenant une réduction exacte du système engendré. Enfin, le troisième exemple, qui est expliqué en Sect. 5.4.3, montre l'existence de flot d'information entre différentes composantes connexes du membre gauche d'une règle, ce qui

une fois encore interdit certaines réductions qui sont pourtant correctes dans le cadre différentiel.

### 5.4.1 Un exemple d'indépendance entre deux liaisons

Voici un modèle dans lequel chaque occurrence d'une protéine peut être découpée en deux morceaux indépendants, ce qui permet de réduire la dimension de l'espace d'états de la chaîne de Markov sous-jacente.

Soit une protéine  $B$  avec deux sites de liaison  $a$  et  $c$ . Dans chaque occurrence de la protéine  $B$ , le site  $a$  peut se lier au site  $b$  des occurrences d'une autre protéine, appelée  $A$ , alors que le  $c$  peut se lier au site  $b$  des occurrences d'une troisième protéine, appelée  $C$ . Les règles de liaison et de déliaison sont dessinées en Fig. ??.



Les constantes de réactions pour les règles d'association et de dissociation sont indépendantes du fait que l'occurrence de la protéine  $B$  est lié, ou non, sur son autre site de liaison. C'est essentiel pour permettre la réduction du modèle.

La sémantique différentielle de ce modèle est défini par le système d'équations différentielles ordinaire suivant.

$$\begin{aligned}
 [A]' &= k_a^{AB}([AB] + [ABC]) - k_d^{AB}([B] + [BC])[A] \\
 [C]' &= k_c^{BC}([BC] + [ABC]) - k_d^{BC}([B] + [AB])[C] \\
 [B]' &= k_a^{AB}[A][B] + k_d^{BC}[BC] - (k_a^{AB}[A] + k_c^{BC}[C])[B] \\
 [AB]' &= k_a^{AB}[A][B] + k_d^{BC}[ABC] - (k_d^{AB} + k_c^{BC}[C])[AB] \\
 [BC]' &= k_c^{BC}[B][C] + k_d^{AB}[ABC] - (k_d^{BC} + k_a^{AB}[A])[BC] \\
 [ABC]' &= k_a^{AB}[A][BC] + k_c^{BC}[AB][C] - (k_d^{AB} + k_d^{BC})[ABC].
 \end{aligned}$$

Le modèle ayant peu de complexes biochimiques différents. Ceux-ci sont représentés par la liste des occurrences des protéines qui les constituent. Ainsi, les notations  $A$ ,  $B$  et  $C$  représentent les protéines correspondantes, alors que les notations  $AB$ ,  $BC$ ,  $ABC$  représentent les complexes formés d'une occurrence de la protéine  $B$  liée à une occurrence de la protéine  $A$ , d'une occurrence de la protéine  $B$  liée à une occurrence de la protéine  $C$  et d'une occurrence de la protéine  $B$  liée à une occurrence de la protéine  $A$  et à une occurrence de la protéine  $C$ .

Il est possible de passer sous silence la corrélation entre l'état des deux sites de liaisons des occurrences de la protéine  $B$ . Ainsi, les equations suivantes :

$$\begin{array}{ll}
 [A]' &= k_a^{AB}[AB?] - k_d^{AB}[A][B?] \\
 [B?]' &= k_a^{AB}[AB?] - k_d^{AB}[A][B?] \\
 [AB?]' &= k_a^{AB}[A][B?] - k_d^{AB}[AB?] \\
 [C]' &= k_c^{BC}[?BC] - k_d^{BC}[?B][C] \\
 [?B]' &= k_c^{BC}[?BC] - k_d^{BC}[?B][C] \\
 [?BC]' &= k_c^{BC}[?B][C] - k_d^{BC}[?BC]
 \end{array}$$

où<sup>1</sup>  $[AB?] := [AB] + [ABC]$ ,  $[B?] := [B] + [BC]$ ,  $[?BC] := [BC] + [ABC]$ , et  $[?B] := [B] + [AB]$ , sont satisfaites.

Est-ce que cette réduction est, elle-aussi, exacte dans le système stochastique engendré par les règles de cet exemple ? Il faut regarder ce qui se passe au niveau de l'évolution de la distribution des différents états du système. L'état du système peut être représenté par un sextuplet d'entiers naturels  $\langle n_A, n_B, n_C, n_{AB}, n_{BC}, n_{ABC} \rangle$  où la composante  $n_X$  représente le nombre d'occurrences du complexe biochimique  $X$  dans cet état, pour n'importe quel complexe biochimique  $X$  du modèle. La probabilité  $P_t(\sigma)$  que le système soit dans l'état  $\sigma$  au

<sup>1</sup>Un point d'interrogation '?' à la gauche (resp. à la droite) de la lettre  $B$  indique que de savoir si l'occurrence correspondante de la protéine  $B$  est liée, ou non, à une occurrence de la protéine  $A$  (resp.  $C$ ) n'a pas d'importance.

temps  $t$  est alors donnée par l'équation maîtresse suivante :

$$\begin{aligned}
P_t(\langle n_A, n_B, n_C, n_{AB}, n_{BC}, n_{ABC} \rangle)' = & \\
& k^{AB}(n_A + 1)(n_B + 1)P_t(\langle n_A + 1, n_B + 1, n_C, n_{AB} - 1, n_{BC}, n_{ABC} \rangle) \\
& + k_d^{AB}(n_{AB} + 1)P_t(\langle n_A - 1, n_B - 1, n_C, n_{AB} + 1, n_{BC}, n_{ABC} \rangle) \\
& + k^{AB}(n_A + 1)(n_{BC} + 1)P_t(\langle n_A + 1, n_B, n_C, n_{AB}, n_{BC} + 1, n_{ABC} - 1 \rangle) \\
& + k_d^{AB}(n_{ABC} + 1)P_t(\langle n_A - 1, n_B, n_C, n_{AB}, n_{BC} - 1, n_{ABC} + 1 \rangle) \\
& + k^{BC}(n_B + 1)(n_C + 1)P_t(\langle n_A, n_B + 1, n_C + 1, n_{AB}, n_{BC} - 1, n_{ABC} \rangle) \\
& + k_d^{BC}(n_{BC} + 1)P_t(\langle n_A, n_B - 1, n_C - 1, n_{AB}, n_{BC} + 1, n_{ABC} \rangle) \\
& + k^{BC}(n_{AB} + 1)(n_C + 1)P_t(\langle n_A, n_B, n_C + 1, n_{AB} + 1, n_{BC}, n_{ABC} - 1 \rangle) \\
& + k_d^{BC}(n_{ABC} + 1)P_t(\langle n_A, n_B, n_C - 1, n_{AB} - 1, n_{BC}, n_{ABC} + 1 \rangle) \\
& - k^{AB}n_A(n_B + n_{BC})P_t(\langle n_A, n_B, n_C, n_{AB}, n_{BC}, n_{ABC} \rangle) \\
& - k_d^{AB}(n_{AB} + n_{ABC})P_t(\langle n_A, n_B, n_C, n_{AB}, n_{BC}, n_{ABC} \rangle) \\
& - k^{BC}(n_B + n_{AB})n_CP_t(\langle n_A, n_B, n_C, n_{AB}, n_{BC}, n_{ABC} \rangle) \\
& - k_d^{BC}(n_{BC} + n_{ABC})P_t(\langle n_A, n_B, n_C, n_{AB}, n_{BC}, n_{ABC} \rangle)
\end{aligned}$$

Dans cet exemple particulier, il est possible d'ignorer la corrélation entre l'état des deux sites des occurrences de la protéine  $B$  également dans la sémantique stochastique. Pour cela, chaque état est abstrait en un triplet  $\langle n_A, n_{B?}, n_{AB?} \rangle$ , dont la première composante,  $n_A$ , représente le nombre d'occurrences de la protéine  $A$  isolée, la seconde composante,  $n_{B?}$ , représente le nombre des occurrences de la protéine  $B$  qui ne sont pas liées à une occurrence de la protéine  $A$ , et la troisième composante,  $n_{AB?}$ , le nombre d'occurrences de la protéine  $B$  qui sont liées à une occurrence de la protéine  $A$ .

Pour cela, chaque état  $\sigma = \langle n_A, n_B, n_C, n_{AB}, n_{BC}, n_{ABC} \rangle$  est abstrait en un triplet  $\beta^A(\sigma) = \langle n_A, n_B + n_{BC}, n_{AB} + n_{ABC} \rangle$ , and by  $\beta^C(\sigma)$  the triple  $\langle n_C, n_B + n_{AB}, n_{BC} + n_{ABC} \rangle$ . The probability  $P_t^A(\sigma^A)$  that the system is in a state  $\sigma$  such that  $\beta^A(\sigma) = \sigma^A$  at time  $t$ , and the probability  $P_t^C(\sigma^C)$  that the system is in a state  $\sigma$  such that  $\beta^C(\sigma) = \sigma^C$  at time  $t$ , satisfy the following master equations:

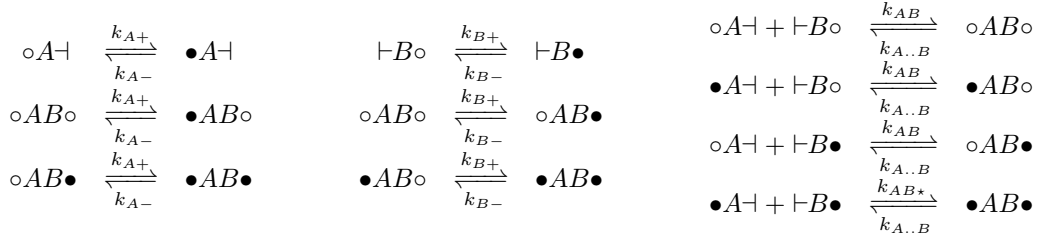
$$\begin{aligned}
P_t^A(\langle n_A, n_{B?}, n_{AB?} \rangle)' = & \\
& k^{AB}(n_A + 1)(n_{B?} + 1)P_t^A(\langle n_A + 1, n_{B?} + 1, n_{AB?} - 1 \rangle) \\
& + k_d^{AB}(n_{AB?} + 1)P_t^A(\langle n_A - 1, n_{B?} - 1, n_{AB?} + 1 \rangle) \\
& - (k^{AB}n_An_{B?} + k_d^{AB}n_{AB?})P_t^A(\langle n_A, n_{B?}, n_{AB?} \rangle) \\
\\
P_t^C(\langle n_C, n_{?B}, n_{?BC} \rangle)' = & \\
& k^{BC}(n_{?B} + 1)(n_C + 1)P_t^C(\langle n_C + 1, n_{?B} + 1, n_{?BC} - 1 \rangle) \\
& + k_d^{BC}(n_{?BC} + 1)P_t^C(\langle n_C - 1, n_{?B} - 1, n_{?BC} + 1 \rangle) \\
& - (k^{BC}n_{?B}n_C + k_d^{BC}n_{?BC})P_t^C(\langle n_C, n_{?B}, n_{?BC} \rangle).
\end{aligned}$$

## 5.4.2 Un exemple avec une déliaison inconditionnelle

Le fait que les occurrences d'un complexe biochimique puisse être séparées en deux parties entièrement indépendantes (comme c'était le cas dans l'exemple précédent) est assez rare en pratique. Ce nouvel exemple contient une règle de déliaison inconditionnelle entre deux occurrences de protéines pouvant prendre plusieurs états. Dans la sémantique différentielle, cette règle n'induit pas de flot d'information et le système d'équations différentielles sous-jacent peut être réduit. En stochastique, la corrélation entre l'état des occurrences de protéines qui sont liées entre-elles a un impact sur la distribution de l'état des monomères. Ceci induit un flux d'information qui empêche de réduire de manière exacte le système stochastique sous-jacent.

Soit  $A$  et  $B$  deux sortes de protéine. Chaque occurrence de ces protéines peut être phosphorylée, ou non. De plus, une occurrence de la protéine  $A$  peut se lier à une occurrence de la protéine  $B$  pour former un dimère.

Le comportement de ce modèle peut être décrit par les règles de réécriture données en Fig. ??.



En particulier, il y a plusieurs constantes de réaction pour les règles de liaison (voir en Fig. ??), selon l'état de phosphorylation des occurrences de protéines qui se lient.

Hormi cette règle de liaison, les interactions potentielles sont purement locales. Ainsi, les constantes de réaction pour la phosphorylation des occurrences de la protéine  $A$  (voir en Fig. ??) et de la protéine  $B$  (voir en Fig. ??) sont indépendantes du fait que ces occurrences appartiennent à des dimères ou non, et la constante de réaction pour la déliaison de deux occurrences de protéine est indépendante de l'état de phosphorylation de ces deux occurrences de protéine (voir en Fig. ??). Par contre, il existe plusieurs règles de liaison (voir en Fig. ??), avec des constantes de réaction pouvant varier selon l'état de phosphorylation des deux occurrences de protéine qui se lient. Par exemple, si la constante de réaction  $k_{AB*}$  est choisie plus grande que la constante de réaction  $k_{AB}$ , alors la formation de dimère est facilitée quand les deux occurrences des protéines qui se lient sont préalablement phosphorylée. En conséquence, l'état de phosphorylation des occurrences de protéine au sein des dimères n'est pas indépendante. Lorsque l'occurrence de la protéine  $A$  est phosphorylée, il y a plus de chance que l'occurrence de la protéine  $B$  le soit également.

Il est tentant de vouloir abstraire cette corrélation. Cela reviendrait à oublier quelles occurrences de protéine sont liées entre-elles, et à suivre indépendamment les occurrences de protéines en ne gardant que leur état de phosphorylation et de liaison. Tous les interactions se traduisent directement à ce niveau d'abstraction, sauf celle de déliaison. En effet, une déliaison fait intervenir deux occurrences de protéines liées entre-elles. Comme les liaisons ne sont plus représentées. Du coup, il semble naturel de représenter les règles de liaisons par deux règles indépendantes, l'une pour libérer les occurrences liées de la protéine  $A$  et l'autre pour libérer les occurrences liées de la protéine  $B$  (ici l'état de liaison peut être assimilé à un état interne puisque l'information de savoir quelle occurrence de protéine est liée à quelle autre, n'est pas maintenue).

Quelles sont les conséquences de cette modification des règles du modèles quant à la dynamique des systèmes engendrés ?

La sémantique différentielle du modèle initial est définie par le système d'équations suivant :

$$\begin{aligned}
[\circ A \dashv]' &= k_{A-}[\bullet A \dashv] + k_{A..B}([\circ AB \circ] + [\circ AB \bullet]) - (k_{A+} + k_{AB}([\vdash B \circ] + [\vdash B \bullet]))[\circ A \dashv] \\
[\bullet A \dashv]' &= k_{A+}[\circ A \dashv] + k_{A..B}([\bullet AB \circ] + [\bullet AB \bullet]) - (k_{A-} + k_{AB}[\vdash B \circ] + k_{AB*}[\vdash B \bullet])[\bullet A \dashv] \\
[\vdash B \circ]' &= k_{B-}[\vdash B \bullet] + k_{A..B}([\circ AB \circ] + [\bullet AB \circ]) - (k_{B+} + k_{AB}([\circ A \dashv] + [\bullet A \dashv]))[\vdash B \circ] \\
[\vdash B \bullet]' &= k_{B+}[\vdash B \circ] + k_{A..B}([\circ AB \bullet] + [\bullet AB \bullet]) - (k_{B-} + k_{AB}[\circ A \dashv] + k_{AB*}[\bullet A \dashv])[\vdash B \bullet] \\
[\circ AB \circ]' &= k_{A-}[\bullet AB \circ] + k_{B-}[\circ AB \bullet] + k_{AB}[\circ A \dashv][\vdash B \circ] - (k_{A+} + k_{B+} + k_{A..B})[\circ AB \circ] \\
[\bullet AB \circ]' &= k_{A+}[\circ AB \circ] + k_{B-}[\bullet AB \bullet] + k_{AB}[\bullet A \dashv][\vdash B \circ] - (k_{A-} + k_{B+} + k_{A..B})[\bullet AB \circ] \\
[\circ AB \bullet]' &= k_{A-}[\bullet AB \bullet] + k_{B+}[\circ AB \circ] + k_{AB}[A][B^*] - (k_{A+} + k_{B-} + k_{A..B})[\circ AB \bullet] \\
[\bullet AB \bullet]' &= k_{A+}[\circ AB \bullet] + k_{B+}[\bullet AB \circ] + k_{AB*}[\bullet A \dashv][\vdash B \bullet] - (k_{A-} + k_{B-} + k_{A..B})[\bullet AB \bullet].
\end{aligned}$$

Dans ce système, la concentration en monomère de la protéine  $A$  non phosphorylée est notée  $[\circ A \dashv]$  et celle de la protéine phosphorylée est notée  $[\bullet A \dashv]$ . La concentration en monomère de la protéine  $B$  non phosphorylée est notée  $[\vdash B \circ]$  et celle de la protéine phosphorylée est notée  $[\vdash B \bullet]$ . La concentration en dimère entièrement non phosphorylé est notée  $[\circ AB \circ]$ , la concentration en dimère avec l'occurrence de la protéine  $A$  phosphorylé, et non l'occurrence de la protéine  $B$ , est notée  $[\bullet AB \circ]$ , la concentration en dimère avec l'occurrence de la protéine  $B$  phosphorylé, et non l'occurrence de la protéine  $A$ , est notée  $[\circ AB \bullet]$  et la concentration en dimère doublement phosphorylé est notée  $[\bullet AB \bullet]$ .

Le modèle simplifié engendre lui le système différentiel suivant :

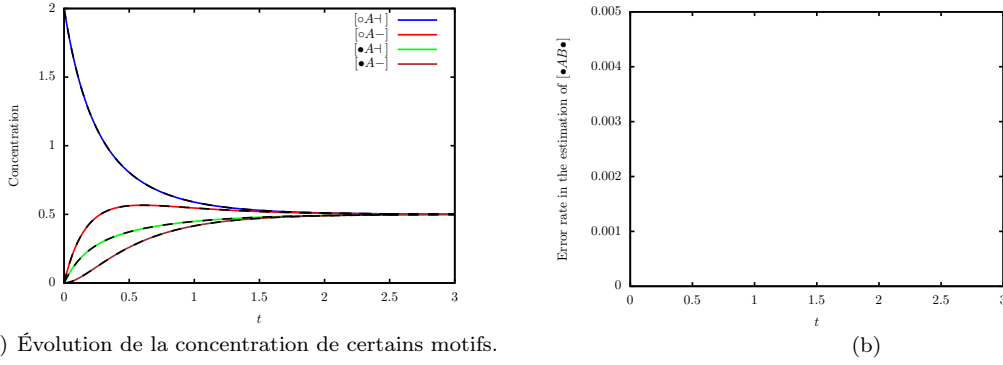


Figure 5.1: À gauche, l'évolution de la concentration des différentes configuration de la protéine  $A$ , qu'elle soit phosphorylée, ou non, et quelle soit liée, ou non. Les courbes colorées sont obtenues en simulant numériquement le système d'équations initial, alors que les courbes le sont à partir du système d'équations réduit. Ce taux est défini comme la différence des deux valeurs (dans le système initial et dans le système réduit) divisé par la plus grande de ces deux valeurs. Toutes les constantes de réaction sont fixées à 1, sauf la constante  $k_{AB*}$  qui est fixée à 10. À l'origine, la concentration en monomère non phosphorylé de la protéine  $A$  en monomère non phosphorylé de la protéine  $B$  sont toutes deux fixées à 2 ; la concentration des autres composants potentiels est fixée à 0.

$$\begin{aligned}
[\circ A+] &= k_{A-}[\bullet A+] + k_{A..B}[\circ A-] - (k_{A+} + k_{AB}([\neg B\circ] + [\neg B\bullet]))[\circ A+] \\
[\bullet A+] &= k_{A+}[\circ A+] + k_{A..B}[\bullet A-] - (k_{A-} + k_{AB}[\neg B\circ] + k_{AB*}[\neg B\bullet])[\bullet A+] \\
[\neg B\circ] &= k_{B-}[\neg B\bullet] + k_{A..B}[\neg B\circ] - (k_{B+} + k_{AB}([\circ A+] + [\bullet A+]))[\neg B\circ] \\
[\neg B\bullet] &= k_{B+}[\neg B\circ] + k_{A..B}[\neg B\bullet] - (k_{B-} + k_{AB}[\circ A+] + k_{AB*}[\bullet A+])[\neg B\bullet] \\
[\circ A-] &= k_{A-}[\bullet A-] + k_{AB}[\circ A+](\neg B\circ] + [\neg B\bullet]) - (k_{A+} + k_{A..B})[\circ A-] \\
[\bullet A-] &= k_{A+}[\circ A-] + k_{AB}[\bullet A+](\neg B\circ] + k_{AB*}[\bullet A+](\neg B\bullet] - (k_{A-} + k_{A..B})[\bullet A-] \\
[\neg B\circ] &= k_{B-}[\neg B\bullet] + k_{AB}[\neg B\circ](\circ A+] + [\bullet A+]) - (k_{B+} + k_{A..B})[\neg B\circ] \\
[\neg B\bullet] &= k_{B+}[\neg B\circ] + k_{AB}[\circ A+](\neg B\bullet] + k_{AB*}[\bullet A+](\neg B\bullet] - (k_{B-} + k_{A..B})[\neg B\bullet],
\end{aligned}$$

dans lequel<sup>2</sup>  $[\circ A-]$  représente la concentration en protéine  $A$  liée et non phosphorylée,  $[\bullet A-]$  la concentration en protéine  $A$  liée et phosphorylée,  $[\neg B\circ]$  la concentration en protéine  $B$  liée et non phosphorylée et  $[\neg B\bullet]$  la concentration en protéine  $B$  liée et phosphorylée.

Le système d'équations engendré par le modèle initial et celui engendré par le modèle simplifié sont liés formellement par les contraintes suivantes :

$$\begin{cases}
[\circ A-] := [AB] + [\circ AB\bullet], \\
[\bullet A-] := [\bullet AB\circ] + [\bullet AB\bullet], \\
[\neg B\circ] := [AB] + [\bullet AB\circ], \\
[\neg B\bullet] := [\circ AB\bullet] + [\bullet AB\bullet].
\end{cases}$$

qui sont obtenues en prenant la définition extensionnelle des parties de complexes biochimiques.

$$\begin{aligned}
[\circ A+] &= k_{A-}[\bullet A+] + k_{A..B}[\circ A-] - (k_{A+} + k_{AB}([\neg B\circ] + [\neg B\bullet]))[\circ A+] \\
[\bullet A+] &= k_{A+}[\circ A+] + k_{A..B}[\bullet A-] - (k_{A-} + k_{AB}[\neg B\circ] + k_{AB*}[\neg B\bullet])[\bullet A+] \\
[\neg B\circ] &= k_{B-}[\neg B\bullet] + k_{A..B}[\neg B\circ] - (k_{B+} + k_{AB}([\circ A+] + [\bullet A+]))[\neg B\circ] \\
[\neg B\bullet] &= k_{B+}[\neg B\circ] + k_{A..B}[\neg B\bullet] - (k_{B-} + k_{AB}[\circ A+] + k_{AB*}[\bullet A+])[\neg B\bullet] \\
[\circ A-] &= k_{A-}[\bullet A-] + k_{AB}[\circ A+](\neg B\circ] + [\neg B\bullet]) - (k_{A+} + k_{A..B})[\circ A-] \\
[\bullet A-] &= k_{A+}[\circ A-] + k_{AB}[\bullet A+](\neg B\circ] + k_{AB*}[\bullet A+](\neg B\bullet] - (k_{A-} + k_{A..B})[\bullet A-] \\
[\neg B\circ] &= k_{B-}[\neg B\bullet] + k_{AB}[\neg B\circ](\circ A+] + [\bullet A+]) - (k_{B+} + k_{A..B})[\neg B\circ] \\
[\neg B\bullet] &= k_{B+}[\neg B\circ] + k_{AB}[\circ A+](\neg B\bullet] + k_{AB*}[\bullet A+](\neg B\bullet] - (k_{B-} + k_{A..B})[\neg B\bullet],
\end{aligned}$$

<sup>2</sup>L'exposant  $\diamond$  représente un état de phosphorylation non spécifié.

where<sup>3</sup>  $[\circ A -] := [AB] + [\circ AB \bullet]$ ,  $[\bullet A -] := [\bullet AB \circ] + [\bullet AB \bullet]$ ,  $[-B \circ] := [AB] + [\bullet AB \circ]$ , and  $[-B \bullet B] := [\circ AB \bullet] + [\bullet AB \bullet]$ , are satisfied.

Yet, this correlation forbids the reduction of the stochastic semantics. Let us explain why. In the stochastic semantics, a chemical soup can be denoted by a 8-tuple  $\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle$  of natural numbers, where  $n_X$  is the number of instance of  $X$  in the chemical soup, for any  $X \in \{A, A^*, B, B^*, AB, \bullet AB \circ, \circ AB \bullet, \bullet AB \bullet\}$ . The probability  $P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle)$  that the system is in a given state  $\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle$  at time  $t$  is given by the following master equation:

$$\begin{aligned}
P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle)' = & \\
& k_{A+}(n_A + 1)P_t(\langle n_A + 1, n_{A^*} - 1, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{A+}(n_{AB} + 1)P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB} + 1, n_{\bullet AB \circ} - 1, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{A+}(n_{\circ AB \bullet} + 1)P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet} + 1, n_{\bullet AB \bullet} - 1 \rangle) \\
& + k_{A-}(n_{A^*} + 1)P_t(\langle n_A - 1, n_{A^*} + 1, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{A-}(n_{\bullet AB \circ} + 1)P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB} - 1, n_{\bullet AB \circ} + 1, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{A-}(n_{\bullet AB \bullet} + 1)P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet} - 1, n_{\bullet AB \bullet} + 1 \rangle) \\
& + k_{B+}(n_B + 1)P_t(\langle n_A, n_{A^*}, n_B + 1, n_{B^*} - 1, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{B+}(n_{AB} + 1)P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB} + 1, n_{\bullet AB \circ}, n_{\circ AB \bullet} - 1, n_{\bullet AB \bullet} \rangle) \\
& + k_{B+}(n_{\bullet AB \circ} + 1)P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ} + 1, n_{\circ AB \bullet}, n_{\bullet AB \bullet} - 1 \rangle) \\
& + k_{B-}(n_{B^*} + 1)P_t(\langle n_A, n_{A^*}, n_B - 1, n_{B^*} + 1, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{B-}(n_{\circ AB \bullet} + 1)P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB} - 1, n_{\bullet AB \circ}, n_{\circ AB \bullet} + 1, n_{\bullet AB \bullet} \rangle) \\
& + k_{B-}(n_{\bullet AB \bullet} + 1)P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ} - 1, n_{\circ AB \bullet}, n_{\bullet AB \bullet} + 1 \rangle) \\
& + k_{AB}(n_A + 1)(n_B + 1)P_t(\langle n_A + 1, n_{A^*}, n_B + 1, n_{B^*}, n_{AB} - 1, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{AB}(n_A + 1)(n_{B^*} + 1)P_t(\langle n_A + 1, n_{A^*}, n_B, n_{B^*} + 1, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet} - 1, n_{\bullet AB \bullet} \rangle) \\
& + k_{AB}((n_{A^*} + 1)(n_B + 1))P_t(\langle n_A, n_{A^*} + 1, n_B + 1, n_{B^*}, n_{AB}, n_{\bullet AB \circ} - 1, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{AB^*}((n_{A^*} + 1)(n_{B^*} + 1))P_t(\langle n_A, n_{A^*} + 1, n_B, n_{B^*} + 1, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} - 1 \rangle) \\
& \text{(continued on the next page)}
\end{aligned}$$

$$\begin{aligned}
& + k_{A..B}(n_{AB} - 1)P_t(\langle n_A - 1, n_{A^*} - 1, n_B, n_{B^*}, n_{AB} + 1, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{A..B}(n_{\circ AB \bullet} - 1)P_t(\langle n_A - 1, n_{A^*}, n_B, n_{B^*} - 1, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet} + 1, n_{\bullet AB \bullet} \rangle) \\
& + k_{A..B}(n_{\bullet AB \circ} - 1)P_t(\langle n_A, n_{A^*} - 1, n_B - 1, n_{B^*}, n_{AB}, n_{\bullet AB \circ} + 1, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& + k_{A..B}(n_{\bullet AB \bullet} - 1)P_t(\langle n_A, n_{A^*} - 1, n_B, n_{B^*} - 1, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} + 1 \rangle) \\
& - k_{A+}(n_A + n_{AB} + n_{\circ AB \bullet})P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& - k_{A-}(n_{A^*} + n_{\bullet AB \circ} + n_{\bullet AB \bullet})P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& - k_{B+}(n_B + n_{AB} + n_{\bullet AB \circ})P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& - k_{B-}(n_{B^*} + n_{AB^*} + n_{\bullet AB \bullet})P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& - k_{AB}((n_A + n_{A^*})(n_B + n_{B^*}) - n_{A^*}n_{B^*})P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& - k_{AB^*}n_{A^*}n_{B^*}P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle) \\
& - k_{A..B}(n_{AB} + n_{\circ AB \bullet} + n_{\bullet AB \circ} + n_{\bullet AB \bullet})P_t(\langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle)
\end{aligned}$$

As in the example of Sect. 5.4.1, we would like to abstract away the correlation between the phosphorylation state of the proteins  $A$  and the phosphorylation state of the proteins  $B$  which belong to the same complex. Given a state  $\sigma = \langle n_A, n_{A^*}, n_B, n_{B^*}, n_{AB}, n_{\bullet AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} \rangle$ , we denote by  $\beta(\sigma)$  the 8-tuple  $\langle n_{\circ A \dagger}, n_{\bullet A \dagger}, n_{\dagger B \circ}, n_{\dagger B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle$  (such a tuple is called an abstract state). The probability

<sup>3</sup>The superscript  $\diamond$  stands for “whatever the phosphorylation state is”.



$P_t^\sharp(\sigma^\sharp)$  that the system is in a state  $\sigma$  such that  $\beta(\sigma) = \sigma^\sharp$  at time  $t$ , satisfies the following equation:

$$\begin{aligned}
P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle)' = & \\
& k_{A+}(n_A + 1)P_t^\sharp(\langle n_{\circ A} + 1, n_{\bullet A} - 1, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& + k_{A+}(n_{\circ A -} + 1)P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -} + 1, n_{\bullet A -} - 1, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& + k_{A-}(n_{A^*} + 1)P_t^\sharp(\langle n_{\circ A} - 1, n_{\bullet A} + 1, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& + k_{A-}(n_{\bullet A -} + 1)P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -} - 1, n_{\bullet A -} + 1, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& + k_{B+}(n_B + 1)P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ} + 1, n_{\vdash B \bullet} - 1, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& + k_{B+}(n_{-B \circ} + 1)P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ} + 1, n_{-B \bullet B} - 1 \rangle) \\
& + k_{B-}(n_{B^*} + 1)P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ} - 1, n_{\vdash B \bullet} + 1, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& + k_{B-}(n_{-B \bullet B} + 1)P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ} - 1, n_{-B \bullet B} + 1 \rangle) \\
& + k_{AB}(n_A + 1)(n_B + 1)P_t^\sharp(\langle n_{\circ A} + 1, n_{\bullet A}, n_{\vdash B \circ} + 1, n_{\vdash B \bullet}, n_{\circ A -} - 1, n_{\bullet A -}, n_{-B \circ} - 1, n_{-B \bullet B} \rangle) \\
& + k_{AB}(n_A + 1)(n_{B^*} + 1)P_t^\sharp(\langle n_{\circ A} + 1, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet} + 1, n_{\circ A -} - 1, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} - 1 \rangle) \\
& + k_{AB}((n_{A^*} + 1)(n_B + 1))P_t^\sharp(\langle n_{\circ A}, n_{\bullet A} + 1, n_{\vdash B \circ} + 1, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -} - 1, n_{-B \circ} - 1, n_{-B \bullet B} \rangle) \\
& + k_{AB^*}((n_{A^*} + 1)(n_{B^*} + 1))P_t^\sharp(\langle n_{\circ A}, n_{\bullet A} + 1, n_{\vdash B \circ}, n_{\vdash B \bullet} + 1, n_{\circ A -}, n_{\bullet A -} - 1, n_{-B \circ}, n_{-B \bullet B} - 1 \rangle) \\
& + k_{A..B}\tilde{E}_t(n_{AB} \mid \langle n_{\circ A} - 1, n_{\bullet A}, n_{\vdash B \circ} - 1, n_{\vdash B \bullet}, n_{\circ A -} + 1, n_{\bullet A -}, n_{-B \circ} + 1, n_{-B \bullet B} \rangle) \\
& + k_{A..B}\tilde{E}_t(n_{\circ AB \bullet} \mid \langle n_A - 1, n_{A^*}, n_B, n_{B^*} - 1, n_{AB} + 1, n_{\circ AB \circ}, n_{\circ AB \bullet}, n_{\bullet AB \bullet} + 1 \rangle) \\
& + k_{A..B}\tilde{E}_t(n_{\bullet AB \circ} \mid \langle n_A, n_{A^*} - 1, n_B - 1, n_{B^*}, n_{AB}, n_{\circ AB \circ} + 1, n_{\circ AB \bullet} + 1, n_{\bullet AB \bullet} \rangle) \\
& + k_{A..B}\tilde{E}_t(n_{\bullet AB \bullet} \mid \langle n_A, n_{A^*} - 1, n_B, n_{B^*} - 1, n_{AB}, n_{\circ AB \circ} + 1, n_{\circ AB \bullet}, n_{\bullet AB \bullet} + 1 \rangle) \\
& - k_{A+}(n_A + n_{\circ A -})P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& - k_{A-}(n_{A^*} + n_{\bullet A -})P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& - k_{B+}(n_B + n_{-B \circ})P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& - k_{B-}(n_{B^*} + n_{-B \bullet B})P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& - k_{AB}((n_A + n_{A^*})(n_B + n_{B^*}) - n_{A^*}n_{B^*})P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& - k_{AB^*}(n_{A^*}n_{B^*})P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) \\
& - k_{A..B}(n_{\bullet A -} + n_{\circ A -})P_t^\sharp(\langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle),
\end{aligned}$$

where for any expression  $X(\sigma)$  and any (abstract) state  $\sigma^\sharp$ , the expression  $\tilde{E}_t(X(\sigma) \mid \sigma^\sharp)$  denotes the product between the conditional expectation  $E_t(X(\sigma) \mid \sigma^\sharp)$  of the expression  $X(\sigma)$  knowing that  $\beta(\sigma) = \sigma^\sharp$ , and the probability  $P_t^\sharp(\sigma^\sharp)$  of being in a state  $\sigma$  such that  $\beta(\sigma) = \sigma^\sharp$ .

Whenever  $k_{AB} = k_{AB^*}$ , we can check that the fact that  $P_t(\sigma) = P_t(\sigma')$  for any pair of states  $\sigma, \sigma'$  such that  $\beta(\sigma) = \beta(\sigma')$  is an invariant. Thus, provided that  $k_{AB} = k_{AB^*}$  and that there is no correlation between the phosphorylation state of the proteins  $A$  and  $B$  which are bound together at time  $t = 0$ , one can use the following properties:

$$\begin{aligned}
E_t(n_{AB} \mid \langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) &= \frac{n_{\circ A} - n_{-B \circ}}{\circ A - + \bullet A -} \\
E_t(n_{\circ AB \bullet} \mid \langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) &= \frac{n_{\circ A} - n_{-B \bullet B}}{\circ A - + \bullet A -} \\
E_t(n_{\bullet AB \circ} \mid \langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) &= \frac{n_{\bullet A} - n_{-B \circ}}{\circ A - + \bullet A -} \\
E_t(n_{\bullet AB \bullet} \mid \langle n_{\circ A}, n_{\bullet A}, n_{\vdash B \circ}, n_{\vdash B \bullet}, n_{\circ A -}, n_{\bullet A -}, n_{-B \circ}, n_{-B \bullet B} \rangle) &= \frac{n_{\bullet A} - n_{-B \bullet B}}{\circ A - + \bullet A -},
\end{aligned}$$

so as to write conditional expectations of  $n_{AB}$ ,  $n_{\circ AB \bullet}$ ,  $n_{\bullet AB \circ}$ , and  $n_{\bullet AB \bullet}$  as time-independent expressions of  $n_{\circ A -}$ ,  $n_{\bullet A -}$ ,  $n_{-B \circ}$ , and  $n_{-B \bullet B}$ .

Whenever  $k_{AB} \neq k_{AB^*}$ , these conditional expectations may be time-dependent. We show in Fig. 5.2(a) that the ratio between the probability of being in the state  $AB + \bullet AB \bullet$  and the probability of being in the state  $\circ AB \bullet + \bullet AB \circ$  is time-dependent. Moreover, we show in Fig. 5.2(b) that the conditional expectation of  $n_{\bullet AB \bullet}$  knowing that we are in the (abstract) state  $\circ A - + \bullet A - + - B \circ + - B \bullet B$  is time-dependent as well, which forbids doing the same simplification as in the differential semantics.

We have seen through this example that some reactions may operate simultaneously over two fragments. This leads to coupled semi-reactions. We have noticed that coupled semi-reactions raise no issue when reducing the differential semantics. We say that the application of semi-rules is fair in the differential semantics, since the proportion of the concentration of a given fragment that is consumed by a semi-reaction does not depend on the correlation between the states of the two fragments. This is not the case in the stochastic semantics: we have noticed that the stochastic semantics can be reduced only if the state of the two fragments are not correlated, otherwise the choice of the fragments on which coupled semi-reactions operate is entangled, which forbids the

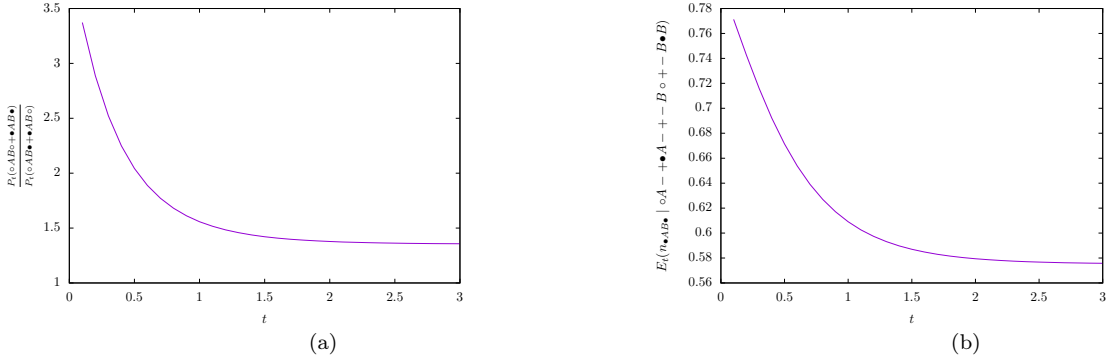


Figure 5.2: On the left, quotient between the probability of being in the state  $AB + \bullet AB \bullet$  and the probability of being in the state  $\circ AB \bullet + \bullet AB \circ$ . On the right, conditional expectation of the number of fully phosphorylated complexes  $\bullet AB \bullet$  knowing that all proteins are bound, and that there is exactly one phosphorylated protein  $A$  and exactly one phosphorylated  $B$ . All rates are set to 1, except  $k_{AB\star}$  which is set to 10. At time 0, the chemical soup is made of two proteins  $A$  and two proteins  $B$ , none of these proteins being phosphorylated or bound.

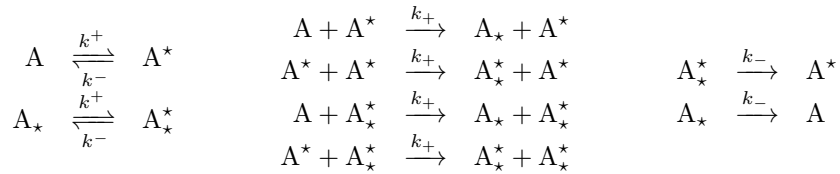
reduction. In other words, we say that in the differential semantics, we can abstract away the correlations which are not observed by rules, whereas in the stochastic semantics, we have to prove that the rules cannot enforce correlations between the state of some fragments and we use this property so as to reduce the dimension of the state space of the system. In the later case, the reduction is only valid when there is no correlation at time  $t = 0$ .

### 5.4.3 An example of distant control

In this section, we show that, in the stochastic semantics, one protein can control the behavior of another one even if they are not in the same connected component in the left hand side of a reaction.

We consider a kind of proteins,  $A$ , which bears two phosphorylation sites. Each phosphorylation site can be unphosphorylated, or phosphorylated. We use the symbol  $\star$  to denote phosphorylated sites. The phosphorylation state of the first site is written as a superscript, whereas the phosphorylation state of the second site is written as a subscript. This way, a protein  $A$  having the first site phosphorylated and the second site unphosphorylated is denoted by  $A^\star$ .

The behavior of a chemical soup can be described by the following set of reactions:



We have assumed (see second column) that the kinetic of the phosphorylation of the second site of a protein depends on the number of the other proteins that are phosphorylated on their first site — that is to say that the proteins that are phosphorylated on their first site catalyzes the phosphorylation of the second site in the other proteins. We have also assumed that other reactions are purely local, that is to say that the kinetic of phosphorylation and dephosphorylation on the first site does not depend on the phosphorylation state of the second site (neither of the protein being phosphorylated, nor of the other proteins) (see first column), and that the kinetic of dephosphorylation of the second site does not depend on the phosphorylation state of the first site of the proteins in the soup (see third column).

In this example, we would like to abstract the correlation between the phosphorylation state of the two sites of each protein. This could be achieved, by splitting each complex into two parts, and by abstracting away which parts are connected together. It raises an issue for reducing the stochastic semantics. Indeed, one can notice that the reaction which activates the second site of protein favors the phosphorylation of the second site of the protein in the state  $A$ . For instance, if we assume that both the number of instances of the protein in state

A and the number of instances of the protein in the state  $A^*$  is equal to  $m$ , and that the number of instances of the protein in the state  $A_\star^*$  is equal to  $n$ . Then, the cumulative activity of the following two reactions:



is equal to  $k_+(n+m)m$ , whereas the cumulative activity of the following two reactions:



is equal to  $k_+(n+m-1)m$  (the subtraction by 1 comes from the fact that each reactant must be mapped to distinct instances of chemical species). Nevertheless, it does not forbid the reduction of the differential semantics: intuitively, the term 1 vanishes because we consider an infinite number of instances, within an infinite volume.

Let us check formally that the differential semantics of this model can be reduced and explain why we do not know how to abstract its stochastic semantics. This differential semantics is defined by the following system of differential equations:

$$\begin{aligned} [A]' &= k^- [A^*] + k_- [A_\star] - (k^+ + k_+ ([A^*] + [A_\star^*])) [A] \\ [A^*]' &= k^+ [A] + k_- [A_\star^*] - (k^- + k_+ ([A^*] + [A_\star^*])) [A^*] \\ [A_\star]' &= k^- [A_\star^*] + k_+ [A] ([A^*] + [A_\star^*]) - (k^+ + k_-) [A_\star] \\ [A_\star^*]' &= k^+ [A_\star] + k_+ [A^*] ([A^*] + [A_\star^*]) - (k^- + k_-) [A_\star^*]. \end{aligned}$$

We notice that the correlation between the two sites can be abstracted away. Indeed, we notice that the following equations:

$$\begin{aligned} [A_\diamond]' &= k^- [A_\diamond^*] - k^+ [A_\diamond] \\ [A_\diamond^*]' &= k^+ [A_\diamond] - k^- [A_\diamond^*] \\ [A^\diamond]' &= k_- [A_\star^\diamond] - k_+ [A^\diamond] [A_\diamond^*] \\ [A_\star^\diamond]' &= k_+ [A^\diamond] [A_\diamond^*] - k_- [A_\star^\diamond], \end{aligned}$$

where  $[A_\diamond] := [A] + [A_\star]$ ,  $[A_\diamond^*] := [A^*] + [A_\star^*]$ ,  $[A^\diamond] := [A] + [A^*]$ , and  $[A_\star^\diamond] := [A_\star] + [A_\star^*]$ , are satisfied.

We now wonder whether the same reduction can be used in the case of the stochastic semantics. In the stochastic semantics, a chemical soup can be denoted by a 4-tuple  $\langle n_A, n_{A^*}, n_{A_\star}, n_{A_\star^*} \rangle$  of natural numbers, where  $n_X$  is the number of instance of  $X$  in the chemical soup, for any  $X \in \{A, A_\star, A^*, A_\star^*\}$ . The probability  $P_t(\langle n_A, n_{A^*}, n_{A_\star}, n_{A_\star^*} \rangle)$  that the system is in a given state  $\langle n_A, n_{A^*}, n_{A_\star}, n_{A_\star^*} \rangle$  at time  $t$  is given by the following master equation:

$$\begin{aligned} P_t(\langle n_A, n_{A^*}, n_{A_\star}, n_{A_\star^*} \rangle)' &= \\ & k^+(n_A + 1)P_t(\langle n_A + 1, n_{A^*} - 1, n_{A_\star}, n_{A_\star^*} \rangle) \\ & + k^+(n_{A_\star} + 1)P_t(\langle n_A, n_{A^*}, n_{A_\star} + 1, n_{A_\star^*} - 1 \rangle) \\ & + k^-(n_{A^*} + 1)P_t(\langle n_A - 1, n_{A^*} + 1, n_{A_\star}, n_{A_\star^*} \rangle) \\ & + k^-(n_{A_\star^*} + 1)P_t(\langle n_A, n_{A^*}, n_{A_\star} - 1, n_{A_\star^*} + 1 \rangle) \\ & + k_+(n_A + 1)(n_{A^*} + n_{A_\star^*})P_t(\langle n_A + 1, n_{A^*}, n_{A_\star} - 1, n_{A_\star^*} \rangle) \\ & + k_+(n_{A^*} + 1)(n_{A^*} + n_{A_\star^*} - 1)P_t(\langle n_A, n_{A^*} + 1, n_{A_\star}, n_{A_\star^*} - 1 \rangle) \\ & + k_-(n_{A_\star} + 1)P_t(\langle n_A - 1, n_{A^*}, n_{A_\star} + 1, n_{A_\star^*} \rangle) \\ & + k_-(n_{A_\star^*} + 1)P_t(\langle n_A, n_{A^*} - 1, n_{A_\star}, n_{A_\star^*} + 1 \rangle) \\ & - k^+(n_A + n_{A_\star})P_t(\langle n_A, n_{A^*}, n_{A_\star}, n_{A_\star^*} \rangle) \\ & - k^-(n_{A^*} + n_{A_\star^*})P_t(\langle n_A, n_{A^*}, n_{A_\star}, n_{A_\star^*} \rangle) \\ & - k_+(n_A(n_{A^*} + n_{A_\star^*}) + n_{A^*}(n_{A^*} + n_{A_\star^*} - 1))P_t(\langle n_A, n_{A^*}, n_{A_\star}, n_{A_\star^*} \rangle) \\ & - k_-(n_{A_\star} + n_{A_\star^*})P_t(\langle n_A, n_{A^*}, n_{A_\star}, n_{A_\star^*} \rangle). \end{aligned}$$

Given a state  $\sigma = \langle n_A, n_{A^*}, n_{A_\star}, n_{A_\star^*} \rangle$ , we denote by  $\beta(\sigma)$  the 4-tuple  $\langle n_{A_\diamond}, n_{A_\diamond^*}, n_{A^\diamond}, n_{A_\star^\diamond} \rangle$  (such a tuple is called an abstract state). The probability  $P_t^\sharp(\sigma^\sharp)$  that the system is in a state  $\sigma$  such that  $\beta(\sigma) = \sigma^\sharp$  at time  $t$ ,

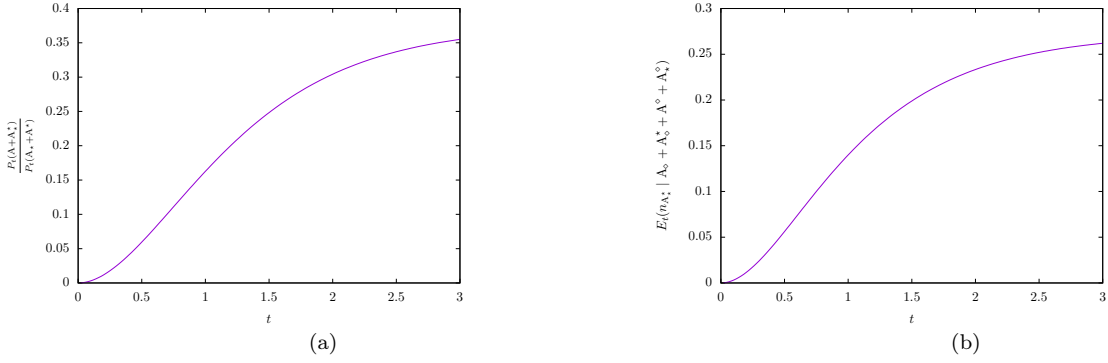


Figure 5.3: On the left, quotient between the probability of being in the state  $A + A^*$  and the probability of being in the state  $A_* + A^*$ . On the right, conditional expectation of the number of protein in the state  $A^*$  knowing that (i) there are two proteins, (ii) exactly one protein is phosphorylated at its the first site, and (iii) exactly one protein (potentially the same) is phosphorylated at its second site. All rates are set to 1. At time 0, the chemical soup is made of two proteins  $A$  fully unphosphorylated.

satisfies the following equation:

$$\begin{aligned}
P_t^\#(\langle n_{A_\diamond}, n_{A_*}, n_{A^\diamond}, n_{A^*} \rangle)' = & \\
& k_+(n_{A_\diamond} + 1)P_t^\#(\langle n_{A_\diamond}+1, n_{A_*}-1, n_{A^\diamond}, n_{A^*} \rangle) \\
& + k_-(n_{A_*} + 1)P_t^\#(\langle n_{A_\diamond}-1, n_{A_*}+1, n_{A^\diamond}, n_{A^*} \rangle) \\
& + k_+(n_{A^\diamond} + 1)n_{A_*}P_t^\#(\langle n_{A_\diamond}, n_{A_*}, n_{A^\diamond}+1, n_{A^*}-1 \rangle) \\
& + k_-(n_{A^*} + 1)P_t^\#(\langle n_{A_\diamond}, n_{A_*}, n_{A^\diamond}-1, n_{A^*}+1 \rangle) \\
& - (k^+n_{A_\diamond} + k^-n_{A_*} + k_+n_{A^\diamond}n_{A_*} + k_-n_{A^*})P_t^\#(\langle n_{A_\diamond}, n_{A_*}, n_{A^\diamond}, n_{A^*} \rangle) \\
& - k_+\tilde{E}_t(n_{A^*} \mid \langle n_{A_\diamond}, n_{A_*}+1, n_{A^\diamond}, n_{A^*}-1 \rangle) \\
& + k_+\tilde{E}_t(n_{A^*} \mid \langle n_{A_\diamond}, n_{A_*}, n_{A^\diamond}, n_{A^*} \rangle),
\end{aligned}$$

where for any expression  $X(\sigma)$  and any (abstract) state  $\sigma^\#$ , the expression  $\tilde{E}_t(X(\sigma) \mid \sigma^\#)$  denotes the product between the conditional expectation  $E_t(X(\sigma) \mid \sigma^\#)$  of the expression  $X(\sigma)$  knowing that  $\beta(\sigma) = \sigma^\#$  and the probability  $P_t^\#(\sigma^\#)$  of being in a state  $\sigma$  such that  $\beta(\sigma) = \sigma^\#$ .

In general, the conditional properties of the number of instances of proteins in the form  $A^*$  having fixed a given abstract state, is time-dependent. We show in Fig. 5.3(a) that the ratio between the probability of being in the state  $A + A_*$  and the probability of being in the state  $A_* + A^*$  is time-dependent. Moreover, we show in Fig. 5.3(b) that the conditional expectation of  $n_{A^*}$  knowing that we are in the (abstract) state  $A_\diamond + A_\diamond^* + A_\diamond^\diamond + A_\diamond^*$  is time-dependent, which forbids doing the same simplification as in the differential semantics.

We have seen through this example that, because a given instance of chemical species can only be used once as a reactant when applying a given chemical reaction, some corrective terms as  $+1$  or  $-1$  may appear in master equations. These corrective terms may forbid the reduction of stochastic semantics. Nevertheless, this is not an issue when reducing differential semantics, since these corrective terms vanish when we consider an infinite number of instances of proteins (within an infinite volume).

## 5.5 Conclusion

In this paper, we have illustrated through small examples why it is more difficult to reduce the dimension of the state space of stochastic semantics than the one of differential semantics. In the case of the differential semantics, it is possible to abstract away some correlations between the state of some fragments of chemical species, because these correlations are not observed by the (groups of) reactions. This is not so easy in the case of stochastic semantics, because a given reaction application may operate on several fragments simultaneously, in such a case the choice for the state of fragments on which semi-reactions are applied is driven by the correlation between the state of these two fragments (see Sect. 5.4.2). Moreover, stochastic semantics counts individuals which leads to some constant corrective terms (such as increment or decrement by 1) which also forbids exact reduction (see Sect. 5.4.3).

Model	early EGF	EGF/Insulin cross talk	SFB
Species	356	2899	$\sim 2.10^{19}$
ODE fragments	38	208	$\sim 2.10^5$
Stochastic fragments	356	618	$\sim 2.10^{19}$

Figure 5.4: Reduction factors for differential fragments (??) and stochastic fragments. We try these reduction methods on three models. The first one is the model of the early events of the EGF pathway (see Sect. ??); the second one, taken from (?, table 7), describes the cross-talk between another model of the early events of the EGF pathway and the insulin receptor; whereas the third one is a version of a pilot study on a larger section of the EGF pathway (??Schoeberl et collab. 2002 ; Brightman et Fell 2000).

In Fig. 5.4, we give the number of chemical species, the number of differential fragments, and the number of stochastic fragments for three bigger models. The reduction factor for the differential semantics is very interesting, whereas there is almost no reduction in the stochastic case. A careful look into the models would show that this is due to coupled semi-reactions. Moreover, the reduction that arises in the second model is due to a protein which has two fully independent parts (as in Sect. 5.4.1).

This emphasizes how interesting the stochastic semantics is: the stochastic semantics does not only describe a limit behavior, but also shows the variability of a system and how robust a system is to stochastic variations. The counterpart is that it is very difficult to handle with (as a formal object) and to simplify.



## Chapter 6

# Symétrie dans les graphes à sites

Les symétries apparaissent dans les modèles décrits par des règles de réécritures sous des formes diverses. Certaines sont indissociables de la sémantique. C'est le cas des automorphismes qui représentent des équivalences entre agents au sein de graphes à sites. D'autres peuvent être considérées pour décrire des équivalences entre sites d'interaction, des équivalences entre sortes de protéines ou encore des équivalences spatiales qui permettent de voir la structure de certains complexes biochimiques modulo des isométries.

Dans (Feret 2014), nous avons introduit un cadre de travail algébrique unifiant pour décrire, inférer, et utiliser les des groupes de transformations de graphes à sites dans le langage Kappa. L'idée principale est de considérer des groupes de transformations qui agissent sur les graphes à sites avec une opération supplémentaire permettant de restreindre une transformation sur les sous-graphes d'un graphe. Il est alors possible de définir quand un ensemble de règle est symétrique par rapport à ces groupes de transformation et d'en déduire des propriétés au niveau du comportement global du modèle, que ce soit pour sa sémantique stochastique ou différentielle. Pour simplifier la présentation, nous nous concentrons ici aux symétries qui correspondent à des permutations de sites, qui sont décrites dans (Camporesi et collab. 2010).

**Travaux voisins.** Plusieurs formalismes permettent de faire apparaître chaque site plusieurs fois dans l'interface des agents. C'est le cas du langage BNGL (Blinov et collab. 2004b). Dans le langage ReactC (John et collab. 2011b), des sites indistinguables peuvent être encodés à l'aide d'hyperliens. Ceci procure un moyen syntaxique de décrire des sites symétriques. En contre-partie, le calcul des ensembles des occurrences des motifs de réécriture devient prohibitif (les graphes ne sont plus rigides), ce qui pose de grands problèmes de complexité pour échantillonner les trajectoires de la sémantique stochastique.

En Kappa, la rigidité des graphes est un principe fondamental, et donc, l'utilisation de plusieurs instances d'un même site dans l'interface d'un agent est interdite. Les équivalences entre sites d'interaction peuvent toutefois être décrites à un plus haut-niveau de spécification (Harmer 2009) puis traduites automatiquement dans le noyau pur de Kappa. Cependant, il faut être conscient que cette traduction peut générer un très grand nombre de règles, là où une seule règle aurait été suffisante en utilisant des sites à occurrences multiples.

Dans ce chapitre, nous proposons une approche pour détecter automatiquement des équivalences entre sites d'interaction qui n'ont pas besoin d'être spécifiées à la main. Puis, nous utilisons ces équivalences pour réduire l'espace d'état ou le nombre d'espèces biochimiques des modèles considérés. Ceci permet de réduire la dimension des systèmes différentiels sous-jacents et facilite le calcul de propriétés sur la distribution des traces de la sémantique stochastique. Ce n'est en revanche pas crucial pour l'échantillonnage des traces de la sémantique stochastique, puisque la simulation travaille directement au niveau des graphes à sites (Danos et collab. 2007c).

Enfin, notre cadre de travail n'est pas limité à la prise en compte des équivalences entre sites d'interaction. Il peut être utilisé pour considérer des chaînes d'agents indépendamment de leur orientation ou des sous-groupes de permutations de sites comme c'est courant en chemoinformatique.

## Remerciements.

Ce chapitre reprends donc les principaux résultats qui avaient été établis avec Ferdinanda Camporesi (Camporesi et collab. 2010) à propos de la détection et de la prise en compte d'équivalences entre sites d'interaction dans un noyau sans effets de bord du langage Kappa, pour réduire la combinatoire de ses modèles. L'extension à des

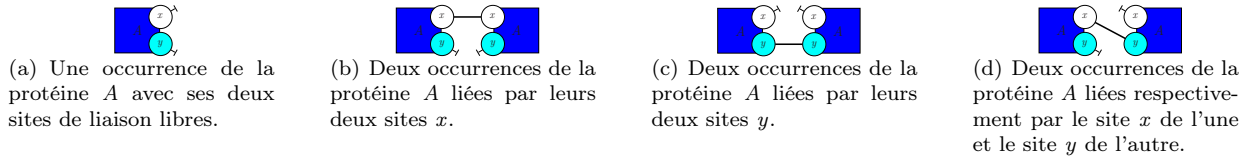


Figure 6.1: The four kinds of molecular species in the initial model.

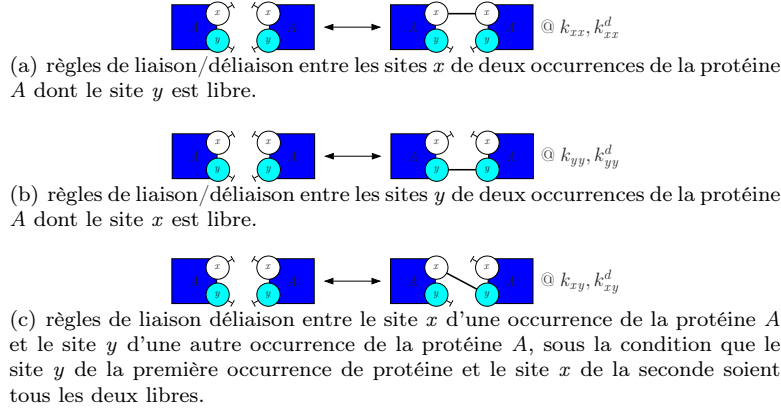


Figure 6.2: Les six règles de réécriture du modèle initial.

groupes de symétries arbitraires et au langage complet qui est introduite dans (Feret 2014) n'est évoquée qu'en conclusion de ce chapitre.

## 6.1 Introduction

Nous illustrons l'usage des symétries dans Kappa à travers un exemple jouet.

### 6.1.1 Un modèle avec des sites équivalents

Pour simplifier la présentation, nous considérons une forme très simple de symétrie, celle où des sites d'une protéine ont exactement les mêmes capacités d'interaction.

Soit une sorte de protéine,  $A$ , chaque occurrence de la protéine  $A$  disposant d'exactly deux sites de liaison que nous noterons  $x$  et  $y$ . Les sites de deux occurrences différentes de la protéine  $A$  peuvent se lier arbitrairement de manière réversible. Il y a donc trois types de liaison,  $x-x$ ,  $x-y$ , et  $y-y$ , selon les sites qui sont impliqués dans la liaison. De plus, il est supposé que chaque occurrence de protéine ne peut être qu'une fois à la fois.

Sous ces hypothèses, il existe exactement quatre sortes de complexe biochimique<sup>1</sup>. Celles-ci sont toutes les quatre dessinées en Fig. 6.1. Une occurrence de complexe biochimique est ainsi formé soit d'une occurrence de protéine avec ses deux sites libres (voir en Fig. 6.1(a)), soit deux occurrences de la protéine  $A$  sont liées par leurs deux sites  $x$  (voir en Fig. 6.1(b)), par leurs deux sites  $y$  (voir en Fig. 6.1(c)) ou par le site  $x$  de l'une et le site  $y$  de l'autre (voir en Fig. 6.1(d)).

Les règles de liaison et de déliaison entre les occurrences de la protéine  $A$  sont décrites en Fig. 6.2. La règle bidirectionnelle dessinée en Fig. 6.2(a) stipule que deux occurrences de la protéine  $A$  dont tous les sites sont libres peuvent lier leurs sites  $x$  respectifs avec la constante de réaction  $k_{xx}$  et qu'un tel lien peut se défaire avec la constante de réaction  $k_{xx}^d$ . La règle bidirectionnelle donnée en Fig. 6.2(b) précise que deux occurrences de la protéine  $A$  dont tous les sites sont libres peuvent lier leurs sites  $y$  respectifs avec la constante de réaction  $k_{yy}$  et qu'un tel lien peut se briser avec la constante de réaction  $k_{yy}^d$ . Enfin, la règle bidirectionnelle dessinée en Fig. 6.2(c) spécifie que deux occurrences de la protéine  $A$  dont tous les sites sont libres, peuvent se lier

<sup>1</sup>voir page ?? pour la définition d'un complexe biochimique.



$$\begin{aligned}
\frac{dP_t(q_{6,0,0,0})}{dt} &= \frac{2k_{xx}^d}{2} P_t(q_{4,1,0,0}) + \frac{2k_{yy}^d}{2} P_t(q_{4,0,1,0}) + k_{xy}^d \cdot P_t(q_{4,0,0,1}) - \frac{30(k_{xx} + k_{yy} + k_{xy})}{2} P_t(q_{6,0,0,0}) \\
\frac{dP_t(q_{4,1,0,0})}{dt} &= \frac{30k_{xx}}{2} P_t(q_{6,0,0,0}) + \frac{4k_{xx}^d}{2} P_t(q_{2,2,0,0}) + \frac{2k_{yy}^d}{2} P_t(q_{2,1,1,0}) + k_{xy}^d \cdot P_t(q_{2,1,0,1}) - \frac{12(k_{xx} + k_{yy} + k_{xy}) + 2k_{xx}^d}{2} P_t(q_{4,1,0,0}) \\
\frac{dP_t(q_{4,0,1,0})}{dt} &= \frac{30k_{yy}}{2} P_t(q_{6,0,0,0}) + \frac{2k_{xx}^d}{2} P_t(q_{2,1,1,0}) + \frac{4k_{yy}^d}{2} P_t(q_{2,0,2,0}) + k_{xy}^d \cdot P_t(q_{2,0,1,1}) - \frac{12(k_{xx} + k_{yy} + k_{xy}) + 2k_{yy}^d}{2} P_t(q_{4,0,1,0}) \\
\frac{dP_t(q_{4,0,0,1})}{dt} &= \frac{30k_{xy}}{2} P_t(q_{6,0,0,0}) + \frac{2k_{xx}^d}{2} P_t(q_{2,1,0,1}) + \frac{2k_{yy}^d}{2} P_t(q_{2,0,1,1}) + 2k_{xy}^d \cdot P_t(q_{2,0,0,2}) - \left( \frac{12(k_{xx} + k_{yy} + k_{xy})}{2} + k_{xy}^d \right) P_t(q_{4,0,0,1}) \\
\frac{dP_t(q_{2,2,0,0})}{dt} &= \frac{12k_{xx}}{2} P_t(q_{4,1,0,0}) + \frac{6k_{xx}^d}{2} P_t(q_{0,3,0,0}) + \frac{2k_{yy}^d}{2} P_t(q_{0,2,1,0}) + k_{xy}^d \cdot P_t(q_{0,2,0,1}) - \frac{2k_{xx} + 2k_{yy} + 2k_{xy} + 4k_{xx}^d}{2} P_t(q_{2,2,0,0}) \\
\frac{dP_t(q_{2,1,1,0})}{dt} &= \frac{12k_{xy}}{2} P_t(q_{4,1,0,0}) + \frac{4k_{xx}^d}{2} P_t(q_{0,2,1,0}) + \frac{4k_{yy}^d}{2} P_t(q_{0,1,1,1}) + k_{xy}^d \cdot P_t(q_{0,1,1,1}) - \frac{2k_{xx} + 2k_{yy} + 2k_{xy} + 2k_{xx}^d + 2k_{yy}^d}{2} P_t(q_{2,1,1,0}) \\
\frac{dP_t(q_{2,1,0,1})}{dt} &= \frac{12k_{xx}}{2} P_t(q_{4,1,0,0}) + \frac{12k_{xy}}{2} P_t(q_{4,1,0,0}) + \frac{4k_{xx}^d}{2} P_t(q_{0,2,0,1}) + \frac{2k_{yy}^d}{2} P_t(q_{0,1,1,1}) + 2k_{xy}^d \cdot P_t(q_{0,1,0,2}) - \left( \frac{2k_{xx} + 2k_{yy} + 2k_{xy} + 2k_{xx}^d}{2} + k_{xy}^d \right) P_t(q_{2,1,0,1}) \\
\frac{dP_t(q_{2,0,2,0})}{dt} &= \frac{12k_{yy}}{2} P_t(q_{4,0,1,0}) + \frac{2k_{xx}^d}{2} P_t(q_{0,1,2,0}) + \frac{6k_{yy}^d}{2} P_t(q_{0,0,3,0}) + k_{xy}^d \cdot P_t(q_{0,0,2,1}) - \frac{2k_{xx} + 2k_{yy} + 2k_{xy} + 4k_{yy}^d}{2} P_t(q_{2,0,2,0}) \\
\frac{dP_t(q_{2,0,1,1})}{dt} &= \frac{12k_{xy}}{2} P_t(q_{4,0,1,0}) + \frac{12k_{xy}}{2} P_t(q_{4,0,1,0}) + \frac{4k_{xx}^d}{2} P_t(q_{0,1,1,1}) + \frac{4k_{yy}^d}{2} P_t(q_{0,0,2,1}) + 2k_{xy}^d \cdot P_t(q_{0,0,1,2}) - \left( \frac{2k_{xx} + 2k_{yy} + 2k_{xy} + 2k_{xx}^d}{2} + k_{xy}^d \right) P_t(q_{2,0,1,1}) \\
\frac{dP_t(q_{2,0,0,2})}{dt} &= \frac{12k_{xy}}{2} P_t(q_{4,0,0,1}) + \frac{2k_{xx}^d}{2} P_t(q_{0,1,0,2}) + \frac{2k_{yy}^d}{2} P_t(q_{0,0,1,2}) + 3k_{xy}^d \cdot P_t(q_{0,0,0,3}) - \left( \frac{2k_{xx} + 2k_{yy} + 2k_{xy}}{2} + 2 \cdot k_{xy}^d \right) P_t(q_{2,0,0,2}) \\
\frac{dP_t(q_{0,3,0,0})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,2,0,0}) - \frac{6k_{xx}^d}{2} P_t(q_{0,3,0,0}) \\
\frac{dP_t(q_{0,2,1,0})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,1,1,0}) + \frac{2k_{yy}}{2} P_t(q_{2,2,0,0}) - \frac{4k_{xx}^d + 2k_{yy}^d}{2} P_t(q_{0,2,1,0}) \\
\frac{dP_t(q_{0,2,0,1})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,1,0,1}) + \frac{2k_{xy}}{2} P_t(q_{2,2,0,0}) - \left( \frac{4k_{xx}^d}{2} + k_{xy}^d \right) P_t(q_{0,2,0,1}) \\
\frac{dP_t(q_{0,1,2,0})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,0,2,0}) + \frac{2k_{yy}}{2} P_t(q_{2,1,1,0}) - \frac{2k_{xx}^d + 4k_{yy}^d}{2} P_t(q_{0,1,2,0}) \\
\frac{dP_t(q_{0,1,1,1})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,0,1,1}) + \frac{2k_{yy}}{2} P_t(q_{2,1,0,1}) + \frac{2k_{xy}}{2} P_t(q_{2,1,1,0}) - \left( \frac{2k_{xx}^d + 2k_{yy}^d}{2} + k_{xy}^d \right) P_t(q_{0,1,1,1}) \\
\frac{dP_t(q_{0,1,0,2})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,0,0,2}) + \frac{2k_{xy}}{2} P_t(q_{2,1,0,1}) - \left( \frac{2k_{xx}^d}{2} + 2k_{xy}^d \right) P_t(q_{0,1,0,2}) \\
\frac{dP_t(q_{0,0,3,0})}{dt} &= \frac{2k_{yy}}{2} P_t(q_{2,0,2,0}) - \frac{6k_{yy}^d}{2} P_t(q_{0,0,3,0}) \\
\frac{dP_t(q_{0,0,2,1})}{dt} &= \frac{2k_{xy}}{2} P_t(q_{2,0,1,1}) + \frac{2k_{xy}}{2} P_t(q_{2,0,2,0}) - \left( \frac{4k_{yy}^d}{2} + k_{xy}^d \right) P_t(q_{0,0,2,1}) \\
\frac{dP_t(q_{0,0,1,2})}{dt} &= \frac{2k_{yy}}{2} P_t(q_{2,0,0,2}) + \frac{2k_{xy}}{2} P_t(q_{2,0,1,1}) - \left( \frac{2k_{yy}^d}{2} + 2k_{xy}^d \right) P_t(q_{0,0,1,2}) \\
\frac{dP_t(q_{0,0,0,3})}{dt} &= \frac{2k_{xy}}{2} P_t(q_{2,0,0,2}) - 3k_{xy}^d \cdot P_t(q_{0,0,0,3})
\end{aligned}$$

Figure 6.3: Équation maîtresse.

respectivement par le site  $x$  de l'une et le site  $y$  de l'autres avec la constante de réaction  $k_{xy}$  et qu'un tel lien peut se casser avec la constante de réaction  $k_{xy}^d$ .

### 6.1.2 sémantique

Ainsi, d'un point de vue qualitatif, les sites  $x$  et  $y$  des occurrences de la protéine  $A$  ont exactement les mêmes capacités d'interaction. Nous allons étudier de manière empirique ce que cela implique au niveau de la sémantique du modèle, à la fois pour la sémantique stochastique et pour la sémantique différentielle.

#### Équation maîtresse

Pour regarder le comportement de la sémantique stochastique du modèle, nous proposons de nous concentrer sur la solution de son équation maîtresse (McQuarrie 1967) qui décrit pour chaque état potentiel du système sous-jacent l'évolution temporelle de la probabilité que le système soit dans cet état. Comme il y a quatre sortes de complexes biochimiques, l'état discret du système peut être représenté par un vecteur de quatre entiers naturels, en notant  $i,j,k,l$  l'état dans lequel il y a exactement  $i$  occurrence(s) de la protéine  $A$  sans sites liés (voir en Fig. 6.1(a)),  $j$  paires d'occurrences de la protéine  $A$  liées par leurs sites  $x$  respectifs (voir en Fig. 6.1(b)),  $k$  paires d'occurrences de la protéine  $A$  liées par leurs sites  $y$  respectifs (voir en Fig. 6.1(c)), et  $l$  paires d'occurrences de la protéine  $A$  liées l'une par son site  $x$  et l'autre par son site  $y$  (voir en Fig. 6.1(d)). L'équation maîtresse nécessite une variable par état possible du système. Afin de faire tenir les équations sur une page, la système débutera avec la probabilité 1 dans un état avec uniquement six occurrences de la protéine  $A$  avec tous leurs sites libres. Aussi, seuls seront considérés les états  $i,j,k,l$  pour lesquels  $i + 2 \cdot (j + k + l) = 6$ . Aussi, l'équation maîtresse admet vingt variables, en l'occurrence,  $q_{6,0,0,0}$ ,  $q_{4,1,0,0}$ ,  $q_{4,0,1,0}$ ,  $q_{4,0,0,1}$ ,  $q_{2,2,0,0}$ ,  $q_{2,1,1,0}$ ,  $q_{2,1,0,1}$ ,  $q_{2,0,2,0}$ ,  $q_{2,0,1,1}$ ,  $q_{2,0,0,2}$ ,  $q_{0,3,0,0}$ ,  $q_{0,2,1,0}$ ,  $q_{0,2,0,1}$ ,  $q_{0,1,2,0}$ ,  $q_{0,1,1,1}$ ,  $q_{0,1,0,2}$ ,  $q_{0,0,3,0}$ ,  $q_{0,0,2,1}$ ,  $q_{0,0,1,2}$ , et  $q_{0,0,0,3}$ .

L'équation maîtresse de notre modèle jouet est donnée en Fig. 6.11, alors que l'évolution temporelle de la distribution de probabilité de chaque état accessible à partir de l'état initial est donné en Fig. 6.4 pour les paramètres cinétiques  $k_{xx} = k_{xy} = 1$ ,  $k_{yy} = k_{xx}^d = k_{yy}^d = 2$ , et  $k_{xy}^d = 4$ .

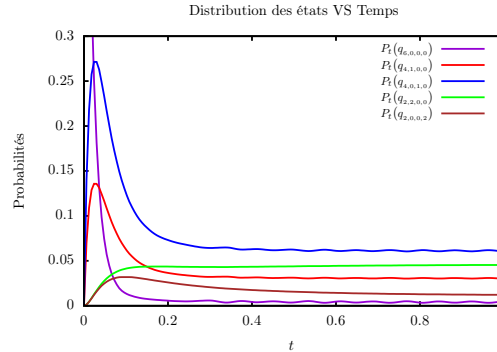


Figure 6.4: Évolution de la distribution de certains états en fonction du temps, avec les paramètres cinétiques  $k_{xx} = k_{xy} = 1$ ,  $k_{yy} = k_{xx}^d = k_{yy}^d = 2$ , et  $k_{xy}^d = 4$ , et la distribution d'états initiale  $P_0(q_{6,0,0,0}) = 1$ .

#### Sémantique différentielle

La sémantique différentielle du modèle est définie par le systèmes d'équations différentielles donné en Fig. ???. Ces équations décrivent l'évolution de la concentration des différents complexes biochimique sous les hypothèses de la loi d'action de masse. Les trajectoires de ce système en prenant pour état initial, l'état où les monomères de la protéine  $A$  ont pour concentration 6 et les dimers ont pour concentration 0 sont décrites en Fig. 6.6.

### 6.1.3 Symétries et propriétés comportementales

Sans considérer les aspects quantitatifs, les sites  $x$  et  $y$  des occurrences de la protéine  $A$  sont équivalents. Dès qu'une règle permet de lier le site  $x$  d'une occurrence de la protéine  $A$  à un site, une autre règle permet de lier le

$$\begin{cases} \frac{d[A]}{dt} = 2k_{xy}^d[A.x - y.A] + 4\frac{k_{yy}^d}{2}2[A.y - y.A] + 4\frac{k_{xx}^d}{2}2[A.x - x.A] - (2\frac{k_{yy}}{2} + 2\frac{k_{xx}}{2} + 2\frac{k_{xy}}{2})[A] \\ \frac{d[A.x - x.A]}{dt} = \frac{k_{xx}}{2}[A]^2 - 2\frac{k_{xx}^d}{2}2[A.x - x.A] \\ \frac{d[A.y - y.A]}{dt} = \frac{k_{xy}}{2}[A]^2 - 2\frac{k_{yy}^d}{2}2[A.y - y.A] \\ \frac{d[A.x - y.A]}{dt} = \frac{k_{yy}}{2}[A]^2 - k_{xy}^d A.x - y.A \end{cases}$$

Figure 6.5: Système d'équations différentielles. La solution de ce système décrit l'évolution des concentrations des différents complexes biochimique en fonction du temps, sous les hypothèses de la loi d'Action de Masse.

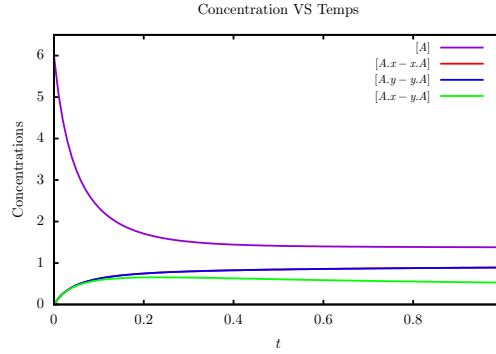


Figure 6.6: Évolution de la concentration des différents complexes biochimiques en fonction du temps, avec les paramètres cinétiques  $k_{xx} = k_{xy} = 1$ ,  $k_{yy} = k_{xx}^d = k_{yy}^d = 2$ , et  $k_{xy}^d = 4$ , et l'état initial  $[A] = 6$  et  $[A.x - x.A] = [A.x - y.A] = [A.y - y.A] = 0$ .

site  $y$  de cette occurrence de protéine au même site. De même, quand une règle permet de casser un lien entre le site  $x$  d'une occurrence de protéine et un autre site, une autre règle permet de casser ce lien, si ce lien avait été porté par le site  $y$  de cette occurrence de protéines.

Il s'en suit la question suivante : quelles conséquences les équivalences entre sites peuvent-elles impliquer en terme de comportement du système sous-jacent et pour quelles conditions supplémentaires sur les paramètres cinétiques du modèle et son état (ou sa distribution d'états) initial(e) ?

### Modèle simplifié

Il semble naturel de vouloir oublier la différence entre des sites équivalents. Dans notre modèle jouet, cela revient à ignorer la différence entre les sites  $x$  et  $y$  dans les occurrences de la protéine  $A$ . À ce niveau d'abstraction, les occurrences de la protéine  $A$  ont chacune deux sites, désormais supposés indistingables.

**Complexes biochimiques.** Comme dessiné en Fig. 6.7, il ne reste alors que deux sortes de complexes biochimiques, les monomères qui sont formés d'une occurrence de la protéine  $A$  avec ses deux sites de liaison libres et les dimères qui sont constitués de deux occurrences de la protéine  $A$  liées exactement par une liaison entre un site de l'une et un site de l'autre.

**États des systèmes stochastiques et différentiels sous-jacent** Les états du système stochastique sous-jacent sont donc décrits par le nombre d'occurrences de monomère (qui sont formées d'une occurrence de la protéine  $A$ ) et le nombre d'occurrences de dimère (qui sont formées de deux occurrences de la protéine  $A$  liées entre-elles). Un tel état sera noté  $Q_{i,j}$  avec  $i$  le nombre d'occurrences de monomère et  $j$  le nombre d'occurrences de dimère.

Quant à eux, les états du système différentiel sous-jacent sont décrits par deux concentrations. La concentration en monomère est notée  $[A_1]$  alors que celle en dimère est notée  $[A_2]$ .

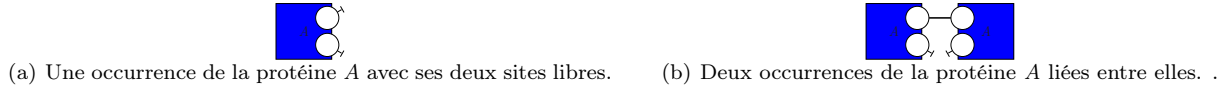


Figure 6.7: Les deux sortes de complexes biochimiques dans le modèle simplifié. Le nom des sites a été retiré car ils sont maintenant supposés indistinguables.

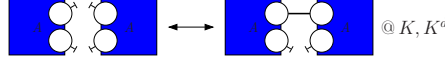


Figure 6.8: La règle bidirectionnelle du modèle simplifié. Deux occurrences de la protéine  $A$  peuvent se lier à une constante de réaction  $K$  (peu importe à quels sites) quand tous leurs sites sont libres. De plus, ce lien peut se rompre avec une constante de réaction  $K^d$ .

**règles simplifiées** Par ailleurs, les règles se simplifient de la même manière. Puisqu'il n'y a plus de distinctions entre les sites de liaison de chaque occurrence de la protéine  $A$ , les trois règles de liaison peuvent se résumer en une seule. Étant donné deux occurrences de la protéine  $A$  dont tous les sites sont libres, il y a trois règles pour créer un lien entre ces deux occurrences et chacune peut s'appliquer de deux manières différentes (selon le choix de quelle occurrence est utilisée pour la première occurrence de la protéine  $A$  du membre gauche de la règle) possibilité de créer un liaison pour une constante de réaction corrigée totale  $\frac{2k_{xx}+2k_{yy}+2k_{xy}}{2}$ . Du coup, pour simuler ces règles et rester fidèle à la cinétique, la constante de réaction corrigée de la nouvelle règle devra être égale elle aussi à  $\frac{2k_{xx}+2k_{yy}+2k_{xy}}{2}$ . Par ailleurs, étant données deux occurrences de la protéine  $A$ , une seule règle s'applique pour briser ce lien. Si le lien est entre deux sites  $x$ , cette règle peut s'appliquer de deux manières pour une constante de réaction cumulée corrigée de  $\frac{2k_{xx}^d}{2}$ . Si le lien est entre deux sites  $y$ , cette règle peut s'appliquer de deux manières pour une constante de réaction cumulée corrigée de  $\frac{2k_{yy}^d}{2}$ . Enfin si le lien est entre un site  $x$  et un site  $y$ , la règle peut s'appliquer de deux manières pour une constante de réaction cumulée corrigée de  $\frac{2k_{xy}^d}{2}$ . Du coup, la simplification n'est possible que si les trois constantes  $k_{xx}^d$ ,  $k_{yy}^d$  et  $k_{xy}^d$  sont égales. Dans ce cas, la constante de la règle de déliaison dans le modèle simplifié sera fixé à  $k_{xx}^d$ . Les deux règles du modèle simplifié sont données en Fig. 6.8.

**systèmes dynamiques sous-jacent** L'équation maîtresse du modèle simplifié est donnée en Fig. 6.9. La solution de ce système d'équations définit donc l'évolution de la distribution de probabilités du nombre de dimers au cours du temps dans la sémantique stochastique. Seuls les états accessibles à partir de six monomères ont été considérés, c'est à dire les états  $Q_{6,0}$ ,  $Q_{4,1}$ ,  $Q_{2,2}$  et  $Q_{0,3}$ .

$$\begin{cases} \frac{dP_t(Q_{6,0})}{dt} = \frac{2K^d}{2}P_t(Q_{4,1}) - \frac{30K}{2}P_t(Q_{6,0}) \\ \frac{dP_t(Q_{4,1})}{dt} = \frac{30K}{2}P_t(Q_{6,0}) + \frac{4K^d}{2}P_t(Q_{2,2}) - \left(\frac{12K}{2} + K^d\right)P_t(Q_{4,1}) \\ \frac{dP_t(Q_{2,2})}{dt} = \frac{12K}{2}P_t(Q_{4,1}) + \frac{6K^d}{2}P_t(Q_{0,3}) - \left(\frac{2K}{2} + \frac{4K^d}{2}\right)P_t(Q_{2,2}) \\ \frac{dP_t(Q_{0,3})}{dt} = \frac{2K}{2}P_t(Q_{2,2}) - \frac{6K^d}{2}P_t(Q_{0,3}) \end{cases}$$

Figure 6.9: Chemical Master Equation of the reduced model.

La sémantique différentielle est quant à elle donnée en Fig. 6.10. Elle donne l'évolution de la concentration en monomère,  $[A_1]$ , et en dimère,  $[A_2]$ , au cours du temps, sous les hypothèses de la loi d'action de masse.

**Comparaison entre les deux modèles** Les composants de modèle simplifié ont été définis de manière intentionnelle sous la forme de graphe à sites. Quelques écarts ont été pris par rapport à la syntaxe de Kappa. En effet, celle-ci ne permet pas de répéter des sites dans l'interface d'une protéine. Cette description intentionnelle permet des raisonnements intuitifs sur le comportement des composants du modèle simplifié. Toutefois, une caractérisation extensionnelle de ces composants est requise pour relier formellement les composants du

$$\begin{cases} \frac{d[A_1]}{dt} = 4\frac{K^d}{2}[A_2] - 2\frac{K}{2}[A_1]^2 \\ \frac{d[A_2]}{dt} = \frac{K}{2}[A_1]^2 - 2\frac{K^d}{2}[A_2] \end{cases}$$

Figure 6.10: système différentiel pour le modèle simplifié.

modèle simplifié au composants du modèle initial et ainsi justifier rigoureusement ces raisonnements. Il suffit pour cela d'interpréter les complexes biochimiques et les états du modèle simplifié respectivement comme des classes d'équivalences de complexes biochimiques et d'états du modèle initial. Les complexes biochimiques sont regroupés en deux classes, les monomères et les dimères. Ainsi, dans le cadre différentiel, les variables du modèle simplifié correspondent à la somme des concentrations des espèce biochimiques dans chaque classe d'équivalences de complexes biochimiques, comme indiqué en Fig. 6.12. De la même manière, les états du système stochastique initial peuvent être regroupés selon le nombre d'occurrences de monomère et de dimère. Du coup, à partir d'un état initial formé de 6 occurrences de la protéine  $A$  avec ses deux sites sont libres, il est possible d'attendre des états dans exactement 4 classes d'équivalence, selon qu'il y ait 0, 1, 2 ou 3 dimères. Ceci peut être étendu aux distributions d'états qui apparaissent dans l'équation maîtresse : la probabilité d'avoir exactement  $i$  monomères et  $j$  dimères est alors vue comme la somme des probabilité des états du modèle initial ayant exactement  $i$  monomères et  $j$  dimères. Cette relation est exprimée en Fig. 6.13 pour les états accessibles à partir de 6 occurrences de la protéine  $A$  avec les deux sites de liaison libres.

Il existe donc deux manières de calculer le comportement du modèle simplifié : soit en résolvant directement les équations du modèle simplifié, soit en résolvant le modèle initial et en calculant le comportement du modèle simplifié à l'aide de la caractérisation extensionnelle de ses composants. Cependant, ceci n'est possible que si les trois paramètres  $k_{xx}^d$ ,  $k_{yy}^d$  et  $k_{xy}^d$  sont égaux. Dans ce cas, le paramètre  $K$  doit être pris comme la somme des paramètres  $k_{xx}$ ,  $k_{yy}$  et  $k_{xy}$ , alors que le paramètre  $K^d$  doit être égal au paramètre  $k_{xx}^d$ . De plus, la distribution du système initial, dans le cadre stochastique, ou l'état initial, dans le cadre différentiel, doivent être choisi pour satisfaire la caractérisation extensionnelle des composants du modèle simplifié.

En Fig. 6.14 est montrée la comparaison empirique entre ces deux méthodes pour plusieurs jeux de paramètres, que ce soit dans la cadre stochastique ou différentiel. La colonne de gauche est consacrée aux solutions des équations maîtresses, alors que la colonne de droite porte sur celles des équations aux concentrations. Quatre jeux de paramètres sont considérés. même si les paramètres n'ont pas la même signification dans le cadre stochastique et dans celui différentiel, la comparaison est faite avec les mêmes valeurs numériques. Dans chaque graphique, les lignes continues sont obtenues en considérant le modèle initial et en projetant les résultats à l'aide de la caractérisation extensionnelle des composants du modèle simplifié alors que les courbes obtenues directement avec le modèle simplifié sont dessinées avec des tirets.

Le but de ces simulations numériques est de tester l'importance de trois contraintes. L'élaboration du modèle simplifié a mis en lumière qu'il était suffisant que que les constantes de dissociation  $k_{xx}^d$ ,  $k_{xy}^d$  et  $k_{yy}^d$  soient tous trois égaux. C'est la première contrainte qui sera considérée. Par ailleurs, il semble légitime de se demander si des contraintes sur la distribution initiale (ou sur l'état initial dans le cadre différentiel) et sur les constantes d'association ont aussi des conséquences notables. Aussi l'importance que les sites  $x$  et  $y$  soient, ou non, indistinguables dans la distribution initiale (ou l'état initial) sera également testé empiriquement et, par ailleurs, l'impact de la contrainte qui impose aux deux constantes  $k_{xx}$  et  $k_{yy}$  d'être égales et à la constante  $k_{xy}$  d'être le double de leur valeur commune sera aussi testé (le choix des rapports sera expliquer plus tard).

1. Dans le premier jeu de paramètres, les trois contraintes sont satisfaites. Les constantes  $k_{xx}$  et  $k_{yy}$  sont fixées à 1, alors que les constantes  $k_{xy}$ ,  $k_{xx}^d$ ,  $k_{yy}^d$  et  $k_{xy}^d$  sont fixées à 2. Le système stochastique sous-jacent débute de l'état formé de 6 occurrences de la protéine  $A$  sans lien, et le système différentiel de l'état où les monomères sont présent en concentration 6, et les dimères absents. Les courbes montrent que la distribution des états du modèle simplifié correspondent exactement (voir en Fig. 6.14(a)) : les mêmes valeurs sont obtenues en calculant l'évolution de la distribution des états dans le modèle initial et en regroupant les états selon le nombre d'occurrences de dimère (courbes continues) ou en calculant la distribution des états dans le modèle simplifié directement (tirets). Le même phénomène se retrouve dans les systèmes différentiels sous-jacents. Les courbes en Fig. 6.14(b) montrent que les mêmes concentrations sont obtenues en regroupant la concentration des dimères dans le modèle initial ou en calculant directement cette concentration dans le modèle simplifié.
2. Dans le second jeu de paramètres, la contrainte sur les constantes de dissociation n'est plus satisfaite. Les

$$\begin{aligned}
\frac{dP_t(q_{6,0,0,0})}{dt} &= \frac{2k^d}{2} P_t(q_{4,1,0,0}) + \frac{2k^d}{2} P_t(q_{4,0,1,0}) + k_{xy}^d \cdot P_t(q_{4,0,0,1}) - \frac{30(k_{xx} + k_{xy} + k_{yy})}{2} P_t(q_{6,0,0,0}) \\
\frac{dP_t(q_{4,1,0,0})}{dt} &= \frac{30k_{xx}}{2} P_t(q_{6,0,0,0}) + \frac{4k^d}{2} P_t(q_{2,2,0,0}) + \frac{2k^d}{2} P_t(q_{2,1,1,0}) + k_{xy}^d \cdot P_t(q_{2,1,0,1}) - \frac{12(k_{xx} + k_{xy} + k_{yy}) + 2k_{xx}^d}{2} P_t(q_{4,1,0,0}) \\
\frac{dP_t(q_{4,0,1,0})}{dt} &= \frac{30k_{yy}}{2} P_t(q_{6,0,0,0}) + \frac{2k_{xx}^d}{2} P_t(q_{2,1,1,0}) + \frac{4k_{yy}^d}{2} P_t(q_{2,0,2,0}) + k_{xy}^d \cdot P_t(q_{2,0,1,1}) - \frac{12(k_{xx} + k_{xy} + k_{yy}) + 2k_{yy}^d}{2} P_t(q_{4,0,1,0}) \\
\frac{dP_t(q_{4,0,0,1})}{dt} &= \frac{30k_{xy}}{2} P_t(q_{6,0,0,0}) + \frac{2k_{xx}^d}{2} P_t(q_{2,1,0,1}) + \frac{2k_{yy}^d}{2} P_t(q_{2,0,1,1}) + 2k_{xy}^d \cdot P_t(q_{2,0,0,2}) - \left( \frac{12(k_{xx} + k_{xy} + k_{yy})}{2} + k_{xy}^d \right) P_t(q_{4,0,0,1}) \\
\frac{dP_t(q_{2,2,0,0})}{dt} &= \frac{12k_{xx}}{2} P_t(q_{4,1,0,0}) + \frac{6k_{xx}^d}{2} P_t(q_{0,3,0,0}) + \frac{2k_{yy}^d}{2} P_t(q_{0,2,1,0}) + k_{xy}^d \cdot P_t(q_{0,2,0,1}) - \frac{2k_{xx} + 2k_{xy} + 2k_{yy} + 4k_{xx}^d}{2} P_t(q_{2,2,0,0}) \\
\frac{dP_t(q_{2,1,1,0})}{dt} &= \frac{12k_{xx}}{2} P_t(q_{4,0,1,0}) + \frac{12k_{yy}}{2} P_t(q_{4,1,0,0}) + \frac{4k_{xx}^d}{2} P_t(q_{0,2,1,0}) + \frac{4k_{yy}^d}{2} P_t(q_{0,1,2,0}) + k_{xy}^d \cdot P_t(q_{0,1,1,1}) - \frac{2k_{xx} + 2k_{xy} + 2k_{yy} + 2k_{xx}^d + 2k_{yy}^d}{2} P_t(q_{2,1,1,0}) \\
\frac{dP_t(q_{2,1,0,1})}{dt} &= \frac{12k_{xx}}{2} P_t(q_{4,0,0,1}) + \frac{12k_{xy}}{2} P_t(q_{4,1,0,0}) + \frac{4k_{xx}^d}{2} P_t(q_{0,2,0,1}) + \frac{2k_{yy}^d}{2} P_t(q_{0,1,1,1}) + 2k_{xy}^d \cdot P_t(q_{0,1,0,2}) - \left( \frac{2k_{xx} + 2k_{xy} + 2k_{yy} + 2k_{xx}^d}{2} + k_{xy}^d \right) P_t(q_{2,1,0,1}) \\
\frac{dP_t(q_{2,0,2,0})}{dt} &= \frac{12k_{yy}}{2} P_t(q_{4,0,1,0}) + \frac{2k_{xx}^d}{2} P_t(q_{0,1,2,0}) + \frac{6k_{yy}^d}{2} P_t(q_{0,0,3,0}) + k_{xy}^d \cdot P_t(q_{0,0,2,1}) - \frac{2k_{xx} + 2k_{xy} + 2k_{yy} + 4k_{yy}^d}{2} P_t(q_{2,0,2,0}) \\
\frac{dP_t(q_{2,0,1,1})}{dt} &= \frac{12k_{yy}}{2} P_t(q_{4,0,0,1}) + \frac{12k_{xy}}{2} P_t(q_{4,1,0,0}) + \frac{4k_{xx}^d}{2} P_t(q_{0,1,1,1}) + \frac{4k_{yy}^d}{2} P_t(q_{0,0,2,1}) + 2k_{xy}^d \cdot P_t(q_{0,0,1,2}) - \left( \frac{2k_{xx} + 2k_{xy} + 2k_{yy} + 2k_{xx}^d}{2} + k_{xy}^d \right) P_t(q_{2,0,1,1}) \\
\frac{dP_t(q_{2,0,0,2})}{dt} &= \frac{12k_{xy}}{2} P_t(q_{4,0,0,1}) + \frac{2k_{xx}^d}{2} P_t(q_{0,1,0,2}) + \frac{2k_{yy}^d}{2} P_t(q_{0,1,0,2}) + 3k_{xy}^d \cdot P_t(q_{0,0,0,3}) - \left( \frac{2k_{xx} + 2k_{xy} + 2k_{yy} + 2k_{xx}^d}{2} + 2k_{xy}^d \right) P_t(q_{2,0,0,2}) \\
\frac{dP_t(q_{0,3,0,0})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,2,0,0}) - \frac{6k_{xx}^d}{2} P_t(q_{0,3,0,0}) \\
\frac{dP_t(q_{0,2,1,0})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,1,1,0}) + \frac{2k_{yy}}{2} P_t(q_{2,2,0,0}) - \frac{4k_{xx}^d + 2k_{yy}^d}{2} P_t(q_{0,2,1,0}) \\
\frac{dP_t(q_{0,2,0,1})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,1,0,1}) + \frac{2k_{xy}}{2} P_t(q_{2,2,0,0}) - \left( \frac{4k_{xx}^d}{2} + k_{xy}^d \right) P_t(q_{0,2,0,1}) \\
\frac{dP_t(q_{0,1,2,0})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,0,2,0}) + \frac{2k_{yy}}{2} P_t(q_{2,1,1,0}) - \frac{2k_{xx}^d + 4k_{yy}^d}{2} P_t(q_{0,1,2,0}) \\
\frac{dP_t(q_{0,1,1,1})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,0,1,1}) + \frac{2k_{yy}}{2} P_t(q_{2,1,0,1}) + \frac{2k_{xy}}{2} P_t(q_{2,1,1,0}) - \left( \frac{2k_{xx}^d + 2k_{yy}^d}{2} + k_{xy}^d \right) P_t(q_{0,1,1,1}) \\
\frac{dP_t(q_{0,1,0,2})}{dt} &= \frac{2k_{xx}}{2} P_t(q_{2,0,0,2}) + \frac{2k_{xy}}{2} P_t(q_{2,1,0,1}) - \left( \frac{2k_{xx}^d}{2} + 2k_{xy}^d \right) P_t(q_{0,1,0,2}) \\
\frac{dP_t(q_{0,0,3,0})}{dt} &= \frac{2k_{xy}}{2} P_t(q_{2,0,2,0}) - \frac{6k_{xy}^d}{2} P_t(q_{0,0,3,0}) \\
\frac{dP_t(q_{0,0,2,1})}{dt} &= \frac{2k_{xy}}{2} P_t(q_{2,0,1,1}) + \frac{2k_{xx}}{2} P_t(q_{2,0,2,0}) - \left( \frac{4k_{xy}^d}{2} + k_{xx}^d \right) P_t(q_{0,0,2,1}) \\
\frac{dP_t(q_{0,0,1,2})}{dt} &= \frac{2k_{xy}}{2} P_t(q_{2,0,0,2}) + \frac{2k_{xx}}{2} P_t(q_{2,0,1,1}) - \left( \frac{2k_{xy}^d}{2} + 2k_{xx}^d \right) P_t(q_{0,0,1,2}) \\
\frac{dP_t(q_{0,0,0,3})}{dt} &= \frac{2k_{xy}}{2} P_t(q_{2,0,0,2}) - 3k_{xy}^d \cdot P_t(q_{0,0,0,3})
\end{aligned}$$

Figure 6.11: Équation maîtresse.

$$\begin{cases} [A_1] = [A], \\ [A_2] = [A.x - x.A] + [A.y - y.A] + [A.x - y.A]. \end{cases}$$

Figure 6.12: Définition extensionnelle des variables du modèle simplifié pour le cadre différentiel.

$$\begin{cases} P_t(Q_{6,0}) = P_t(q_{6,0,0,0}) \\ P_t(Q_{4,1}) = P_t(q_{4,1,0,0}) + P_t(q_{4,0,1,0}) + P_t(q_{4,0,0,1}) \\ P_t(Q_{2,2}) = P_t(q_{2,2,0,0}) + P_t(q_{2,1,1,0}) + P_t(q_{2,1,0,1}) + P_t(q_{2,0,2,0}) + P_t(q_{2,0,1,1}) + P_t(q_{2,0,0,2}) \\ P(Q_{0,3}) = P_t(q_{0,3,0,0}) + P_t(q_{0,2,1,0}) + P_t(q_{0,2,0,1}) + P_t(q_{0,1,2,0}) + P_t(q_{0,1,1,1}) + P_t(q_{0,1,0,2}) \\ \quad + P_t(q_{0,0,3,0}) + P_t(q_{0,0,2,1}) + P_t(q_{0,0,0,3}) + P_t(q_{0,0,1,2}) + P_t(q_{0,1,0,2}). \end{cases}$$

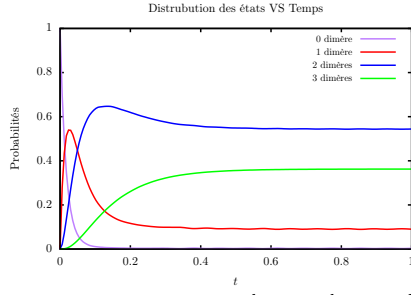
Figure 6.13: Définition extensionnelle des variables de l'équation maîtresse du modèle simplifié.

constantes d'association et la distribution des états initiaux (ou l'état initial dans le cadre différentiel) sont gardées telles quelles, ainsi que les constantes de dissociation  $k_{xx}^d$  et  $k_{yy}^d$ . Mais la constante de dissociation  $k_{xy}^d$  est maintenant fixée à 4. Dans le modèle simplifié, la constante de dissociation est prise comme étant égale à 3 pour tenir compte de ce changement. On peut alors constater à la fois dans le cadre stochastique et dans le cadre différentiel un écart entre le comportement du modèle initial et du modèle simplifié (voir en Fig. 6.14(c) et en Fig. 6.14(d)).

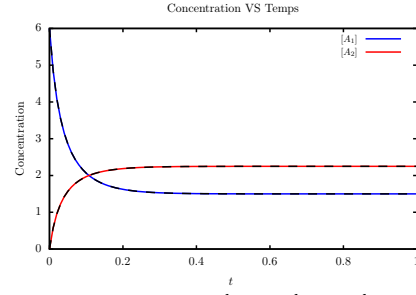
3. Le troisième jeu de paramètres permet de tester l'importance éventuelle de l'état initial. Par rapport au premier jeu de paramètre, seule la distribution des états initiaux (ou l'état initial dans le cadre différentiel) est modifiée. Le système stochastique débute dans un état avec deux occurrences de la protéine  $A$  sans lien, et deux occurrences du dimère chacun débute de deux occurrences de la protéine  $A$  lié par leurs sites  $x$  respectifs. Quant au système différentiel, il démarre d'un état où les monomères et les dimères  $x - x$  sont en concentration 2, alors que les deux autres formes de dimère sont en concentration 0. L'évolution des distributions d'états des deux systèmes stochastiques sous-jacents est dessinée en Fig. 6.14(e), alors que l'évolution des concentrations en monomère et en dimère dans les deux systèmes différentiels sous-jacents est donnée en Fig. 6.14(f). à la fois, le cadre stochastique et dans le cadre différentiel, les courbes  $\text{co}\ddot{A}^-$  ncident entre le modèle initial et le modèle simplifié, ce qui nous conforte dans l'idée que la correction du modèle simplifié ne dépend pas de la distribution initiale (ou de l'état initial dans le cadre stochastique).
4. Enfin, le quatrième jeu de paramètres nous conforte dans l'idée que les constantes d'association n'ont pas d'importance pour prouver la correction du modèle simplifié. Les paramètres sont les mêmes que dans le premier jeu, sauf la constante  $k_{yy}$  qui est fixée à 2 et la constante  $k_{xy}$  qui est fixée à 4. De plus, la constante d'association dans le modèle simplifié,  $K$ , reste la somme des constantes d'association dans le modèle initial, c'est à dire 7. Là encore, le comportement du modèle initial et celui du modèle simplifié coïncident, que ce soit pour la sémantique stochastique ou la sémantique différentielle (voir en Fig. 6.14(g) et en Fig. 6.14(h)).

Ainsi, si l'égalité entre les constantes de dissociation semble cruciale pour pouvoir simplifier le modèle de manière exacte, le choix de la distribution initiale (ou de l'état initial) et des constantes d'association semblent ne pas importer. Pour les constantes d'association, ceci s'explique par la simplicité du cas d'étude, puisque que les règles de liaison ne lient que des monomères et que ces monomères sont invariants par échange de leurs sites  $x$  et  $y$ . De manière général, les conditions sur les constantes cinétique portent sur des règles qui effectuent des actions similaires sur des motifs équivalents à permutation de sites près. La correction du modèle simplifiée peut alors s'expliquer par une bisimulation en-avant (Buchholz 2008) sur l'espace des états du système stochastique sous-jacent ou de son analogue différentiel (Cardelli et collab. 2015).

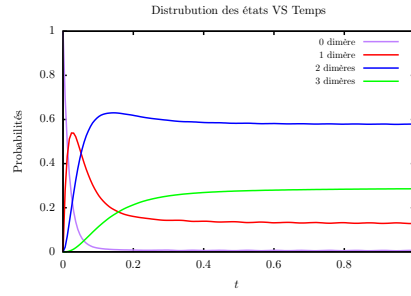
**Invariants quantitatifs** Dans le paragraphe précédent, le fait que certains sites puissent partager exactement les mêmes capacités d'interaction a été exploité pour simplifier un modèle jouet de manière exacte. Les composés du modèle simplifié ont été reliés formellement à ceux du modèle initial et les mêmes résultats ont été obtenus, en exécutant le modèle initial avant d'utiliser ces relations formelles pour en déduire l'évolution des composants du modèle simplifié ou en exécutant directement le modèle simplifié.



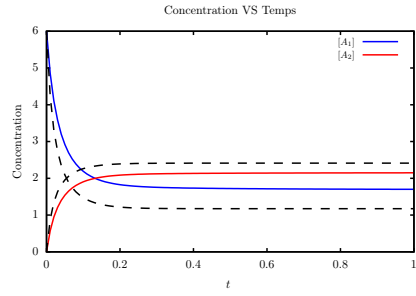
(a)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$  et  $P_0(q_{6,0,0,0}) = 1$  ;  $K = 4$ ,  $K^d = 2$ , et  $P_0(Q_{6,0}) = 1$ .



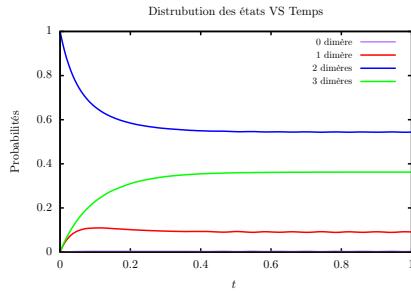
(b)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$ , initialement  $[A] = 6$  et  $[A.x-x.A] = [A.x-y.A] = [A.y-y.A] = 0$  ;  $K = 4$ ,  $K^d = 2$ , initialement  $[A_1] = 6$  et  $[A_2] = 0$ .



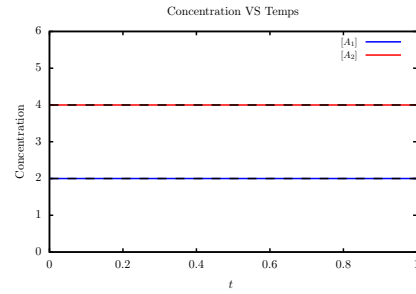
(c)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = 2$ ,  $k_{xy}^d = 4$  et  $P_0(q_{6,0,0,0}) = 1$  ;  $K = 4$ ,  $K^d = 3$  et  $P_0(Q_{6,0}) = 1$ .



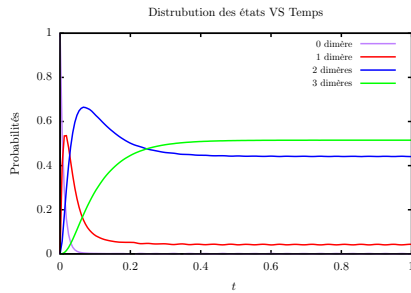
(d)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = 2$ ,  $k_{xy}^d = 4$ , initialement  $[A] = 6$  sans dimère ;  $K = 4$ ,  $K^d = 3$ , initialement  $[A_1] = 6$  et  $[A_2] = 0$ .



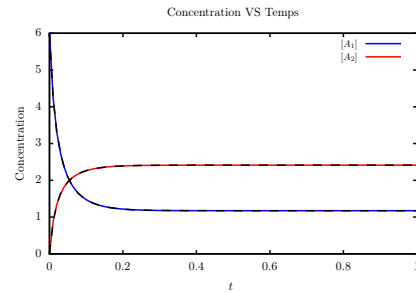
(e)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$ ,  $P_0(q_{2,2,0,0}) = 1$  ;  $K = 4$ ,  $K^d = 2$ ,  $P_0(Q_{2,2}) = 1$ .



(f)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$ , initialement  $[A] = [A.x-x.A] = 2$  et  $[A.y-y.A] = [A.x-y.A] = 0$  ;  $K = 4$ ,  $K^d = 2$ , initialement  $[A_1] = [A_2] = 2$ .



(g)  $k_{xx} = 1$ ,  $k_{yy} = 2$ ,  $k_{xy} = 4$ ,  $k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$  et  $P_0(q_{6,0,0,0}) = 1$  ;  $K = 7$ ,  $K^d = 2$  et  $P_0(Q_{6,0}) = 1$ .



(h)  $k_{xx} = 1$ ,  $k_{yy} = 2$ ,  $k_{xy} = 4$ ,  $k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$  initialement  $[A] = 6$  sans dimère ;  $K = 7$ ,  $K^d = 2$ , initialement  $[A_1] = 6$  et  $[A_2] = 0$ .

Figure 6.14: Comparaison entre le comportement du modèle initial (traits continus) et celui du modèle simplifié (tirets). à gauche, évolution de la distribution de probabilités du nombre de dimères pour différents paramètres cinétiques et différentes distributions d'états initiaux dans le système stochastique sous-jacent. à droite, évolution de la concentration en dimère pour différents paramètres cinétiques et différentes concentrations initiales.



Une autre classe de propriétés intéressante est celle des invariants quantitatifs. Il semble naturel de supposer que lorsque deux sites sont équivalents dans toutes les occurrences d'une protéine, alors, dans le cadre stochastique, les états obtenus en permutant ces deux sites dans une ou plusieurs occurrences de cette protéine dans un état donné sont équiprobables. De même, dans le cadre différentiel, la concentration des espèces biochimiques obtenues en permutant ces deux sites dans une ou plusieurs occurrences de cette protéine dans un complexe biochimique donné sont égales.

En fait, ces quantités ne sont pas égales en général. Des rapports de proportionnalité apparaissent naturellement. Prenons l'exemple d'un jeu de "pile" ou "face". Si deux tirages à "pile" ou "face" sont réalisés. Deux faces "piles" sont obtenues avec une probabilité un quart, deux faces "faces" avec une probabilité un quart, et un face "pile" et une face "face" avec une probabilité un demi alors qu'il est possible d'intervertir dans un des tirages ou dans les deux tirages la face "pile" et la "face" sans changer la nature du jeu. Le rapport de proportionnalité entre les probabilités d'être dans une configuration et celui d'être dans une autre est en réalité égal au rapport entre le nombre de permutations de faces de pièces qui laisse ses deux configurations inchangées. Ainsi, la configuration une face "pile" et une face "face" n'est pas modifiée lorsque le rôle des faces "pile" et "face" est échangé simultanément dans les deux tirages (c'est le seul changement non trivial qui laisse cette configuration inchangée), alors qu'un tirage double est changé pour tout changement non trivial, d'où un rapport de 2 sur 1 (en tenant compte de la permutation triviale).

Il semble légitime de se demander quelles conditions sont nécessaires pour retrouver de tels rapports de proportionnalité. Dans le cas d'étude, ces conditions doivent assurer, dans le cadre stochastique, que les probabilités d'être dans deux états obtenus en remplaçant une occurrence d'un type de dimère par une occurrence d'un autre type restent dans le même rapport de proportionnalité au cours du temps. Dans le cadre différentiel, ces conditions doivent assurer que les concentrations entre les différentes sortes de dimères restent dans le même rapport de proportionnalité aux cours du temps.

Pour simplifier le raisonnement, et quitte à renforcer les conditions nécessaires trouvées, supposons que la valeur des rapports de proportionnalité éventuels soit dictée par les rapports entre les constantes d'association. Pour que retrouver ces rapports de proportionnalités, il faut qu'ils soient également vérifiés dans la distribution initiale (dans le cadre stochastique) ou dans l'état initial (dans le cadre différentiel). Comme ces rapports de proportions ne sont pas a priori connus, le plus simple est de considérer, dans le cadre stochastique, une distribution initiale avec un état avec uniquement des occurrences de monomères avec probabilité 1, et, dans le cadre différentiel, un état initial avec uniquement des monomères. Par ailleurs, si le système est dans un état transitoire, vu la simplicité du modèle en question, le système va converger vers une distribution ou un état stationnaire dans lequel la quantité de chaque sorte de dimère sera proportionnelle au rapport entre les constantes d'association et de dissociation correspondantes. Du coup, pour que les rapports de proportions soient maintenus, il est nécessaire que les constantes de dissociations soient égales.

En Fig. 6.15 est vérifié de manière empirique si de tels rapports de proportionnalité se manifeste pour plusieurs jeux de paramètres, que ce soit dans la cadre stochastique (colonne de gauche) ou différentiel (colonne de droite), en faisant varier la valeur des constantes de dissociations, la distribution initiale (ou l'état initial), et les constantes d'association. Dans le cadre différentiel, l'évolution de trois rapports est considérée : le rapport entre l'état composé quatre occurrences de monomère et d'une occurrence de dimère formé par une liaison entre deux sites  $x$  et celui comprenant quatre occurrences de monomère et une occurrence de dimère formé par une liaison entre deux sites  $y$  ; le rapport entre l'état composé de quatre occurrences de monomère et d'une occurrence de dimère formé par une liaison entre deux sites  $x$  et celui comportant quatre occurrences de monomère et une occurrence de dimère formé par une liaison entre le site  $x$  d'une occurrence de protéine et le site  $y$  de l'autre occurrence de protéine ; et le rapport entre l'état composé de deux occurrences de monomère et de deux occurrences de dimère formé par une liaison entre le site  $x$  d'une occurrence de protéine et le site  $y$  de l'autre occurrence de protéine, et celui formé de deux occurrences de monomère, d'une occurrence de dimère formé par une liaison entre deux sites  $x$  et d'une occurrence de dimère formé par une liaison entre deux sites  $y$  ;

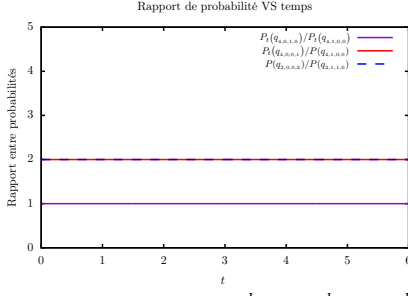
1. Dans le premier jeu de paramètres, les constantes d'association sont choisies pour que les liaisons se forment de manière équitable entre les sites  $x$  et  $y$  des occurrences de la protéine  $A$ . Les constantes de dissociation prennent toutes les trois la même valeur afin de préserver les rapports de proportionnalité quels qu'ils soient. Enfin, aucun dimère n'est présent ni dans la distribution initiale des états du système stochastique, ni dans l'état initial du système différentiel, de sorte que la distribution initiale du système stochastique, et l'état initial du système différentiel satisfont n'importe quels rapports de proportionnalité à propos de l'abondance des différentes sortes de dimère. En Fig. 6.15(a), les courbes montrent de manière

empirique que la probabilité d'être dans l'état avec quatre occurrences de la protéine  $A$  sous forme de monomère et deux occurrences de la protéine  $A$  liées par leurs sites  $x$  est toujours la même que celle d'être dans l'état avec quatre occurrences de la protéine  $A$  sous forme de monomère et deux occurrences de la protéine  $A$  liées par leurs sites  $y$ . Par ailleurs, cette probabilité est toujours la moitié de celle d'être dans l'état avec quatre occurrences de la protéine  $A$  sous forme de monomère et deux occurrences de la protéine  $A$  liées par le site  $x$  de l'un et le site  $y$  de l'autre. Ces rapports de proportionnalité correspondent à ceux déjà observés dans le jeu de "pile" ou "face". Enfin, la probabilité d'être dans un état avec deux occurrences de la protéine  $A$  sous forme de monomère et deux occurrences de dimères asymétriques est toujours deux fois plus grande que celle d'être dans un état avec deux occurrences de la protéine  $A$  sous forme de monomère et d'une occurrence de chaque forme de dimère symétrique. Ce rapport s'explique que la formation d'un dimère asymétrique est deux fois plus probable que celle d'un type donné de dimère symétrique (d'où un facteur 4). Cependant, il faut diviser ce facteur par 2, car deux dimères symétriques différents peuvent être obtenus en prenant le premier formé par une liaison entre deux sites  $x$  et le second entre deux sites  $y$ , ou l'inverse, soit deux fois plus de possibilités.

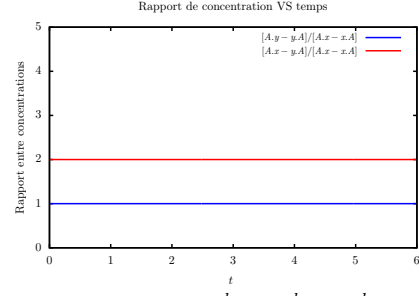
En ce qui concerne le comportement de la solution du système différentiel sous-jacent, les courbes en Fig. 6.15(b) montrent que les concentrations des deux sortes de dimère symétrique restent toujours égales, alors que la concentration en dimère asymétrique est toujours égale au double de cette valeur commune.

2. Dans le second jeu de paramètres, la contrainte sur les constantes de dissociation est relâchée. Les constantes de dissociations sont ainsi fixées de manière arbitraire à  $k_{xx}^d = 2$ ,  $k_{yy}^d = 2$  et  $k_{xy}^d = 4$ . Alors que les autres paramètres restent les mêmes que dans le premier jeu de paramètres. Les courbes dessinées en Fig. 6.15(c) montrent que la probabilité d'être dans l'état avec quatre occurrences de la protéine  $A$  sous forme de monomère et deux occurrences de la protéine  $A$  liées par leurs sites  $x$  est toujours la même que celle d'être dans l'état avec quatre occurrences de la protéine  $A$  sous forme de monomère et deux occurrences de la protéine  $A$  liées par leurs sites  $y$ . Par contre, les autres probabilités étudiées ne sont pas proportionnelles. Initialement, la probabilité d'être dans un état avec une occurrence de dimère asymétrique et quatre occurrences de la protéine  $A$  sous la forme de monomère est deux fois plus grande que celle d'être dans un état avec une occurrence de dimère asymétrique et quatre occurrences de la protéine  $A$  sous la forme de monomère, mais ce rapport tend vers un avec le temps. D'autre part, la probabilité d'être dans un état avec deux occurrences de dimère asymétrique et deux occurrences de la protéine  $A$  sous la forme de monomère est deux fois plus grande au début que la probabilité d'être dans un état avec une occurrence de chacun des types de dimère symétrique et deux occurrences de la protéine  $A$  sous la forme de monomère, mais ce rapport tend vers un demi avec le temps. Dans le système différentiel, la concentration des deux types de dimère symétrique est toujours la même, alors que la concentration en dimère asymétrique est initialement deux fois plus grande, pour finalement converger vers cette même valeur. Les rapports initiaux sont dictés par les constantes d'association, ce sont les mêmes que pour le premier jeu de paramètres. En revanche, les rapports à la limite sont dictés par le rapport entre les constantes respectives d'association et de dissociation (et la répétition des occurrences du dimère asymétrique pour le facteur un demi additionnel).
3. Dans le troisième jeu de paramètres, c'est la distribution d'état initiale (dans le cadre stochastique) et l'état initial (dans le cadre différentiel) qui sont fixés arbitrairement. Aucun rapport de proportionnalité ne se dessine, que ce soit dans le cadre stochastique (voir en Fig. 6.15(e)) ou dans le cadre différentiel (voir en Fig. 6.15(f)). Initialement, les rapports de probabilités et de concentrations sont imposés par les abondances au début de l'exécution des deux systèmes. à la limite, ces rapports sont fixés par le rapport entre les constantes respectives d'association et de dissociation, ce qui explique qu'ils sont les mêmes que pour le premier jeu de paramètres.
4. Enfin, le quatrième jeu de paramètres est obtenu en fixant de manière arbitraire les constantes d'association. Ceci a pour effet de déplacer les rapports de probabilités et les rapports de concentrations. Par contre, comme c'était attendu, ces rapports sont constants au cours du temps, que ce soit pour la sémantique stochastique ou la sémantique différentielle (voir en Fig. 6.14(g) et en Fig. 6.14(h)).

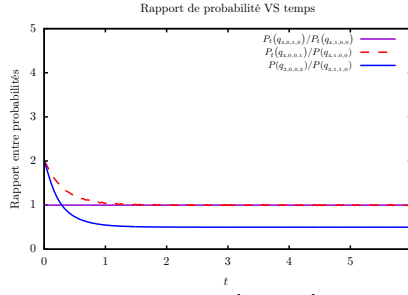
L'invariance de ces rapports de proportionnalité, pour certains jeux de paramètres est révélateur de la présence d'une relation de groupage faible. Celle-ci permet de quotienter l'exécution d'une chaîne de Markov pour certaines distributions d'états initiaux tout en restant Markovien (Buchholz 1994). De même, cela permet de réduire le système différentiel sous-jacent pour certains états initiaux.



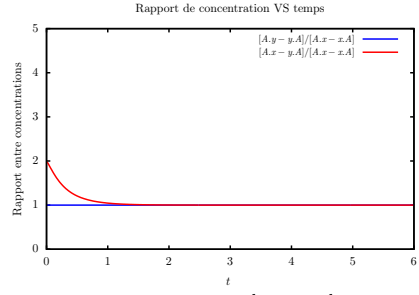
(a)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$  et  $P_0(q_{6,0,0,0}) = 1$ .



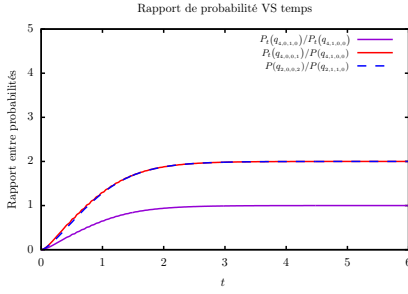
(b)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$ , initialement  $[A] = 6$  et  $[A.x - x.A] = [A.x - y.A] = [A.y - y.A] = 0$ .



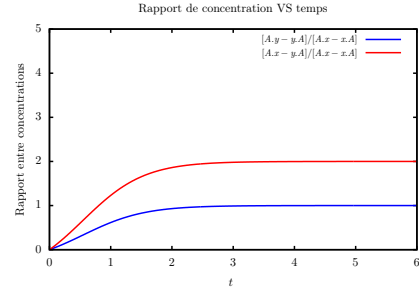
(c)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = 2$ ,  $k_{xy}^d = 4$  et  $P_0(q_{6,0,0,0}) = 1$ .



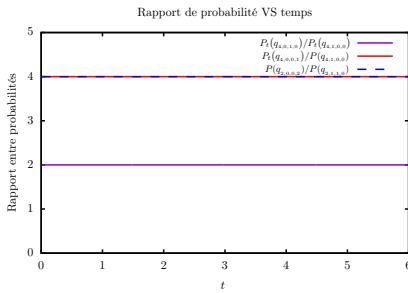
(d)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = 2$ ,  $k_{xy}^d = 4$ , initialement  $[A] = 6$  sans dimère.



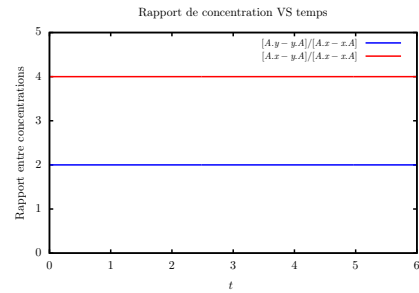
(e)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$ ,  $P_0(q_{2,2,0,0}) = 1$ .



(f)  $k_{xx} = k_{yy} = 1$ ,  $k_{xy} = k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$ , initialement  $[A] = [A.x - x.A] = 2$  et  $[A.y - y.A] = [A.x - y.A] = 0$ .



(g)  $k_{xx} = 1$ ,  $k_{yy} = 2$ ,  $k_{xy} = 4$ ,  $k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$  et  $P_0(q_{6,0,0,0}) = 1$ .



(h)  $k_{xx} = 1$ ,  $k_{yy} = 2$ ,  $k_{xy} = 4$ ,  $k_{xx}^d = k_{yy}^d = k_{xy}^d = 2$  initialement  $[A] = 6$  sans dimère.

Figure 6.15: Relations de proportionnalité dans le modèle initial. à gauche, évolution de rapports entre la probabilité d'être dans certains états pour différents paramètres cinétiques et différentes distributions d'états initiaux dans le système stochastique sous-jacent. à droite, évolution de rapports entre la concentration de différents complexes biochimiques pour différents paramètres cinétiques et différentes concentrations initiales.

Dans le cas du premier jeu de paramètres, le quotient opère même à plus bas-niveau, puisqu'il s'exprime au niveau réactionnel: à chaque réaction, peut être associée une réaction symétrique obtenue en échangeant les instances des sites  $x$  et  $y$  dans certaines occurrences de la protéine  $A$ , ce qui permet de montrer l'existence d'une bisimulation arrière (Buchholz 2008) sur système de transition stochastique sous-jacent ou de réduire le système différentiel sous-jacent.

### 6.1.4 Symetries and model reduction

To conclude with this motivating example, we have seen that symetries can arise in molecular species, in the states of the systems, in distributions of states and in the action of the rules and that the symetries have an impact in our capability to coarse-grain the dynamic of our system. In particular, when the action of the rules are symetric, we can safely quotient the states of the systems, which makes us think of a forward bisimulation. Moreover, in the initial distribution is symetric as well, then the reduced system carries a statistic invariant (namely, the quotient between probability of two symetric states is invariant and is equal to the inverse of the quotient between their number of automorphisms, which suggest a backward bisimulation.

Let us see how these results can be generalised to all the models written in the language *Kappa*.

## 6.2 Group actions over rule-based models

Now we specialise the notions of groups and actions of group to the case of site graphs. Our goal is to define groups of transformations between site graphs, such that the action of each transformation can be lifted to embeddings, rules, and rule refinements. Then, we will define the models that are symetric as the ones in which the set of rules (and their kinetics rates) are invariant by these transformations and show that, in such a case, two states which are equivalent modulo the group of transformations behave similarly by the means of bisimulations (?Buchholz 2008).

### 6.2.1 Group actions over site graphs

We want to provide a definition of transformations between site graphs which enables to lift the action of these transformations over the basic elements of the semantics, namely fully specified site graphs, weak embeddings, embeddings, rules, and refinements. To achieve this goal, we require that, for any weak embedding  $h$  between two site graphs  $E$  and  $F$ , each transformation which can be applied with the image  $F$  of the weak embedding  $h$  can be restricted into a transformation which can be applied to the domain  $E$  of the weak embedding  $h$ , hence providing a weak embedding between the image (by this transformation) of the domain  $E$  and the image (by this transformation) of the image  $F$  of the embedding  $h$ .

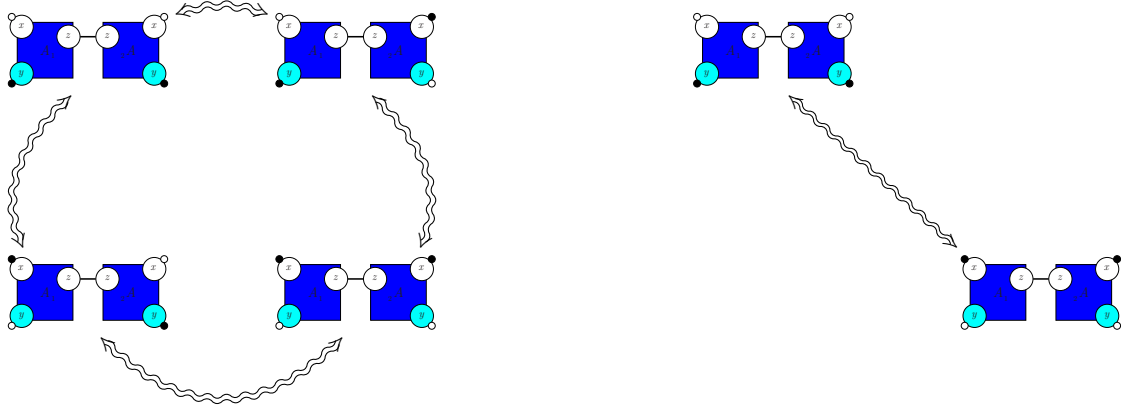
**Definition 6.2.1 (set of transformations)** *A set of transformations  $\mathbb{G}$  over site graphs is defined as a quadruple  $((\mathbb{G}_E, \circ_E), \cdot, ', '')$ , made of a family of finite groups  $(\mathbb{G}_E, \circ_E)$  indexed by site graphs, and three functions  $\cdot$ ,  $'$ , and  $''$ , where:*

- the function  $\cdot$  maps each pair  $(\sigma_E, E)$  such that  $E$  is a site graph and  $\sigma_E$  is an element of  $\mathbb{G}_E$  to a site graph, that is written  $\sigma_E.E$ ;
- the function  $'$  maps each pair  $(f, \sigma_F)$  such that  $f$  is a weak embedding between two site graphs  $E$  and  $F$ , and  $\sigma_F$  is an element of  $\mathbb{G}_F$  to an element in  $\mathbb{G}_E$ , that is written  $f.'\sigma_F$ ;
- the function  $''$  maps each pair  $(\sigma_F, f)$  such that  $f$  is a weak embedding between two site graphs  $E$  and  $F$ , and  $\sigma_F$  is an element of  $\mathbb{G}_F$  to a weak embedding  $\sigma_F.''f$  between the site graph  $(f.'\sigma_F).E$  and the site graph

$$\begin{array}{ccc}
 E & \xrightarrow{f} & F \\
 \sigma_F.F & \begin{array}{c} \Downarrow f.'\sigma_F \\ \Downarrow \sigma_F \end{array} & \\
 (f.'\sigma_F).E & \xrightarrow{\sigma_F.''f} & \sigma_F.F
 \end{array}$$

such that, for any site graph  $E$  and any element  $\sigma_E \in \mathbb{G}_E$ , the following properties are satisfied:

1. the groups  $(\mathbb{G}_E, \circ_E)$  and  $(\mathbb{G}_{\sigma_E.E}, \circ_{\sigma_E.E})$  are the same;



(a) Orbit of the site graph that is denoted by the pattern  $A_1(x_u, y_p, z^{A_2 @ z})$ ,  $A_2(x_u, y_p, z^{A_1 @ z})$  with the group of heterogeneous permutations of site  $\mathbb{G}^{\text{AGENT}}$  that consist, for each agent of type A, in permuting, or not, the internal state of two sites  $x$  and  $y$  of this agent.

(b) Orbit of the site graph that is denoted by the pattern  $A_1(x_u, y_p, z^{A_2 @ z})$ ,  $A_2(x_u, y_p, z^{A_1 @ z})$  with the group of homogeneous permutations of sites  $\mathbb{G}^{\text{SPECIES}}$  that consist in, for each connected component, either keeping the connected component as it is, or permuting the internal state of two sites  $x$  and  $y$  in all the agents of this connected component which have the type A.

Figure 6.16: Examples of transformations between site graphs.

2. the function which maps each element  $\sigma_E$  of the group  $\mathbb{G}_E$  and any site graph  $E'$  in the set  $\{\sigma_E.E \mid \sigma_E \in \mathbb{G}_E\}$  to the site graph  $\sigma_E.E'$  is a group action.

The result of the application of a transformation to a site graph (resp. to a weak embedding) is also called the image of the site graph (resp. weak embedding) by this transformation.

**Example 6.2.1 (heterogeneous permutations of sites)** As an example, we consider the transformations that consist in swapping (or not) the internal state of two sites  $x$  and  $y$  in agents of mixtures. We consider only one type of agents A, with three sites  $x, y, z$ . We assume that sites  $x$  and  $y$  can bear the internal state  $u$ , or the internal state  $p$ , and that sites  $z$  can be bound to the site  $z$  of another agent.

Thus, the signature of our model is the following one:

- $= \{A\}$ ;
- $= \{x, y, z\}$ ;
- $= \{u, p\}$ ;
- $(A) = \{x, y\}$ ;
- $(A) = \{z\}$ .

For any site graph  $E$ , we introduce  $\mathbb{G}_E^{\text{AGENT}}$  as the set  $\wp([E])$  of the subsets of the set of the agents of  $E$ . Intuitively, an element  $X \in \mathbb{G}_E^{\text{AGENT}}$  denotes the agents in which we are going to swap the state of the site  $x$  and the site  $y$ .

The set  $\mathbb{G}_E^{\text{AGENT}}$  is a group for the symmetric difference  $\otimes$  which is defined as follows:

$$\otimes : \begin{cases} [E] \times [E] & \rightarrow [E] \\ (X, Y) & \rightarrow (X \setminus Y) \cup (Y \setminus X). \end{cases}$$

The empty set is the identity element and each element is its own inverse.

The image of a site graph  $E$  by a transformation  $X \in \mathbb{G}_E^{\text{AGENT}}$  is obtained by swapping the internal state of sites  $x$  and  $y$  in the agents that belongs to the set  $X$ , that is to say that, if we write  $E =$  and  $\sigma$  the permutation over the set mapping the site identifier  $x$  to the site identifier  $y$ , the site identifier  $y$  to the site identifier  $x$ , and the site identifier  $z$  to itself, the site graph  $X.E$  is defined as the site graph  $(, ', ', ')$  where the set  $'$  is defined as the join between the set  $\{(n, i) \in \mid n \notin X\}$  and the set  $\{(n, \sigma(i)) \mid n \in X, (n, i) \in\}$ , and for any site  $(n, i) \in'$ ,

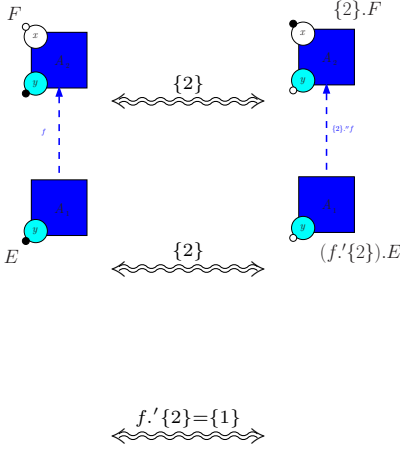


Figure 6.17: An example of two symmetric embeddings. We consider the embedding  $f$  between the site graph  $E \triangleq A_1(y_p)$  and the site graph  $F \triangleq A_2(x_u, y_p)$ . The permutation of sites  $\{2\} \in \mathbb{G}_F$  swaps the state of the sites  $x$  and  $y$  in the unique agent of the site graph  $F$ . The permutation of sites  $\{2\}$  can also be restricted to the domain  $E$  of the embedding  $f$ , which yields the permutation of sites  $f.'\{2\} \in \mathbb{G}_E$ , which is indeed equal to the permutation of sites  $\{1\}$  and that consists in replacing the site  $y$  into a site  $x$  (with the same internal state). The image  $\{2\}.''f$  of the embedding  $f$  by the permutation of sites  $\{2\}$  is then the embedding mapping the unique agent of the image  $(f.'\{2\}).E$  of site graph  $E$  by the permutation of sites  $f.'\{2\}$  into the unique agent of the image  $\{2\}.F$  of the site graph  $F$  by the permutation of sites  $\{2\}$ .

$'(n, i)$  is defined as  $'(n, i)$  is defined as  $(n, \sigma(i))$  whenever the agent identifier  $n$  belongs to the set  $X$  and,  $'(n, i)$  is defined as  $(n, i)$  otherwise.

The orbit of the site graph that is denoted by the expression  $A_1(x_u, y_p, z^{A_2 @ z})$ ,  $A_2(x_u, y_p, z^{A_1 @ z})$  is given in Fig. 6.16(a).

Given an embedding  $f$  between two site graphs  $E$  and  $F$ , and  $X \in \mathbb{G}_F^{\text{AGENT}}$  a permutation of the sites of the site graph  $F$ , the restriction of the symmetry  $X$  to the domain  $E$  of the embedding  $f$  is defined as  $\{a \in [E] \mid \tilde{f}(a) \in X\}$ . Intuitively, this means that the transformation  $f.'X$  swaps the state of the sites  $x$  and  $y$  in a given agent of  $E$  if, and only if, the transformation  $X$  swaps the states of the sites  $x$  and  $y$  in the image by the embedding  $f$  of this agent.

Lastly, given an embedding  $f$  between two site graphs  $E$  and  $F$  by a permutation  $X \in \mathbb{G}_E^{\text{AGENT}}$  of the sites of the site graph  $E$ , we notice that, on the first hand, the site graphs  $E$  and  $(f.'X).E$  have the same set of agents, and that, on the second hand, the site graphs  $F$  and  $X.F$  have the same set of agents. Moreover, we notice that the function  $\tilde{f}$  also induces an embedding between the site graphs  $(f.'X).E$  and  $X.F$ . We define the embedding  $X.f$  as this embedding.

In Fig. 6.17, we give an example of an embedding  $f$  between two site graphs  $E$  and  $F$ , and we give the image of this embedding by a permutation of sites.

In Exa. 6.2.1, the permutation of the sites  $x$  and  $y$  is done independently for each agent. We call such permutations of sites heterogeneous. Now we would like to define a new set of transformation in which the same permutation has to be applied to both agents in each dimer. Checking the assumptions of Def. 6.2.1 can be cumbersome. Since we introduce a more direct way to introduce sets of transformations over site graphs as subsets of existing ones. More precisely, one can define a new set of transformations over site graphs as a subset of an existing one, providing that the former one preserves the structure of the latter one, that is to say, that this new set must be close upon product of transformations, and upon the restriction of a transformation to the domain of weak embeddings. This is formalised in the following definition.

**Definition 6.2.2 (subset of transformations over site graphs)** Let us consider  $\mathbb{G} := ((\mathbb{G}_E, \circ_E), \cdot, \cdot', \cdot'')$  a set of transformations over site graphs. We call a subset of transformations of  $\mathbb{G}$ , a family  $(\tilde{\mathbb{G}}_E)$  of sets indexed by site graphs which satisfies the following properties:

1. for any site graph  $G$ , the set  $\tilde{\mathbb{G}}_G$  is a non empty subset of the set  $\mathbb{G}_G$ ;
2. for any site graph  $G$  and any transformation  $\sigma \in \tilde{\mathbb{G}}_G$  over the site graph  $G$ , the set  $\tilde{\mathbb{G}}_G$  and the set  $\tilde{\mathbb{G}}_{\sigma.G}$  are the equal;

3. for any site graph  $G$  and any two transformations  $\sigma, \sigma' \in \tilde{\mathbb{G}}_G$  of the set  $\tilde{\mathbb{G}}_G$ , the transformation  $\sigma \circ_G \sigma'$  belongs to the set  $\tilde{\mathbb{G}}_G$  as well;
4. for any weak embedding  $f$  between two site graphs  $E$  and  $F$  and any transformation  $\sigma \in \tilde{\mathbb{G}}_F$  over the site graph  $F$ , the restriction  $f.\sigma$  of the transformation  $\sigma$  to the domain of  $f$  also belongs to the set  $\tilde{\mathbb{G}}_E$ .

**Propriété 6.2.1** Let  $((\mathbb{G}_E, \circ_E), \cdot, \cdot', \cdot'')$  be a set of transformations over site graphs and  $(\mathbb{G}'_E)$  be a subset of transformations of the the set of transformations  $(\mathbb{G}_E)$ . Then the tuple  $((\tilde{\mathbb{G}}_E, \tilde{\circ}_E), \tilde{\cdot}, \tilde{\cdot}', \tilde{\cdot}'')$  where :

1. for any site graph  $E$ , the function  $\tilde{\circ}_E$  maps any couple of transformations  $\sigma, \sigma' \in \tilde{\mathbb{G}}_E$  that can be applied to the site graph  $E$ , to the transformation  $\sigma \circ_E \sigma'$ ,
2. the function  $\tilde{\cdot}$  maps any site graph  $E$  and any transformation  $\sigma \in \tilde{\mathbb{G}}_E$  that can be applied to the site graph  $E$ , to the site graph  $\sigma.E$ ,
3. the function  $\tilde{\cdot}'$  maps any weak embedding  $f$  between two site graphs  $E$  and  $F$  and any transformation  $\sigma \in \tilde{\mathbb{G}}_F$  that can be applied to the site graph  $F$ , to the transformation  $f.\sigma$ ,
4. the function  $\tilde{\cdot}''$  maps any weak embedding  $f$  between two site graphs  $E$  and  $F$  and any transformation  $\sigma \in \tilde{\mathbb{G}}_F$  that can be applied to the site graph  $F$ , to the embedding  $\sigma.f$ ,

is a set of transformations over site graphs.

Now we define another set of transformations over site graphs, that we will call the set of homogeneous permutations of sites. Unlike a heterogeneous permutation of sites, a homogeneous permutation of sites consists in applying the same permutation to all the agents of a given connected component. The set of the homogeneous permutations of sites can be introduced as a subset of the heterogeneous permutations of sites.

**Exemple 6.2.2 (homogeneous permutations of sites)** With the same signature as in Exa. 6.2.1, we introduce the set of homogeneous permutations of sites as a subset of the set of the heterogeneous ones. More precisely, for any site graph  $G$ , we consider the subset  $\mathbb{G}_G^{\text{SPECIES}} \subseteq \mathbb{G}_G^{\text{AGENT}}$  of the heterogeneous permutations of sites  $\mathbb{G}_G^{\text{AGENT}}$  such that for any element  $\mathcal{X} \in \mathbb{G}_G^{\text{SPECIES}}$  and any pair of agents  $(n, n') \in [G]^2$ , if the agents  $n$  and  $n'$  are bound together (via their respective site  $z$ ) then  $n \in \mathcal{X}$  if, and only if,  $n' \in \mathcal{X}$ .

The orbit of the site graph that is denoted by the expression  $A_1(x_u, y_p, z^{A_2 @ z})$ ,  $A_2(x_u, y_p, z^{A_1 @ z})$  is given in Fig. 6.16(b).

Given a set of transformations over site graphs, two site graphs are said to be symetric if one is the image of the other one by a given transformation (in that set of transformations over site graphs). This is formalized in the following definition:

**Definition 6.2.3 (pairs of symetric site graphs)** Let  $\mathbb{G} = ((\mathbb{G}_E, \circ_E), \cdot, \cdot', \cdot'')$  be a set of transformations over site graphs. We say that two graphs  $E$  and  $F$  are  $\mathbb{G}$ -symetric to one another with respect to the set of transformations over site graphs  $\mathbb{G}$  if and only if there exists  $\sigma_E \in \mathbb{G}_E$  such that  $F = \sigma_E.E$ .

In such a case, we write  $EF$ .

**Propriété 6.2.2** Let  $\mathbb{G}$  be a set of transformations over site graphs. The relation  $\sim$  is an equivalence relation over site graphs.

The same way, two weak embeddings are said to be symetric if one is the image of the other one by a given transformation. This is formalized in the following definition:

**Definition 6.2.4 (pairs of symetric weak embeddings)** Let  $\mathbb{G} = ((\mathbb{G}_E, \circ_E), \cdot, \cdot', \cdot'')$  be a set of transformations over site graphs. We say that two embeddings  $f : EF$  and  $g : GH$  are  $\mathbb{G}$ -symetric to one another with respect to the set of transformations over site graphs  $\mathbb{G}$  if and only if there exists  $\sigma_F \in \mathbb{G}_F$  such that  $g = \sigma_F.f$ .

In such a case, we write  $fg$ .

**Propriété 6.2.3** Let  $\mathbb{G}$  be a set of transformations over site graphs. The relation  $\sim$  is an equivalence relation over weak embeddings.

Given a weak embedding  $f$  between the site graph  $E$  and the site graph  $F$ , and a transformation over site graphs  $\sigma_F \in \mathbb{G}_F$ . The transformation  $\sigma_F$  can be applied to the image  $F$  of the weak embedding  $f$  and the result is written as  $\sigma_F.F$ ; the transformation  $\sigma_F$  can also be restricted to the domain  $E$  of the weak embedding  $f$  and the result is written  $\sigma_F.''f$  (or  $\sigma_F.''f$ ). The constraint on  $.$  in Def. 6.2.1 ensures that  $\sigma_F.''f$  is a weak embedding between  $(f.\sigma).E$  and  $\sigma_F.F$ . Yet, there may be several ways to restrict a transformation to the domain of a weak embedding, because a transformation over site graphs can be described as the product of two others or because the weak embedding can be described as the composition of two others. In such a case, it is important to get the same result. Thus, we need to strengthen Def. 6.2.1. This is actually the purpose of the two following definitions.

**Definition 6.2.5 (distributivity of the restriction of the transformations over site graphs, over the product between transformations)**

Let  $\mathbb{G}$  be a set of transformations over site graphs. We say that restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings distributes over the product between these transformations, if and only if, for any two site graphs  $E$  and  $F$ , any weak embedding  $f$  between the site graph  $E$  and the site graph  $F$ , the following constraints are satisfied:

1. the function mapping each pair  $(\sigma_F, f')$  such that  $\sigma_F \in \mathbb{G}_F$  and  $f'$  is a weak embedding such that  $(f')F$ , to the weak embedding  $\sigma_F.''f''$ , is a group action;
2.  $f.\varepsilon_F = \varepsilon_E$ , where  $\varepsilon_F$  (resp.  $\varepsilon_E$ ) is the identity element of the group  $\mathbb{G}_F$  (resp.  $\mathbb{G}_E$ );
3.  $f.(\sigma' \circ_F \sigma) = ((\sigma.''f).\sigma') \circ_E (f.\sigma)$ .

**Definition 6.2.6 (distributivity of the restriction of transformations over site graphs, over the composition of transformations)**

Let  $\mathbb{G}$  be a set of transformations over site graphs. We say that the restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings distributes over the composition of these weak embeddings, if and only if, for any site graphs  $E, F, G$ , any weak embeddings  $f$  between the site graph  $E$  and the site graph  $F$ , and  $g$  between the site graph  $F$  and the site graph  $G$ , and any transformation  $\sigma_G \in \mathbb{G}_G$  that can be applied to the site graph  $G$ , the following constraints hold:

1.  $G.\sigma_G = \sigma_G$ ;
2.  $\sigma_G.''G = \sigma_G.G$ ;
3.  $(gf).\sigma_G = f.(g.\sigma_G)$ ;
4. and  $\sigma_G.''(gf) = (\sigma_G.''g)((g.\sigma_G)''f)$ .

The purpose of Def. 6.2.5 is to ensure that the operation of restricting the transformations that can be applied to the image of weak embeddings to their domains is compatible with the group structure of the transformations over site graphs, as illustrated in Fig. 6.18. Let us explain this definition. We consider  $\mathbb{G}$  a group of transformations over site graphs such that the restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings distributes over the product between these transformations. According to Def. 6.2.1, for any site graph  $E$ , the function  $.$  is a group action between the group  $\mathbb{G}_E$  of the transformations which can be applied to the site graph  $E$  and the site graphs that are  $\mathbb{G}$ -symetric with the site graph  $E$ . Thus, applying the identity element to a site graph does not change this site graph, and one can apply a product of two transformations to a site graph, or apply successively these two transformations in the reverse order and gets the same site graph as result. Def. 6.2.5 ensures the same properties over the application of symetries to weak embeddings: for any weak embedding  $f$  between  $E$  and  $F$ , the function  $.$  is a group action between the group  $\mathbb{G}_F$  of the transformations which can be applied to the embedding  $f$  and the weak embeddings that are  $\mathbb{G}$ -symetric with  $f$ . Lastly, the constraints 2 and 3 in Def. 6.2.5 ensure that restricting transformations over site graphs to the domain of weak embeddings is compatible with the group structure of transformations over site graphs. More precisely, the constraint 2 of Def. 6.2.5 ensure that the restriction the identity element of the group  $\mathbb{G}_F$  to the domain  $E$  of a weak embedding embedding  $f$  between the site graphs  $E$  and  $F$ , is the identity element of the group  $\mathbb{G}_E$  of the transformations that can be applied to the site graph  $E$  (eg. see Fig. 6.18(a)). Moreover, the constraint 3 in Def. 6.2.5 ensures, that the restriction a product  $\sigma'_F \circ \sigma_F$  between two transformations that can be applied to the site graph  $F$  to the domain of a weak embedding  $f$  between two site graphs  $E$  and  $F$ , can be done in two equivalent ways (eg. see Fig. 6.18(b)): either in one step, or in two steps. In one step, we get the transformation  $f.(\sigma'_F \circ \sigma_F)$  that can be applied to the site graph  $E$ . In two steps, firstly, we restrict



the transformation  $\sigma_F$  to the domain  $E$  of the weak embedding  $f$ . Doing this, we get the transformation  $f.\sigma_F$  that can be applied to the site graph  $E$ . Then we restrict the transformation  $\sigma_F$  to the domain of the weak embedding  $\sigma_F.''f$  (which is the image by the transformation  $\sigma_F$  of the weak embedding  $f$ ), and get the transformation  $\sigma_F.'(\sigma_F.''f)$  that can be applied to any site graph that is  $\mathbb{G}$ -symmetric to the site graph  $E$ . Then we make the product between the transformation  $\sigma_F.'(\sigma_F.''f)$  and the transformation  $f.\sigma_F$  in  $\mathbb{G}_E$ .

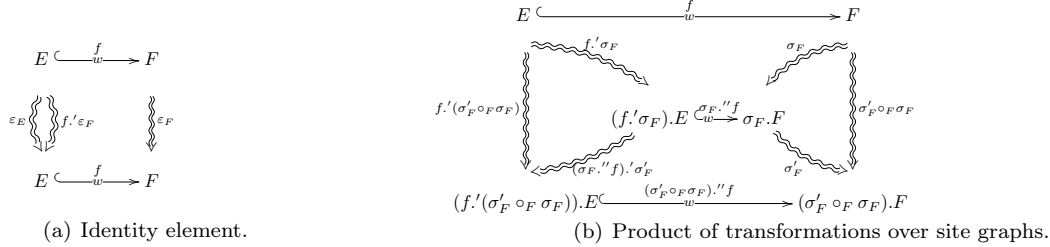


Figure 6.18: Commutative diagrams that describe the compatibility between the restriction of transformations over site graphs to the domain of weak embeddings and the group structure of these transformations.

Applying transformations over site graphs to weak embeddings should also be compatible with the composition of weak embeddings. This is ensured by Def. 6.2.6, which is illustrated in Fig. 6.19. More precisely, Fig. 6.19(a) describes what happens when we apply a transformation between site graphs to an identity embedding: restricting a transformation  $\sigma_G \in \mathbb{G}_G$  that can be applied to a site graph  $G$  to the domain of the identity embedding  $G$  over the site graph  $G$  does not change the transformation  $\sigma_G$  (constraint 1 of Def. 6.2.6) and the image of the identity embedding  $G$  over a site graph  $E$  by a transformation  $\sigma_G$  that can be applied to the site graph  $G$ , is the identity embedding  $\sigma_G.G$  over the image  $\sigma_G.G$  of the site graph  $G$  by the transformation  $\sigma_G$  (constraint 2 of Def. 6.2.6). Moreover, we can also restrict a transformation  $\sigma_G \in \mathbb{G}_G$  over site graphs to the domain of the composition of two weak embeddings  $gf$ , in two different ways, either in one step, or in two steps (by firstly restricting it to the domain of  $f$  and then restricting the result to the domain of  $g$ ). The constraint 3 of Def. 6.2.6 ensures that the two so obtained transformations  $(gf).\sigma_G$  and  $f.\sigma_G$  are equal. Moreover, the constraint 4 of Def. 6.2.6 the image of the composition  $gf$  of the weak embeddings  $g$  and  $f$  by the transformation  $\sigma_G$  is equal to the composition of the weak embedding  $\sigma_G.''g$  which is the image of the weak embedding  $g$  by the transformation  $\sigma_G$  and the weak embedding  $(g.\sigma_G)''f$  which is the image of the weak embedding  $f$  by the transformation  $g.\sigma_G$  that is obtained by restricting the transformation  $\sigma_G$  to the domain of the weak embedding  $g$ .

In particular, whenever the restriction of transformations over site graphs to the domain of weak embeddings distributes over the composition of weak embeddings, one can safely apply a transformation over a commuting square of weak embeddings as showed in the following Lemma.

**Lemme 6.2.1** *Let  $\mathbb{G}$  be a set of transformations over site graphs such that the restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings, distributes over the composition of these weak embeddings. Let  $E, F, F'$  and  $G$  be four site graphs, and  $f : EF, f' : EF', g : FG$ , and  $g' : F'G$  be four weak embeddings such*

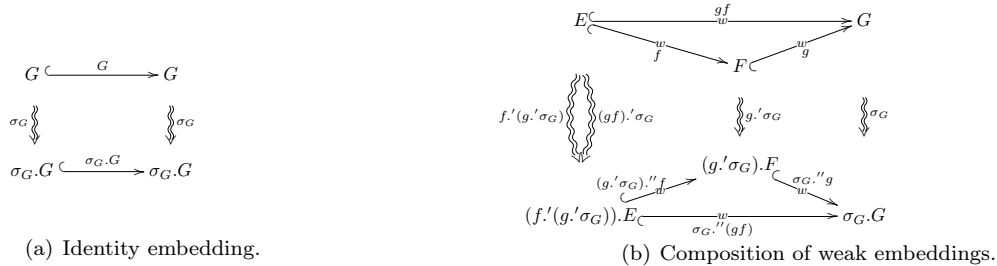


Figure 6.19: Commutative diagrams that describe the compatibility between the application of transformations between site graphs and the composition of weak embeddings.

that:  $gf = g'f'$ , as depicted in the following commutative diagram:

$$\begin{array}{ccc} F & \xrightarrow{g} & G \\ \uparrow f\psi & & \uparrow \psi g' \\ E & \xrightarrow{f'} & F' \end{array}$$

Let  $\sigma_G \in \mathbb{G}_G$  be a transformation that can be applied to the site graph  $G$ .

Under these assumptions, we have:

$$\sigma_G g f = \sigma_G g' f',$$

as depicted in the following commutative diagram:

$$\begin{array}{ccc} (g.'\sigma_G).F & \xrightarrow{\sigma_G.'g} & \sigma_G.G \\ \uparrow (g.'\sigma_G).''f\psi & & \uparrow \psi\sigma_G.'g' \\ ((gf).'\sigma_G).E & \xrightarrow{(g'.'\sigma_G).''f} & (g'.'\sigma_G).F' \end{array}$$

Now we introduce some additional constraints so as to ensure that the transformations between site graphs preserve the basic objects of our semantics.

Since fully specified site graphs are key elements in the definition of the states of the semantics. We have to assume that the image of a fully specified site graph by a transformation over site graphs is a fully specified site graph which is the goal of the following definition.

**Definition 6.2.7 (preservation of fully specified site graphs)** *Let  $\mathbb{G}$  be a set of transformations over site graphs. We say that the transformations in  $\mathbb{G}$  preserve fully specified site graphs, if and only if, for any fully specified site graph  $E$  and any transformation  $E \in \mathbb{G}_E$  that can be applied to the site graph  $E$ , the site graph  $\sigma_E.E$  is fully specified as well.*

Embeddings are crucial as well in the definition of computation steps, and we have to assume that the image of an embedding by a transformation over site graphs is an embedding as well, which is the purpose of the following definition.

**Definition 6.2.8 (preservation of embeddings)** *Let  $\mathbb{G}$  be a set of transformations over site graphs. We say that the transformations in  $\mathbb{G}$  preserve embeddings, if and only if, for any embedding  $f$  and any transformation  $\sigma_{(f)} \in \mathbb{G}_{(f)}$  that can be applied to the embedding  $f$ , the weak embedding  $\sigma_{(f)}.f$  is an embedding as well.*

We can simultaneously apply a pair of transformations to the left hand side and to the right of side of a partial embedding  $\phi$ , providing that these transformations have the same restriction over the domain of the partial embedding. Yet, it is important that the set of pairs of transformations that can be applied to a partial embedding, forms a group. That is why we assume in the following definition, that the set of pairs of transformations that can be applied to a given partial embedding is stable upon pairwise product.

**Definition 6.2.9 (composition of the pairs of transformations over site graphs, that can be applied to a partial embedding)** *Let  $\mathbb{G}$  be a set of transformations over site graphs. We say that the pairs of the transformations in  $\mathbb{G}$  that can be applied to partial embeddings compose pairwise, if and only if, for any partial embedding*

$$L[h_L]D[h_R]R,$$

*any four transformations  $\sigma_L \in \mathbb{G}_L$ ,  $\sigma'_L \in \mathbb{G}_L$ ,  $\sigma_R \in \mathbb{G}_R$ , and  $\sigma'_R \in \mathbb{G}_R$  over site graphs, such that  $h_L.'\sigma_L = h_R.'\sigma_R$  and  $h_L.'\sigma'_L = h_R.'\sigma'_R$ , we have  $h_L.'(\sigma'_L \circ_L \sigma_L) = h_R.'(\sigma'_R \circ_R \sigma_R)$ .*

Whenever in a set of transformations, the restriction of transformations to the domain of weak embeddings distributes over both the product between these transformations and the composition between embeddings (e.g. see Def. 6.2.5) and Def. 6.2.6), the fact that the pairs of the transformations that can be applied to partial embeddings compose pairwise, is equivalent to two simpler alternative properties, which are given in the next proposition.

**Propriété 6.2.4** *Let  $\mathbb{G}$  be a set of transformations over site graphs. We assume that the restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings distributes both over the product between these transformations (e.g. see 6.2.5) and over the composition of weak embeddings (e.g. see Def. 6.2.6).*

*Under this assumption, the following three assertions are equivalent:*

1. *The pairs of transformations in  $\mathbb{G}$  that can be applied to partial embeddings compose pairwise (eg. see Def. 6.2.9).*
2. *For any two site graphs  $E$  and  $F$ , and any embedding  $f$  between the site graph  $E$  and  $F$ , any two transformations  $\sigma_F, \sigma'_F \in \mathbb{G}_F$  that can be applied to the site graph  $F$ , we have:  $f.'(\sigma'_F \circ_F \sigma_F) = (f.'\sigma'_F) \circ_E (f.'\sigma_F)$ .*
3. *For any two site graphs  $E$  and  $F$ , and any embedding  $f$  between the site graph  $E$  and  $F$ , any two transformations  $\sigma_F, \sigma'_F \in \mathbb{G}_F$  that can be applied to the site graph  $F$ , we have:  $(\sigma'_F.''f)'\sigma_F = f.'\sigma_F$ .*

In Prop.6.2.4, the second assertion can be interpreted as the fact that each embedding  $f : EF$  between two site graphs  $E$  and  $F$ , the function that maps each transformation  $\sigma_F \in \mathbb{G}_F$  that can be applied to the site graph  $F$ , into the transformation  $f.'\sigma_F \in \mathbb{G}_E$  is a morphism between the group  $\mathbb{G}_F$  of the transformations that can be applied to the site graph  $F$  and the group  $\mathbb{G}_E$  of the transformations that can be applied to the site graph  $E$ ; the third assertion can be interpreted as the fact that for any embedding  $f : EF$  between two site graphs  $E$  and  $F$  and any transformation  $\sigma_F$  that can be applied to the site graph  $F$ , then the embedding  $f$  and its image  $\sigma_F.''f$  by the transformation  $\sigma_F$  operates the same way with the group  $\mathbb{G}_F$  of the transformations that can be applied to the site graph  $F$ .

We introduce explicitly these two new notions in the following definitions.

**Definition 6.2.10 (morphism induced by an embedding)** *Let  $\mathbb{G}$  be a set of transformations over site graphs and  $f : E \llbracket F$  be an embedding between two site graphs  $E$  and  $F$ . We say that the embedding  $f$  induces a morphism between the group of the transformations  $\mathbb{G}_F$  that can be applied to the site graph  $F$  and the group of the transformations  $\mathbb{G}_E$  that can be applied to the site graph  $E$ , if and only if, the function  $h_f$  between the group  $\mathbb{G}_F$  and the group  $\mathbb{G}_E$ , mapping each transformation  $\sigma_F \in \mathbb{G}_F$  into the transformation  $f.'\sigma_F$ , is a morphism.*

**Definition 6.2.11 (invariance of the action on an embedding over transformations)** *Let  $\mathbb{G}$  be a set of transformations over site graphs. An embedding  $f : EF$  between two site graphs  $E$  and  $F$  operates the same way over the transformations which can be applied to the site graph  $F$ , as any of its image by such a transformation if and only if for any pair of transformations  $(\sigma, \sigma') \in \mathbb{G}_F^2$  that can be applied to the site graph  $F$ , we have:  $(\sigma'.''f)'\sigma = f.'\sigma$ .*

Lastly, we have to assume that the image of a rule by a transformation over site graphs, is a rule and that transformations over site graphs, preserve the existence of refinements. This is the purpose of the next two definitions.

**Definition 6.2.12 (preservation of rules)** *Let  $\mathbb{G}$  be a set of transformations over site graphs. We say that the transformations in  $\mathbb{G}$  preserve rules, if and only if, for any rule*

$$L[h_L]D[h_R]R$$

*and any pair of transformations  $(\sigma_L, \sigma_R) \in \mathbb{G}_L \times \mathbb{G}_R$  that can be applied respectively to the left hand side  $L$  of the rule and to the right hand  $R$  side of this rule and such that  $h_L.'\sigma_L = h_R.'\sigma_R$ , the partial embedding*

$$\sigma_L.L[\sigma_L.''h_L](h_L.'\sigma_L).D[\sigma_R.''R]\sigma_R.R$$

*is a rule as well.*

**Definition 6.2.13 (preservation of the existence of refinements)** *Let  $\mathbb{G}$  be a set of transformations over site graphs that preserve rules, we say that the transformations in  $\mathbb{G}$  preserve the existence of refinements as well, if and only if, for any rule*

$$r : L[h_L]D[h_R]R,$$

*for any embedding  $h_{L'}$  between the site graphs  $L$  and a site graph  $L'$ , and any pair of transformations over site graphs  $(\sigma_{L'}, \sigma_R) \in \mathbb{G}_{L'} \times \mathbb{G}_R$  that can be applied respectively to the site graph  $L'$  and to the site graph  $R$ , such that:*

$$1. (h_{L'}h_L).'\sigma_{L'} = h_R.'\sigma_R,$$

2. there exist a rule  $r'$  and an embedding  $h_{R'}$  such that the tuple  $(r, r', h_{L'}, h_{R'})$  is a refinement,

there exist a rule  $r'_\sigma$  and an embedding  $h_{R'_\sigma}$  such that the tuple  $(r_\sigma, r'_\sigma, \sigma_{L'}, h_{L'}, h_{R'_\sigma})$  is a refinement as well, where the  $r_\sigma$  is defined as the rule:  $(\sigma_{L'}.''h_{L'}).L[(h_{L'}.'\sigma_{L'}).''h_L](h_R.'\sigma_R).D[\sigma_R.''h_R]\sigma_R.R$ .

We are now ready to define the valid sets of transformations over site graphs as those which satisfy the additional assumptions that we have given in Definitions 6.2.5–6.2.13.

**Definition 6.2.14 (valid set of transformations over site graphs)** *We say that a set of transformations  $\mathbb{G}$  over site graphs is valid, if and only if,*

1. the transformations in  $\mathbb{G}$  distribute over the composition of weak embeddings;
2. the restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings distributes over the product between these transformations;
3. the transformations over site graphs, in  $\mathbb{G}$ , preserve fully specified site graphs;
4. the transformations over site graphs, in  $\mathbb{G}$ , preserve embeddings;
5. the pairs of transformations over site graphs, in  $\mathbb{G}^2$  that can be applied to partial embeddings compose pairwise;
6. each embedding between two site graphs induces a morphism between the group of the transformations that can be applied to the image of this embedding, and the group of the transformations that can be applied to the domain of this embedding;
7. each embedding operates the same way over the transformations that can be applied to its image, as any of its image by such a transformation;
8. the transformation over site graphs, in  $\mathbb{G}$ , preserve rules;
9. the transformation over site graphs, in  $\mathbb{G}$ , preserve the existence of refinements.

As noticed in Prop. 6.2.4, whenever the conditions 1 and 2 in Thm.6.2.14 are satisfied, the conditions 5, 6, and 7 are equivalent. Thus in order to prove that a set of transformations is valid, it is not necessary to prove all of the three properties 5, 6, and 7 (only one among them is enough).

Looking at our favorites examples of set of transformations over site graphs, we can check that the sets of transformations  $\mathbb{G}^{\text{AGENT}}$  and  $\mathbb{G}^{\text{SPECIES}}$  that we have introduced in Exa. 6.2.1 and in Exa. 6.2.2 are both valid. Moreover, there is no need to check the assumptions of Def. 6.2.14 for proving that the set  $\mathbb{G}^{\text{SPECIES}}$  of homogeneous permutations of sites is valid, since in a subset of transformations over site graphs, these assumptions follow from the overlying structure.

**Exemple 6.2.3** *The set  $\mathbb{G}^{\text{AGENT}}$  of the heterogeneous permutations of sites (see Exa. 6.2.1) is a valid set of symetries.*

**Propriété 6.2.5** *A subset of a valid set transformations over site graphs is a valid set of transformations over site graphs, as well.*

**Exemple 6.2.4** *As a direct consequence of Prop. 6.2.5, the set  $\mathbb{G}^{\text{SPECIES}}$  of the homogeneous permutations of sites (see Exa. 6.2.2) is a valid set of transformations over site graphs.*

## 6.2.2 Induced group actions

A valid set of transformations over site graphs induces group actions over the basic elements of the semantics of Kappa. We are going to itemise them in this subsection.

## Induced group action over fully specified site graphs

Def. 6.2.7 ensures that the image of a fully specified site graph by a transformation within a valid set of transformations over site graphs is a fully specified site graph. Thus, by Def. ??, the group action of a valid set of transformations over site graphs induces another group action over fully specified site graphs, as written in the following proposition.

**Propriété 6.2.6 (action of transformations over fully specified site graphs)** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs. For a given fully specified site graph  $E$ , the function mapping each pair  $(\sigma'_E, E') \in \mathbb{G}_E \times \{\sigma.E \mid \sigma \in \mathbb{G}_E\}$  to the fully specified site graph  $\sigma'_E.E'$ , is a group action.*

## Induced group action over embeddings

**Applying transformations to embeddings.** Def. 6.2.8 ensures that the image of an embedding, by a transformation in a valid set of transformations over site graphs, is always an embedding. Thus, by Def. ??, in a valid set of transformations over site graphs, the group action of transformations over weak embeddings induces another group action over embeddings, as written in the following proposition.

**Propriété 6.2.7 (action of transformations over embeddings)** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs. For a given embedding  $f$  between two site graphs  $E$  and  $F$ , the function mapping each pair  $(\sigma'_F, f') \in \mathbb{G}_F \times \{\sigma.f \mid \sigma \in \mathbb{G}_F\}$  to the embedding  $\sigma'_F.f'$ , is a group action.*

**Extending a transformation over the domain of an embedding.** In Def. 6.2.1, we have assumed that we can always restrict a transformation that can be applied to the image of a weak embedding into a transformation that can be applied to the domain of this weak embedding. Yet, for the sake of generality, we have not assumed that the converse operation is possible: it is not always possible to extend a transformation that can be applied to the domain of a weak embedding to a transformation that can be applied to the image of this weak embedding (e.g. see Fig. 6.20(b) for a counter example). Now we focus on the case of embeddings: given a set  $\mathbb{G}$  of transformations over site graphs, we say that a given embedding  $f$  is  $\mathbb{G}$ -compatible if and only if any transformation that can be applied to the domain of the embedding  $f$  can be extended into a transformation that can be apply to the image of the embedding  $f$ . Notice that we could have defined the notion of compatible weak embedding the same way, but we will not use this notion in the following.

**Definition 6.2.15 (compatible embedding)** *Let  $\mathbb{G}$  be a set of transformations over site graphs and  $f : E \llbracket F$  be an embedding between two site graphs  $E$  and  $F$ . We say that the embedding  $f$  is  $\mathbb{G}$ -compatible if and only if for any transformation  $\sigma_E \in \mathbb{G}_E$ , there exists a transformation  $\sigma_F \in \mathbb{G}_F$  such that  $f.\sigma_F = \sigma_E$ .*

**Exemple 6.2.5** *We consider the set  $\mathbb{G}^{\text{AGENT}}$  of heterogeneous permutations of site (as defined in Exa. 6.2.1). It turns out that every embedding is  $\mathbb{G}^{\text{AGENT}}$ -compatible. Indeed, let us consider  $f$  an embedding  $EF$  between two site graphs  $E$  and  $F$ , and  $\sigma_E$  be a symetry in  $\mathbb{G}_E^{\text{AGENT}}$ . With our notations, the symetry  $\sigma_E$  is encoded as a subset of  $[E]$ . Then, the symetry  $\sigma_F \in \mathbb{G}_F^{\text{AGENT}}$  that is defined as the set  $\{f(a) \mid a \in \sigma_E\}$  satisfying the property:  $f.\sigma_F = \sigma_E$ .*

*We illustrate this on a particular example in Fig. 6.20(a). We take the same signature as in Exa. 6.2.1 and we consider two site graphs. The first one:*

$$E := A_1(x_u, y_p), A_2(x_u, y_p)$$

*is made of two agents of type  $A$  with labels 1 and 2 and with a site  $x$  unphosphorylated and a site  $y$  phosphorylated. Their sites  $z$  are not documented. The second one:*

$$F := A_3(x_u, y_p, z^{A_4 @ z}), A_4(x_u, y_p, z^{A_3 @ z})$$

*is made of two agents of type  $A$  with label 3 and 4, with a site  $x$  unphosphorylated and a site  $y$  phosphorylated, and bound together via their respective site  $z$ .*

*The site graph  $E$  can be embedded in the site graph  $F$  by the following embedding:*

$$f := [(A, 1) \mapsto 3, (A, 2) \mapsto 4].$$

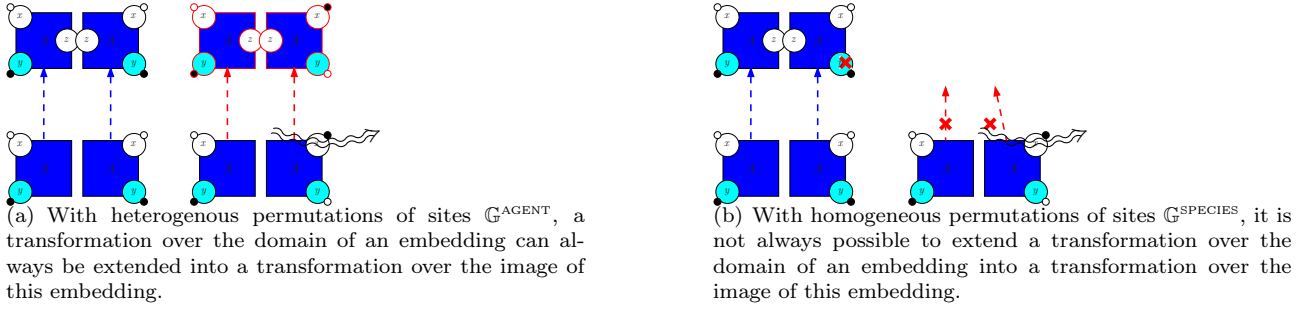


Figure 6.20: An example of an embedding which is  $\mathbb{G}^{\text{AGENT}}$ -compatible (i.e. each heterogeneous permutation of sites, as defined in Exa. 6.2.1, that can be applied to its domain can be extended into an heterogeneous permutation of sites that can be applied to its image), but not  $\mathbb{G}^{\text{SPECIES}}$ -compatible (i.e. there exists some homogeneous permutations of sites, as defined in Exa. 6.2.2). We take the site graph that is made of two agents of type  $A$  with a site  $x$  unphosphorylated and a site  $y$  phosphorylated, the embedding that adds a site  $z$  to both of these agents and binds these two sites  $z$  together, hence merging the two connected components of the site graph, and the transformation that swaps the state of the sites  $x$  and  $y$  of the agent on the right. With heterogeneous permutations of sites, permutations can be chosen independently for each agent, no matter they belong, or not, to the same connected component. Thus, the transformation that is applied to the domain of the embedding can be extended into a transformation that can be applied to the image of the embedding, as shown in Fig. 6.20(a). With homogeneous permutations, for each connected components, the same permutation must be applied to all the agents with a same type in a that connected component. Since the two connected components of the initial site graphs have been merged, it is no longer possible to swap the state of the sites  $x$  and  $y$  in the agent on the right, without swapping those of the agent on the left. Thus it is impossible to extend the transformation that is applied to the domain of the embedding into a transformation that is applied to the image of the embedding, as shown in Fig. 6.20(b).



Figure 6.21: An example of transformation that can be applied to the domain of an embedding and which can be extended into a transformation that can be applied to the image of this embedding in various ways. We consider a single agent of type  $A$  with a site  $x$  unphosphorylated and a site  $y$  phosphorylated. We consider the transformation in  $\mathbb{G}^{\text{AGENT}}$  which consists in swapping the state of the sites  $x$  and  $y$  of the unique agent of this site graph. Then we introduce the embedding that adds a site  $z$  to this agent and binds it to the site  $z$  of a new agent of type  $A$  also with the site  $x$  unphosphorylated and the site  $y$  phosphorylated. When we extend the transformation to the image of the embedding, we have to swap the state of the sites  $x$  and  $y$  of the agent on the right, since it is the image of the unique agent of the domain of the embedding. But for the agent on the left, we are free to swap, or not, the state of these sites, which gives two potential extensions of the embedding: in Fig. 6.21(a), we have swapped the states of the sites  $x$  and  $y$ , whereas in Fig. 6.21(b), we leave them unchanged.

Now we apply the permutation  $\{2\} \in \mathbb{G}_E^{\text{AGENT}}$  to the site graph  $E$ . This permutation consists in swapping the state of the site  $x$  and  $y$  in the agent of  $E$  with the label 2. Thus we get the following site graph:

$$\{2\}.E = A_1(x_u, y_p), A_2(x_p, y_u).$$

We would like to extend the permutation  $\{2\}$  to the image  $F$  of the embedding  $f$ . Since each agent of  $F$  is the image of an agent of  $E$ , we have no choice: we have to permute the states of the sites  $x$  and  $y$  of the agent with the label 4, since it is the image of the agent with the label 2 and because the label 2 belongs to the set  $\{2\}$ ; and we have to keep the states of the sites  $x$  and  $y$  of the agent with the label 3, since it is the image of the agent with the label 1 and because the label 1 does not belong to the set  $\{2\}$ . Thus, we get the permutation  $\{4\} \in \mathbb{G}_F^{\text{AGENT}}$  which transform the site graph  $F$  into the following site graph:

$$\{4\}.F = A_3(x_u, y_p, z^{A_4 @ z}), A_4(x_p, y_u, z^{A_3 @ z}).$$

The image of the embedding  $f$  by the permutation  $\{4\}$  is defined by the same mapping as the embedding  $f$ :

$$\{4\}.f = [(A, 1) \mapsto 3, (A, 2) \mapsto 4].$$

We can check that  $\{4\}.f$  is indeed an embedding between the site graph  $\{2\}.E$  and the site graph  $\{4\}.F$ .

**Example 6.2.6** Now we consider the set of homogeneous (as defined in Exa. 6.2.2), we notice that there exists embeddings which are not  $\mathbb{G}^{\text{SPECIES}}$ -compatible.

A counter example is given in Fig. 6.20(b). We consider the same graphs  $E$  and  $F$  and the same embedding  $f$  as in Exa. 6.2.5. But now we see the transformation  $\{2\}$  as an element of set  $\tilde{\mathbb{G}}_E$  of the homogeneous permutations of sites, that can be applied to the site graph  $E$ .

We want to extend the transformation  $\{2\}$  to a permutation of sites that can be apply to the image  $F$  of the embedding  $f$ . As it was the case for the heterogeneous permutations of site (see Exa. 6.2.5), if such a permutation of site were existing, it would necessarily swap the states of the site  $x$  and  $y$  of the agent with the label 4 and necessarily leave the states of these sites unchanged in the agent with the label 3. Yet, the agents with the labels 3 and 4 belongs to the same connected component in the site graph  $H$ , thus an homogeneous permutation that can be applied to the site graph  $F$  either swap the states of the sites  $x$  and  $y$  both in the agents with labels 3 and 4, or leave the states of these four sites unchanged. As a consequence, there is no homogenous permutation of sites which extend the permutation  $\{2\}$  that can be applied to domain  $E$  of the embedding  $f$ , into an homogeneous permutation of sites which can be applied to image  $H$  of the embedding  $f$ .

We have seen in Exa. 6.2.6 that the fact that an embedding is compatible with a set of symetries, does not entail that this embedding is compatible with a subset of this set of symetries.

In general, the extension of a transformation that can be applied to the domain of an embedding into a transformation that can be applied to the image of an embedding, if it exists, is not unique. We give in Exa. 6.2.7, an example in which there exists two distinct ways to extend such a transformation.

**Example 6.2.7** We show in Fig. 6.21 such an example of a transformation that can be applied to the domain of an embedding which can be extended into two distinct transformations that can both be applied to the image of this embedding.

We take the same signature as in Exa. 6.2.1 and we consider two site graphs. The first one:

$$G := A_2(x_u, y_p)$$

is made of one agent of type  $A$  with label 2 and with a site  $x$  unphosphorylated and a site  $y$  phosphorylated. Its site  $z$  is not documented. The second one:

$$H := A_3(x_u, y_p, z^{A_4 @ z}), A_4(x_u, y_p, z^{A_3 @ z})$$

is made of two agents of type  $A$  with label 3 and 4, with a site  $x$  unphosphorylated and a site  $y$  phosphorylated, and bound together via their respective site  $z$ .

We consider the following embedding  $g$  between the site graphs  $G$  and  $H$ ,

$$g := [(A, 2) \mapsto 4],$$

which maps the unique agent of the site graph  $G$  to the agent with the label 4 in the site graph  $H$ .

Now we apply the permutation  $\{2\} \in \mathbb{G}_G^{\text{AGENT}}$  to the site graph  $G$ . This permutation consists in swapping the state of the site  $x$  and  $y$  in the unique agent of the site graph  $G$ . Thus we get the following site graph:

$$\{2\}.G = A_2(x_p, y_u).$$

We would like to extend the permutation  $\{2\}$  to the image  $H$  of the embedding  $g$ . The agent with the label 4 in the site graph  $H$ , is the image of the unique agent of the site graph  $G$ . Thus, if it exists, our new transformation has to swap the states of the sites  $x$  and  $y$  of the agent with the label 4 in the site graph  $H$ . But for the agent with the label 3, we are not constrained, because it is the image of no agent of the site graph  $G$ . Thus, either we swap also the states of the sites  $x$  and  $y$  of the agent of the site graph  $H$  with the label 3 (e.g. see Fig. 6.21(a)):

$$\{3, 4\}.H = A_3(x_p, y_u, z^{A_4 \otimes z}), A_4(x_p, y_u, z^{A_3 \otimes z})$$

or we keep the states of these sites unchanged (e.g. see Fig. 6.21(b)):

$$\{4\}.H = A_3(x_u, y_p, z^{A_4 \otimes z}), A_4(x_p, y_u, z^{A_3 \otimes z}).$$

**Decomposing a set of transformations along an embedding.** In a valid set of transformations, an embedding  $f$  between two site graphs  $E$  and  $F$  induces a morphism between the group of the transformations  $\mathbb{G}_F$  that can be applied to the image  $F$  of the embedding  $f$ , and the group of the transformations  $\mathbb{G}_E$  that can be applied to the domain  $E$  of the embedding  $f$  (e.g. see Def. 6.2.10). Thanks to this property, we can properly decompose the group of transformations that can be applied to the image of an embedding into two subgroups, by applying the first isomorphism theorem (e.g. see Thm.??).

**Théorème 6.2.1** *Let  $\mathbb{G}$  be a set of transformations over site graphs and  $f$  be an embedding between two site graphs  $E$  and  $F$ .*

*Then three assertions are satisfied:*

1. *The set*

$$\{f.'\sigma_F \mid \sigma_F \in \mathbb{G}_F\}$$

*and the set*

$$\mathbb{G}_F \setminus \{\sigma_F \in \mathbb{G}_F \mid f.'\sigma_F = \varepsilon_E\}$$

*are two isomorphic groups.*

2. *The number of transformations in the group  $\mathbb{G}_F$  is equal to the product between the cardinal of the group*

$$\{f.'\sigma_F \mid \sigma_F \in \mathbb{G}_F\}$$

*and the cardinal of the group*

$$\{\sigma \in \mathbb{G}_F \mid f.'\sigma = \varepsilon_E\};$$

3. *For any transformation  $\sigma_E \in \mathbb{G}_E$ , the set*

$$\{\sigma_F \in \mathbb{G}_F \mid f.'\sigma_G = \sigma_E\}$$

*is either empty, or of the form:*

$$\{\sigma_F \circ_F \sigma'_F \mid \sigma'_F \in \mathbb{G}_F \text{ tel que } f.'\sigma'_F = \varepsilon_{\mathbb{G}_E}\},$$

*for a given transformation  $\sigma_G \in \mathbb{G}_G$ .*

In Thm. 6.2.1, the group  $\{f.'\sigma_F \mid \sigma_F \in \mathbb{G}_F\}$  denotes the set of the transformations that can be applied to the domain  $E$  of the embedding  $f$  and that can be extended to transformations over the image  $F$  of the embedding  $f$ , whereas the group  $\{\sigma_F \in \mathbb{G}_F \mid f.'\sigma_F = \varepsilon_E\}$  is the kernel of morphism  $h_f$  that is induced by the embedding  $f$  and denotes the set of transformations over the image  $F$  of the embedding  $f$  that do not modify the domain  $E$  of the embedding  $f$  (when we restrict this transformation to the domain  $E$  of the embedding  $f$ ).

Now we illustrate the usefulness of Thm.6.2.1, by applying it to the embedding  $f$  between the site graphs  $E$  and  $F$  that we had introduced in Exa. 6.2.5 (e.g. see Fig. 6.20) and to the embedding  $g$  between the site graphs  $G$  and  $H$  that we had introduced in Exa. 6.2.7 (e.g. see Fig. 6.21). For each of these two embeddings, we use Thm. 6.2.1 to decompose the set of the heterogeneous (resp. homogeneous) permutations of sites  $\mathbb{G}^{\text{AGENT}}$  (resp.  $\mathbb{G}^{\text{SPECIES}}$ ) that can be applied to the image of this embedding.



**Exemple 6.2.8 (Decomposition of the group  $\mathbb{G}_F^{\text{AGENT}}$ )** We consider the site graphs:

$$E := A_1(x_u, y_p), A_2(x_u, y_p)$$

and

$$F := A_3(x_u, y_p, z^{A_4 \otimes z}), A_4(x_u, y_p, z^{A_3 \otimes z}),$$

and the embedding:

$$f := [(A, 1) \mapsto 3, (A, 2) \mapsto 4],$$

as in Exa. 6.2.5 (also see Fig. 6.20). A heterogeneous permutations of sites which can be applied to the site graph  $F$  consists in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 3, and in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 4. Thus we have:

$$\mathbb{G}_F^{\text{AGENT}} = \wp(\{3, 4\}).$$

The same way, a heterogeneous permutations of sites which can be applied to the site graph  $E$  consists in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 1, and in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 2. Thus we have:

$$\mathbb{G}_E^{\text{AGENT}} = \wp(\{1, 2\}).$$

We have seen in Exa. 6.2.5, that any embedding is  $\mathbb{G}^{\text{AGENT}}$ -compatible, so the embedding  $f$  is  $\mathbb{G}^{\text{AGENT}}$ -compatible and any transformation in  $\mathbb{G}_E^{\text{AGENT}}$  can be extended into a transformation that can be applied to the site graph  $F$ .

By applying Thm. 6.2.1, since both the sets  $\mathbb{G}_F^{\text{AGENT}}$  and  $\{f.'\sigma_F \mid \sigma_F \in \mathbb{G}_F^{\text{AGENT}}\}$  have four elements, we can deduce that the set  $\{\sigma_F \in \mathbb{G}_F^{\text{AGENT}} \mid f.'\sigma_F = \emptyset\}$  contains only the identity element  $\emptyset$  (which is actually the case).

**Exemple 6.2.9 (Decomposition of the group  $\mathbb{G}_H^{\text{AGENT}}$ )** We consider the site graphs:

$$G := A_2(x_u, y_p)$$

and

$$H := A_3(x_u, y_p, z^{A_4 \otimes z}), A_4(x_u, y_p, z^{A_3 \otimes z}),$$

and the embedding:

$$g := [(A, 2) \mapsto 4],$$

as in Exa. 6.2.7 (also see Fig. 6.21). A heterogeneous permutations of sites which can be applied to the site graph  $H$  consists in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 3, and in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 4. Thus we have:

$$\mathbb{G}_H^{\text{AGENT}} = \wp(\{3, 4\}).$$

Beside, a heterogeneous permutation of sites which can be applied to the site graph  $G$  consists in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 2. Thus we have:

$$\mathbb{G}_G^{\text{AGENT}} = \{\emptyset, \{2\}\}.$$

We have seen in Exa. 6.2.5, that any embedding is  $\mathbb{G}^{\text{AGENT}}$ -compatible, so the embedding  $g$  is  $\mathbb{G}^{\text{AGENT}}$ -compatible and any transformation in  $\mathbb{G}_G^{\text{AGENT}}$  can be extended into a transformation that can be applied to the site graph  $H$ . Thus the set  $\{g.'\sigma_H \mid \sigma_H \in \mathbb{G}_H^{\text{AGENT}}\}$  has two elements. Since the set  $\mathbb{G}_H^{\text{AGENT}}$  has four elements and the set  $\{g.'\sigma_H \mid \sigma_H \in \mathbb{G}_H^{\text{AGENT}}\}$  has two elements, we can deduce from Thm. 6.2.1 that the set  $\{\sigma_H \in \mathbb{G}_H^{\text{AGENT}} \mid g.'\sigma_H = \emptyset\}$  has two elements. Indeed, the set  $\{\sigma_H \in \mathbb{G}_H^{\text{AGENT}} \mid g.'\sigma_H = \emptyset\}$  contains exactly the permutations of sites that leave unchanged all the agents in the image of the domain of the embedding  $g$ , that is to say the permutation  $\emptyset$  and the permutation  $\{3\}$ .

Said differently, we can use the embedding  $g$  to decompose the set  $\mathbb{G}_H^{\text{AGENT}}$  into the subgroup  $\{\emptyset, \{3\}\}$  that contains all the heterogeneous permutations of sites which do not change the agent that are in the image of  $G$  by the embedding  $g$ , and a subgroup that is isomorphic to the subgroup  $\{\emptyset, \{2\}\}$  that contains all the heterogeneous permutations of sites which can be applied to the domain  $G$  of the embedding.

**Exemple 6.2.10 (Decomposition of the group  $\mathbb{G}_F^{\text{SPECIES}}$ )** We consider the site graphs:

$$E := A_1(x_u, y_p), A_2(x_u, y_p)$$

and

$$F := A_3(x_u, y_p, z^{A_4 @ z}), A_4(x_u, y_p, z^{A_3 @ z}),$$

and the embedding:

$$f := [(A, 1) \mapsto 3, (A, 2) \mapsto 4],$$

as in Exa. 6.2.5 (also see Fig. 6.20). A homogeneous permutation of sites which can be applied to the site graph  $F$  consists either in swapping the states of the sites  $x$  and  $y$  in both agents of the site graph  $F$ , or in leaving the site graph  $F$  unchanged, because these agents belongs to a same connected component. Thus we have:

$$\mathbb{G}_F^{\text{AGENT}} = \{\emptyset, \{3, 4\}\}.$$

The two agents of the site graph  $E$  belongs to two different connected components, thus a homogeneous permutation of sites which can be applied to the site graph  $E$  consists in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 1, and in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 2. That is to say:

$$\mathbb{G}_E^{\text{AGENT}} = \wp(\{1, 2\}).$$

Not all the homogeneous permutation of sites that can be applied to the site graph  $E$  can be extended into a homogeneous permutation of sites that can be applied to the site graph  $F$ , only the permutation  $\emptyset$  and the permutation  $\{1, 2\}$  (which are mapped respectively into the permutations  $\emptyset$  and  $\{3, 4\}$ ).

By applying Thm. 6.2.1, since both the sets  $\mathbb{G}_F^{\text{AGENT}}$  and  $\{f.'\sigma_F \mid \sigma_F \in \mathbb{G}_F^{\text{AGENT}}\}$  have two elements, we can deduce that the set  $\{\sigma_F \in \mathbb{G}_F^{\text{AGENT}} \mid f.'\sigma_F = \varepsilon_E\}$  only the identity element  $\emptyset$  (which is actually the case).

**Exemple 6.2.11 (Decomposition of the group  $\mathbb{G}_H^{\text{SPECIES}}$ )** We consider the site graphs:

$$G := A_2(x_u, y_p)$$

and

$$H := A_3(x_u, y_p, z^{A_4 @ z}), A_4(x_u, y_p, z^{A_3 @ z}),$$

and the embedding:

$$g := [(A, 2) \mapsto 4],$$

as in Exa. 6.2.7 (also see Fig. 6.21). A homogeneous permutations of sites which can be applied to the site graph  $H$  consists either in swapping the states of the sites  $x$  and  $y$  in both agents of the site graph  $H$ , or in leaving the site graph  $H$  unchanged, because these agents belongs to a same connected component. Thus we have:

$$\mathbb{G}_H^{\text{AGENT}} = \{\emptyset, \{3, 4\}\}.$$

Beside, a homogeneous permutations of sites which can be applied to the site graph  $G$  consists in swapping or not the states of the sites  $x$  and  $y$  of the agent with the label 2. Thus we have:

$$\mathbb{G}_G^{\text{AGENT}} = \{\emptyset, \{2\}\}.$$

We notice that the homogeneous permutation  $\emptyset$  (resp.  $\{2\}$ ) that can be applied to the site graph  $G$ , can be extended into the homogeneous permutation  $\emptyset$  (resp.  $\{3, 4\}$ ) that can be applied to the site graph  $H$ .

By applying Thm. 6.2.1, since both the set  $\mathbb{G}_H^{\text{AGENT}}$  and  $\{f.'\sigma_F \mid \sigma_F \in \mathbb{G}_F^{\text{AGENT}}\}$  have two elements, we can deduce that the set  $\{\sigma_F \in \mathbb{G}_F^{\text{AGENT}} \mid f.'\sigma_F = \varepsilon_E\}$  contains only the identity element  $\emptyset$  (which can be checked easily).

## Induced group action over isomorphisms

**Applying transformations to isomorphisms** The image of an isomorphism by a transformation within a valid set of transformations over site graphs, is an isomorphism as well. Moreover, the image of an embedding by a transformation within a valid set of transformations over site graphs, is an isomorphism if and only if the former is an isomorphism as well. This is expressed in the following property.

**Propriété 6.2.8** *Let  $\mathbb{G}$  be a set of transformations over site graphs. We assume that the restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings distributes over the composition of weak embeddings (e.g. see 6.2.6) and that the image of an embedding by a transformation in  $\mathbb{G}$  is an embedding (e.g. see 6.2.8). Let  $f$  be an embedding between two site graphs  $E$  and  $F$ , and  $\sigma_F \in \mathbb{G}_F$  be a transformation that can be applied to the site graph  $F$ .*

*Then the following assertions are equivalent:*

1. *the embedding  $f$  is an isomorphism.*
2. *the embedding  $\sigma_F."f$  is an isomorphism.*

*In such a case, the inverse of the isomorphism  $\sigma_F."f$  is the isomorphism  $(f.'\sigma_F)."f^{-1}$ .*

We notice that we have proved 6.2.8 for a large class of sets of transformations over site graphs, not only for the valid sets of transformations. Indeed, we only need the restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings to distribute over the composition of weak embeddings (e.g. see 6.2.6) and the image of any embedding by a transformation in  $\mathbb{G}$  to be an embedding (e.g. see 6.2.8).

A consequence of Prop. 6.2.8 is that the structure of group action can be lifted to isomorphisms. This is expressed in the following property.

**Propriété 6.2.9 (action of transformations over isomorphisms)** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs. For any site graph  $F$ , the function which maps each pair  $(\sigma_F, \phi)$  such that  $\sigma_F$  is a transformation in  $\mathbb{G}_F$  and  $\phi$  is an isomorphism between two site graphs  $E$  and  $F'$  such that  $FF'$ , to the isomorphism  $\sigma_F." \phi$ , is a group action.*

**Applying transformations to automorphisms** The image of an automorphism by a transformation within a valid set of transformations over site graphs, is an isomorphism but not necessarily an automorphism. Yet, the number of automorphisms of a site graph  $E$  and the number of the site graphs that are both symmetric and isomorphic to the site graph  $E$  are closely related. Given two symmetric site graphs  $E$  and  $\sigma.E$ , the product between the number of automorphisms of  $E$  and the number of the site graphs that are both symmetric and isomorphic to  $E$ , and the product between the number of automorphisms of  $F$  and the number of the site graphs that are both symmetric and isomorphic to  $F$ , are equal, as stated in the following proposition:

**Propriété 6.2.10** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs. For any site graph  $E$  and for any permutation  $\sigma_E \in \mathbb{G}_E$ ,*

$$\frac{[E, E]}{[\sigma_E.E, \sigma_E.E]} = \frac{\text{card}([\sigma_E.E]_{iso} \cap [\sigma.E])}{\text{card}([E]_{iso} \cap [E])}.$$

**Transformations over isomorphic site graphs** We investigate the relationships between the sets of transformations that can be applied to two isomorphic site graphs.

The following property establishes that in a valid set of transformations, the sets to the transformations that can be applied to two isomorphic site graphs are in bijection.

**Propriété 6.2.11** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs. Let  $\phi$  be an isomorphism between two site graphs  $E$  and  $F$ . Then, the two following functions:*

$$\begin{cases} \mathbb{G}_F & \rightarrow & \mathbb{G}_E \\ \sigma_F & \mapsto & \phi.'\sigma_F \end{cases}$$

*and*

$$\begin{cases} \mathbb{G}_E & \rightarrow & \mathbb{G}_F \\ \sigma_E & \mapsto & \phi^{-1}.'\sigma_E \end{cases}$$

*form a pair of inverse functions.*

A first consequence of Prop. 6.2.11 is that we can compare the number of the transformations that can be applied to two isomorphic site graphs.

**Corolaire 6.2.1** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs and,  $E$  and  $F$  be two isomorphic graphs.*

*Then:*

$$\text{card}(\mathbb{G}_E) = \text{card}(\mathbb{G}_F)$$

Moreover, thanks to the proof of Prop. 6.2.11, we can establish that, in a valid set of transformations  $\mathbb{G}$ , restricting a transformation  $\sigma \in \mathbb{G}_E$  that can be applied to the image  $E$  of the inverse of an isomorphism  $\phi$ , into a transformation that can be applied to the domain  $E$  of the inverse of  $\phi$ , comes down to extend the transformation  $\sigma$  that can be applied to the domain  $E$  of the isomorphism  $\phi$ , into a transformation that can be applied to the image  $F$  of the isomorphism  $\phi$ . As a consequence, every isomorphism is  $\mathbb{G}$ -compatible.

**Corolaire 6.2.2** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs. Every isomorphism is  $\mathbb{G}$ -compatible.*

Within a valid set of transformations over site graphs, the set of the pairs of transformations that can be applied to a weak partial embedding forms a group, as established in the following property.

**Propriété 6.2.12** *Let  $\mathbb{G}$  be a set of valid transformations over site graphs and:*

$$h : L[f]D[g]R$$

*be a weak partial embedding.*

*The set of the pairs  $(\sigma_L, \sigma_R) \in \mathbb{G}_L \times \mathbb{G}_R$  such that  $f.\sigma_L = g.\sigma_R$  forms a group for the pairwise composition of transformations.*

*We denote this group by  $\mathbb{G}_h$ . The identity element of  $\mathbb{G}_h$  is the pair of permutation  $(\varepsilon_{\mathbb{G}_L}, \varepsilon_{\mathbb{G}_R})$ , where  $\varepsilon_{\mathbb{G}_L}$  is the identity element of the group  $\mathbb{G}_L$  and  $\varepsilon_{\mathbb{G}_R}$  is the identity element of the group  $\mathbb{G}_R$ .*

As an example, we investigate what are the groups of the pair of heterogeneous and homogeneous permutations of site that can be applied to the partial embedding that is given in Fig. 6.22(a).

**Exemple 6.2.12** *We take the same signature as in Exa. 6.2.1.*

*We consider the following three site graphs:*

$$\begin{aligned} D &:= A_1(x), A_2(z^-) \\ L &:= A_3(x), A_4(y, z^{A_5 \otimes z}), A_5(x, z^{A_4 \otimes z}) \\ R &:= A_6(x, z^{A_7 \otimes z}), A_7(y, z^{A_6 \otimes z}), A_8(y) \end{aligned}$$

*and the following two embeddings:*

$$\begin{aligned} f &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7] \end{aligned}$$

*We consider the partial embedding (eg. see Fig. 6.22(a)) :*

$$h : L[f]D[g]R.$$

*The group of the pairs of heterogeneous permutations  $\mathbb{G}_h^{\text{AGENT}}$  of sites that can be applied to the partial embedding  $h$ , is the set of the pairs  $(X, Y) \in \wp(\{3, 4, 5\}) \times \wp(\{6, 7, 8\})$  such that  $3 \in X$  if and only if  $6 \in Y$ , and  $4 \in X$  if and only if  $7 \in Y$ . Thus, there exist exactly  $2^4$  distinct heterogeneous permutations.*

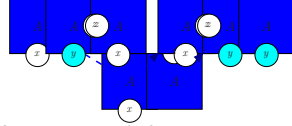
*For instance,*

1. *the image of the partial embedding  $h$  by the pairs of heterogeneous permutations of sites  $(\{3\}, \{6\})$  is given in Fig. 6.22(b) and is defined as the partial embedding:*

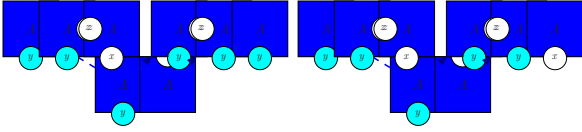
$$(\{3\}, \{6\}).h : L_{\{3,6\}}[f_{\{3,6\}}]D_{\{3,6\}}[g_{\{3,6\}}]R_{\{3,6\}}.$$

*where:*

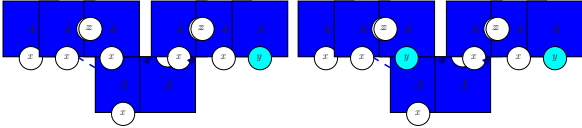
$$\begin{aligned} D_{\{3,6\}} &:= A_1(y), A_2(z^-) \\ L_{\{3,6\}} &:= A_3(y), A_4(y, z^{A_5 \otimes z}), A_5(x, z^{A_4 \otimes z}) \\ R_{\{3,6\}} &:= A_6(y, z^{A_7 \otimes z}), A_7(y, z^{A_6 \otimes z}), A_8(y) \\ f_{\{3,6\}} &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g_{\{3,6\}} &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7]; \end{aligned}$$



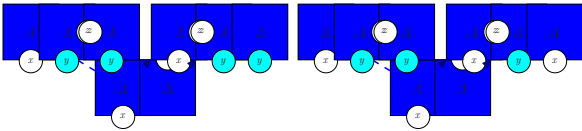
(a) The partial embedding  $h : A_3(x), A_4(y, z^{A_5 @ z}), A_5(x, z^{A_4 @ z}) \parallel A_1(x), A_2(z^-) \parallel A_6(x, z^{A_7 @ z}), A_7(y, z^{A_6 @ z}), A_8(y)$  with the pair of embeddings  $[(A, 1) \mapsto 3, (A, 2) \mapsto 4], [(A, 1) \mapsto 6, (A, 2) \mapsto 7]$ .



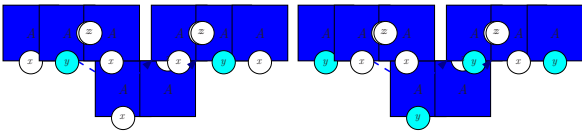
(b) The image of  $h$  by the pair of permutations of sites  $(\{3\}, \{6\})$ .



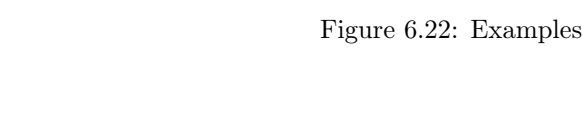
(c) The image of  $h$  by the pair of permutations of sites  $(\{3\}, \{6, 8\})$ .



(d) The image of  $h$  by the pair of permutations of sites  $(\{4\}, \{7\})$ .



(e) The image of  $h$  by the pair of permutations of sites  $(\{4, 5\}, \{7\})$ .



(f) The image of  $h$  by the pair of permutations of sites  $(\{5\}, \{8\})$ .



(g) The image of  $h$  by the pair of permutations of sites  $(\{3, 4, 5\}, \{6, 7\})$ .

Figure 6.22: Examples of symmetric partial embeddings.

2. the image of the partial embedding  $h$  by the pairs of heterogeneous permutations of sites  $(\{4\}, \{7\})$  is given in Fig. 6.22(d) and is defined as the partial embedding:

$$(\{4\}, \{7\})."h : L_{\{4,7\}}[f_{\{4,7\}}]D_{\{4,7\}}[g_{\{4,7\}}]R_{\{4,7\}}.$$

where:

$$\begin{aligned} D_{\{4,7\}} &:= A_1(x), A_2(z^-) \\ L_{\{4,7\}} &:= A_3(x), A_4(x, z^{A_5 \otimes z}), A_5(x, z^{A_4 \otimes z}) \\ R_{\{4,7\}} &:= A_6(x, z^{A_7 \otimes z}), A_7(x, z^{A_6 \otimes z}), A_8(y) \\ f_{\{4,7\}} &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g_{\{4,7\}} &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7]; \end{aligned}$$

3. the image of the partial embedding  $h$  by the pairs of heterogeneous permutations of sites  $(\{5\}, \emptyset)$  is given in Fig. 6.22(f) and is defined as the partial embedding:

$$(\{5\}, \emptyset)."h : L_{\{5\}}[f_{\{5\}}]D_{\{5\}}[g_{\{5\}}]R_{\{5\}}.$$

where:

$$\begin{aligned} D_{\{5\}} &:= A_1(x), A_2(z^-) \\ L_{\{5\}} &:= A_3(x), A_4(y, z^{A_5 \otimes z}), A_5(y, z^{A_4 \otimes z}) \\ R_{\{5\}} &:= A_6(x, z^{A_7 \otimes z}), A_7(y, z^{A_6 \otimes z}), A_8(y) \\ f_{\{5\}} &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g_{\{5\}} &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7]; \end{aligned}$$

4. the image of the partial embedding  $h$  by the pairs of heterogeneous permutations of sites  $(\emptyset, \{8\})$  is given in Fig. 6.22(h) and is defined as the partial embedding:

$$(\emptyset, \{8\})."h : L_{\{8\}}[f_{\{8\}}]D_{\{8\}}[g_{\{8\}}]R_{\{8\}}.$$

where:

$$\begin{aligned} D_{\{8\}} &:= A_1(x), A_2(z^-) \\ L_{\{8\}} &:= A_3(x), A_4(y, z^{A_5 \otimes z}), A_5(x, z^{A_4 \otimes z}) \\ R_{\{8\}} &:= A_6(x, z^{A_7 \otimes z}), A_7(y, z^{A_6 \otimes z}), A_8(x) \\ f_{\{8\}} &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g_{\{8\}} &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7]; \end{aligned}$$

We say that the four following pairs of heterogeneous permutations  $(\{3\}, \{6\})$ ,  $(\{4\}, \{7\})$ ,  $(\{5\}, \emptyset)$ , and  $(\emptyset, \{8\})$  elementary, because they are minimal for the pairwise extension of the inclusion order  $\subseteq$  among the pair of heterogeneous permutations  $\mathbb{G}_h^{\text{AGENT}} \setminus \{(\emptyset, \emptyset)\}$  which are not the identity element. We notice that any pair of heterogeneous permutations in  $\mathbb{G}_h$  can be obtained as a composition of these elementary pairs of heterogeneous permutations.

**Example 6.2.13** With the same signature and the same partial embedding  $h$  as in Exa. 6.2.12.

The group of the pairs of homogeneous permutations  $\mathbb{G}_h^{\text{SPECIES}}$  of sites that can be applied to the partial embedding  $h$ , is the set of the pairs  $(X, Y) \in \wp(\{3, 4, 5\}) \times \wp(\{6, 7, 8\})$  such that  $3 \in X$  if and only if  $4 \in X$  if and only if  $5 \in X$  if and only if  $6 \in Y$  if and only if  $7 \in Y$ . Thus, there exist exactly  $2^2$  distinct homogenous permutations: the permutation  $(\emptyset, \emptyset)$  (the image of  $h$  is given in Fig. 6.22(a)), the permutation  $(\emptyset, \{8\})$  (the image of  $h$  is given in Fig. 6.22(h)), the permutation  $(\{3, 4, 5\}, \{6, 7\})$  (the image of  $h$  is given in Fig. 6.22(i)), and the permutation  $(\{3, 4, 5\}, \{6, 7, 8\})$ .

We say that the two following pairs of homogeneous permutations  $(\emptyset, \{8\})$  and  $(\{3, 4, 5\}, \{6, 7\})$  are elementary (because they are minimal for the pairwise extension of the inclusion order  $\subseteq$  among the pair of homogeneous permutations  $\mathbb{G}_h^{\text{SPECIES}} \setminus \{(\emptyset, \emptyset)\}$  which are not the identity element). We also notice that any of the two others pairs of homogeneous permutations in  $\mathbb{G}_h^{\text{SPECIES}}$  can be obtained as a composition of these elementary pairs of homogeneous permutations.

Then, the structure of group action can be lifted to weak partial embeddings, as expressed as follows.

**Definition 6.2.16 (pairs of symmetric weak partial embeddings)** Let  $\mathbb{G}$  be a valid set of permutations. We say that two weak partial embeddings  $h$  and  $h'$  are  $\mathbb{G}$ -symmetric to one another with respect to the set of transformations  $\mathbb{G}$  if and only if there exists  $(\sigma_L, \sigma_R) \in \mathbb{G}_h$  such that  $h' = \sigma_L \sigma_R h$ .

In such a case, we write  $hh'$ .

**Propriété 6.2.13** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs. then the relation  $\sim$  is an equivalence relation over weak partial embeddings.*

**Propriété 6.2.14** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs, and  $h$  and  $h'$  be two weak partial embeddings that are  $\mathbb{G}$ -symetric to one another.*

*Then  $\mathbb{G}_h$  and  $\mathbb{G}_{h'}$  are equal.*

**Propriété 6.2.15 (action of pairs of transformations over weak partial embeddings)** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs and  $h$  be a weak partial embeddings.*

*Then the function which maps any pair  $((\sigma_L, \sigma_R), h')$  such that  $(\sigma_L, \sigma_R) \in \mathbb{G}_h$  is a pair of transformations that can be applied to the weak partial embedding  $h$ , and  $h'$  is a weak partial embedding which is  $\mathbb{G}$ -symetric to the weak partial embedding  $h$ , into the weak partial embedding  $\sigma_L \sigma_R h$ , is a group action.*

We keep on our examples, by providing some samples of pairs of symetric partial embeddings and some examples of pairs of non symetric partial embeddings.

**Exemple 6.2.14** *We take the same signature as in Exa. 6.2.1.*

*We consider the two following partial embeddings:*

$$h : L[f]D[g]R$$

and

$$h' : L'[f']D'[g']R'.$$

with:

$$\begin{aligned} D &:= A_1(x), A_2(z^-) \\ L &:= A_3(x), A_4(y, z^{A_5 \otimes z}), A_5(x, z^{A_4 \otimes z}) \\ R &:= A_6(x, z^{A_7 \otimes z}), A_7(y, z^{A_6 \otimes z}), A_8(y) \\ f &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7] \end{aligned}$$

and:

$$\begin{aligned} D' &:= A_1(x), A_2(z^-) \\ L' &:= A_3(x), A_4(x, z^{A_5 \otimes z}), A_5(y, z^{A_4 \otimes z}) \\ R' &:= A_6(x, z^{A_7 \otimes z}), A_7(x, z^{A_6 \otimes z}), A_8(y) \\ f' &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g' &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7] \end{aligned}$$

A graphical representation of the partial embedding  $h$  is given in Fig. 6.22(a) and a graphical representation of the partial embedding  $h'$  is given in Fig. 6.22(e).

We can check that:  $h[\mathbb{G}^{\text{AGENT}}]h'$ , but  $h[\mathbb{G}^{\text{SPECIES}}]h'$ . Indeed the sites  $x\hat{A}$  and  $y\hat{A}$  has been swapped in the agent with the label 7 of the right hand side of the partial embedding, but not in the agent with the label 8, although these two agents belong to a same connected component.

**Exemple 6.2.15** *We take the same signature as in Exa. 6.2.1.*

*We consider the two following partial embeddings:*

$$h : L[f]D[g]R$$

and

$$h'' : L''[f'']D''[g'']R''.$$

with:

$$\begin{aligned} D &:= A_1(x), A_2(z^-) \\ L &:= A_3(x), A_4(y, z^{A_5 \otimes z}), A_5(x, z^{A_4 \otimes z}) \\ R &:= A_6(x, z^{A_7 \otimes z}), A_7(y, z^{A_6 \otimes z}), A_8(y) \\ f &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7] \end{aligned}$$

and:

$$\begin{aligned} D'' &:= A_1(y), A_2(z^-) \\ L'' &:= A_3(y), A_4(x, z^{A_5 \otimes z}), A_5(y, z^{A_4 \otimes z}) \\ R'' &:= A_6(y, z^{A_7 \otimes z}), A_7(x, z^{A_6 \otimes z}), A_8(y) \\ f'' &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g'' &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7]. \end{aligned}$$

A graphical representation of the partial embedding  $h$  is given in Fig. 6.22(a) and a graphical representation of the partial embedding  $h''$  is given in Fig. 6.22(i).

We can check that:  $h[\mathbb{G}^{\text{AGENT}}]h'$ . As a consequence, we also have:  $h[\mathbb{G}^{\text{SPECIES}}]h'$ .

**Pushing a transformation forwards or backwards along a partial embedding.** It is not always possible to push a transformation over site graphs forwards or backwards along a partial embedding.

Let us explain what happen when one wants to push a transformation that can be applied to the left hand side of a partial embedding forwards along this partial embedding. Let

$$h : L[f]D[g]R$$

be a partial embedding and let  $\sigma_L$  be a transformation that can be applied with the left hand side  $L$  of a partial embedding  $h$ . The transformation  $\sigma_L$  can be restricted to the domain of the left embedding  $f$  of the partial embedding  $h$ . This way, we get the transformation  $f.\sigma_L$  which can be applied with the domain  $D$  of the partial embedding  $h$ . Then, either there exists a transformation  $\sigma_R$  such that  $g.\sigma_R = f.\sigma_L$ , or not. Moreover, when such a transformation  $\sigma_R$  exists (which is always the case if the embedding  $g$  is compatible), it is usually not unique, because the choice of the transformations that can be applied to the agents of  $R$  which are not documented in the domain  $D$  of the partial embedding  $h$  is free.

**Example 6.2.16** We take the same signature as in Exa. 6.2.1.

We consider the following partial embedding (e.g. see Fig. 6.22(a)):

$$h : L[f]D[g]R$$

with:

$$\begin{aligned} D &:= A_1(x), A_2(z^-) \\ L &:= A_3(x), A_4(y, z^{A_5 \otimes z}), A_5(x, z^{A_4 \otimes z}) \\ R &:= A_6(x, z^{A_7 \otimes z}), A_7(y, z^{A_6 \otimes z}), A_8(y) \\ f &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7], \end{aligned}$$

and the heterogeneous permutation of sites  $\{3\} \in \mathbb{G}_L^{\text{AGENT}}$  that can be applied to the site graph  $L$ . The restriction  $f.\{3\}$  of this permutation to the domain of the partial embedding is the heterogeneous permutation of sites  $\{1\} \in \mathbb{G}_D^{\text{AGENT}}$ . The heterogeneous permutation of sites can be extended to the image of  $g$  in two ways depending whether the sites  $x\hat{A}$  and  $y\hat{A}$  of the agent with label 8 are swapped, or not, in the site graph  $R$ . Thus we get either the heterogeneous permutation of sites  $\{6\} \in \mathbb{G}_R^{\text{AGENT}}$  (e.g. see Fig. 6.22(b)), or the heterogeneous permutation of sites  $\{6, 8\} \in \mathbb{G}_R^{\text{AGENT}}$  (e.g. see Fig. 6.22(c)).

**Example 6.2.17** We take the same signature as in Exa. 6.2.1 and the same partial embedding as in Exa. 6.2.16.

We notice that the permutation of sites  $\{3\} \in \mathbb{G}_L^{\text{AGENT}}$  is also homogeneous (i.e.  $\{3\} \in \mathbb{G}_L^{\text{SPECIES}}$ ). Yet there is no way to extend the permutation  $\{1\} \in \mathbb{G}_D^{\text{SPECIES}}$  along the embedding  $g$  into a homogeneous permutation of sites that can be applied to the site graph  $R$ , since the agents with the labels 3 and 4 have been bound by the partial embedding, and they now belong to a same connected component. Thus the homogeneous permutation of sites  $\{3\} \in \mathbb{G}_L^{\text{SPECIES}}$  cannot be pushed forward the partial embedding  $h$ .

The same phenomena may be encountered when wanting to push a transformation over site graphs backwards along a partial embedding

**Example 6.2.18** We take the same signature as in Exa. 6.2.1 and the same partial embedding as in Exa. 6.2.16.

We consider the heterogeneous permutation of sites  $\{7\} \in \mathbb{G}_R^{\text{AGENT}}$  that can be applied to the site graph  $R$ . The restriction  $g.\{7\}$  of this permutation to the domain of the partial embedding is the heterogeneous permutation of sites  $\{2\} \in \mathbb{G}_D^{\text{AGENT}}$ . The heterogeneous permutation of sites can be extended to the image of  $f$  in two ways



depending whether the sites  $x\hat{A}$  and  $y\hat{A}$  of the agent with label 5 are swapped, or not, in the site graph  $L$ . Thus we get either the heterogeneous permutation of sites  $\{4\} \in \mathbb{G}_R^{\text{AGENT}}$  (e.g. see Fig. 6.22(d)), or the heterogeneous permutation of sites  $\{4, 7\} \in \mathbb{G}_R^{\text{AGENT}}$  (e.g. see Fig. 6.22(e)).

**Exemple 6.2.19** We take the same signature as in Exa. 6.2.1 and the same partial embedding as in Exa. 6.2.16. Any homogeneous permutation of sites in  $\mathbb{G}_R^{\text{SPECIES}}$  can be pushed backward the partial embedding  $h$ . Indeed, even if the agents with the labels 4 and 5 are bound together in the left hand side  $L$  of the partial embedding  $h$ , it is not constraining since the agent with the label 5 is not associated with an agent in the right hand side  $R$  of the partial embedding  $h$ .

**Exemple 6.2.20** Examples of transformations over site graphs that cannot be pushed backwards along a partial embedding can be obtained by taking examples of transformations over site graphs that cannot be pushed forwards along a partial embedding, and by reverting the partial embedding (by swapping the left and the right hand sides and by swapping the left and the right embeddings).

We say that a partial embedding is forward (resp. backward) compatible with a set of transformations if any of the transformations that can be applied to its left hand side (resp. right hand side) can be pushed forwards (resp. backwards) this partial embedding, as stated in the following definitions.

**Definition 6.2.17** Let  $\mathbb{G}$  be a valid set of transformations over site graphs and  $h$  be a partial embedding between two site graphs  $L$  and  $R$ . We say that  $h$  is  $\mathbb{G}$ -forward compatible if and only if for any permutation  $\sigma_L \in \mathbb{G}_L$ , there exists a transformation  $\sigma_R \in \mathbb{G}_R$  such that  $(\sigma_L, \sigma_R) \in \mathbb{G}_h$ .

**Definition 6.2.18** Let  $\mathbb{G}$  be a set of transformations over site graphs and  $h$  be a partial embedding between two site graphs  $L$  and  $R$ . We say that  $h$  is  $\mathbb{G}$ -backward compatible if and only if for any permutation  $\sigma_R \in \mathbb{G}_R$ , there exists a transformation  $\sigma_L \in \mathbb{G}_L$  such that  $(\sigma_L, \sigma_R) \in \mathbb{G}_h$ .

**Propriété 6.2.16** Let  $\mathbb{G}$  be a valid set of transformations over site graphs and

$$h : L[f]D[g]R$$

be a partial embedding between two site graphs  $L$  and  $R$ . We consider the partial embedding  $h'$  that is defined as follows:

$$h' : R[g]D[f]L.$$

Then, the two following equivalences are satisfied:

1. the partial embedding  $h$  is  $\mathbb{G}$ -forward compatible, if and only if, the partial embedding  $h'$  is  $\mathbb{G}$ -backward compatible.
2. the partial embedding  $h$  is  $\mathbb{G}$ -backward compatible, if and only if, the partial embedding  $h'$  is  $\mathbb{G}$ -forward compatible.

**Propriété 6.2.17** Let  $\mathbb{G}$  be a valid set of transformations over site graphs and

$$h : L[f]D[g]R$$

be a partial embedding between two site graphs  $L$  and  $R$ .

The following implications hold:

1. If the right embedding  $g$  is  $\mathbb{G}$ -compatible, then the partial embedding  $h$  is  $\mathbb{G}$ -forward compatible.
2. If the left embedding  $f$  is  $\mathbb{G}$ -compatible, then the partial embedding  $h$  is  $\mathbb{G}$ -backward compatible.

**Exemple 6.2.21** Any partial embedding is  $\mathbb{G}^{\text{AGENT}}$ -forwards and  $\mathbb{G}^{\text{AGENT}}$ -backwards compatible.

**Exemple 6.2.22** We take the same signature as in Exa. 6.2.1 and the same partial embedding as in Exa. 6.2.16. The partial embedding  $h$  is  $\mathbb{G}^{\text{SPECIES}}$ -backwards compatible, but not  $\mathbb{G}^{\text{SPECIES}}$ -forwards compatible.

The set of the transformations that can be applied with the right hand side of a partial embedding and which are compatible with a given transformation of the left hand side of this partial embedding, enjoys a nice algebraic structure, as described in the following proposition.

**Propriété 6.2.18** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs,*

$$h : L[f]D[g]R$$

*be a partial embedding between two site graphs  $L$  and  $R$ , and  $\sigma_L \in \mathbb{G}_L$  be a transformation that can be applied to the left hand side  $L$  of the partial embedding  $h$ .*

*The set:*

$$\{\sigma_R \in \mathbb{G}_R \mid (\sigma_L, \sigma_R) \in \mathbb{G}_h\}$$

*is either empty, or of the form:*

$$\{\sigma_R \circ_R \sigma'_R \mid \sigma'_R \in \mathbb{G}_R \text{ tel que } g.\sigma'_R = \varepsilon_{\mathbb{G}_D}\},$$

*for a given transformation  $\sigma_R \in \mathbb{G}_R$ .*

The reasoning is the same when trying to pull a transformation that can be applied to the right hand side of a partial embedding backwards, so as to get a transformation that can be applied to the right hand side of this partial embedding.

**Propriété 6.2.19** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs,*

$$h : L[f]D[g]R$$

*be a partial embedding between two site graphs  $L$  and  $R$ , and  $\sigma_L \in \mathbb{G}_L$  be a transformation that can be applied to the left hand side  $L$  of the partial embedding  $h$ .*

*The set:*

$$\{\sigma_L \in \mathbb{G}_L \mid (\sigma_L, \sigma_R) \in \mathbb{G}_h\}$$

*is either empty, or of the form:*

$$\{\sigma_L \circ_L \sigma'_L \mid \sigma'_L \in \mathbb{G}_L \text{ tel que } f.\sigma'_L = \varepsilon_{\mathbb{G}_D}\},$$

*for a given transformation  $\sigma_L \in \mathbb{G}_L$ .*

**Preuve 6.2.1** *We get Prop. 6.2.18 by applying Thm. 6.2.1.(3) with the embedding  $f$  between the two site graphs  $D$  and  $L$ , and with the transformation  $g.\sigma_R \in \mathbb{G}_D$ .*

**Decomposing the group of the transformations that can be applied to a partial embedding** In a valid set of transformations, a partial embedding  $h$  between two site graphs  $L$  and  $R$ , and with domain  $D$ , induces a morphism between the group of the pairs of transformations  $\mathbb{G}_h$  that can be applied to the partial embedding  $h$ , and the group of the transformations  $\mathbb{G}_D$  that can be applied to the domain of the partial embedding  $h$ . Thanks to this property, we can properly decompose the group of the transformations that can be applied to a partial embedding  $h$  into three subgroups: the groups of the transformations that can be applied to the domain of the partial embedding  $h$ , the group of the transformations that can be applied to the left hand side  $L$  of the partial embedding  $h$  that becomes the identity transformation when restricted to the domain  $D$  of the partial embedding  $h$ , and the group of the transformations that can be applied to the right hand side  $R$  of the partial embedding  $h$  that becomes the identity transformation when restricted to the domain  $D$  of the partial embedding  $h$ ,

**Propriété 6.2.20** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs and*

$$h : L[f]D[g]R$$

*be a partial embedding between two site graphs  $L$  and  $R$ .*

*The following function:*

$$\begin{cases} \mathbb{G}_h \rightarrow \mathbb{G}_D \\ (\sigma_L, \sigma_R) \mapsto \sigma_L \sigma_R h \end{cases}$$

*is a morphism between the groups  $\mathbb{G}_h$  and  $\mathbb{G}_D$ .*

**Preuve 6.2.1** Let  $\mathbb{G}$  be a valid set of transformations over site graphs and:

$$h : L[f]D[g]R$$

be a partial embedding between two site graphs  $L$  and  $R$ .

Let  $(\sigma_L, \sigma_R)$  and  $(\sigma'_L, \sigma'_R)$  be two elements of  $\mathbb{G}_h$ .

Thanks to Defs. 6.2.10 and ??, we have:

$$\begin{aligned} h.'((\sigma_L, \sigma_R) \circ_h (\sigma'_L, \sigma'_R)) &= \sigma_L \circ_L \sigma'_L \sigma_R \circ_R \sigma'_R h \\ h.'((\sigma_L, \sigma_R) \circ_h (\sigma'_L, \sigma'_R)) &= f.'(\sigma_L \circ_L \sigma'_L) \\ h.'((\sigma_L, \sigma_R) \circ_h (\sigma'_L, \sigma'_R)) &= ((f.'\sigma_L) \circ_D (f.'\sigma'_L)) \\ h.'((\sigma_L, \sigma_R) \circ_h (\sigma'_L, \sigma'_R)) &= (\sigma_L \sigma_R h) \circ_D (\sigma'_L \sigma'_R h). \end{aligned}$$

**Théorème 6.2.2** Let  $\mathbb{G}$  be a valid set of transformations over site graphs and let:

$$h : L[f]D[g]R$$

be a partial embedding.

Both following assertions are satisfied:

1. The set:

$$\{\sigma_L \sigma_R h \mid (\sigma_L, \sigma_R) \in \mathbb{G}_h\}$$

and the set:

$$\mathbb{G}_h \setminus \{(\sigma_L, \sigma_R) \in \mathbb{G}_h \mid \sigma_L \sigma_R h = \varepsilon_D\}$$

are two isomorphic groups.

2. The cardinal of the group  $\mathbb{G}_h$  is equal to the product between:

(a) the cardinal of the group:

$$\{\sigma_L \sigma_R h \mid (\sigma_L, \sigma_R) \in \mathbb{G}_h\},$$

(b) the cardinal of the group:

$$\{\sigma_L \in \mathbb{G}_L \mid f.'\sigma_L = \varepsilon_D\},$$

(c) the cardinal of the group:

$$\{\sigma_R \in \mathbb{G}_R \mid g.'\sigma_R = \varepsilon_D\}.$$

**Preuve 6.2.2** Let  $\mathbb{G}$  be a valid set of transformations over site graphs and let:

$$h : L[f]D[g]R$$

be a partial embedding.

According to Prop. 6.2.20, the function between the group  $\mathbb{G}_h$  and the group  $\mathbb{G}_D$  mapping each pair of transformations  $\sigma_h \in \mathbb{G}_h$  into the transformation  $h.'\sigma_h \in \mathbb{G}_D$ , is a morphism. The first assertion follows from the first isomorphism theorem (Thm. ??).

The second assertion is proved in two steps. From Cor. ??, we get that the cardinal of the group  $\mathbb{G}_h$  is equal to the product between:

1. the cardinal of the group:

$$\{\sigma_L \sigma_R h \mid (\sigma_L, \sigma_R) \in \mathbb{G}_h\},$$

2. the cardinal of the group:

$$\{\sigma_h \in \mathbb{G}_h \mid h.'\sigma_h = \varepsilon_D\}.$$

Then, we have:

$$\begin{aligned} & \{\sigma_h \in \mathbb{G}_h \mid h.'\sigma_h = \varepsilon_D\} \\ = & \left\{ (\sigma_L, \sigma_R) \in \mathbb{G}_L \times \mathbb{G}_R \left| \begin{array}{l} f.'\sigma_L = g.'\sigma_R \\ \sigma_L \sigma_R h = \varepsilon_D \end{array} \right. \right\} \\ = & \left\{ (\sigma_L, \sigma_R) \in \mathbb{G}_L \times \mathbb{G}_R \left| \begin{array}{l} f.'\sigma_L = g.'\sigma_R \\ f.'\sigma_L = \varepsilon_D \end{array} \right. \right\} \\ = & \left\{ (\sigma_L, \sigma_R) \in \mathbb{G}_L \times \mathbb{G}_R \left| \begin{array}{l} f.'\sigma_L = \varepsilon_D \\ g.'\sigma_R = \varepsilon_D \end{array} \right. \right\} \\ = & \{\sigma_L \in \mathbb{G}_L \mid f.'\sigma_L = \varepsilon_D\} \times \{\sigma_R \in \mathbb{G}_R \mid g.'\sigma_R = \varepsilon_D\}. \end{aligned}$$

In Thm. 6.2.2, the group  $\{\sigma_L \sigma_R f \mid (\sigma_L, \sigma_R) \in \mathbb{G}_h\}$  denotes the set of the transformations that can be applied to the domain  $D$  of the partial embedding  $h$  and that can be extended into a pair of transformations that can be applied respectively to the left hand side  $L$  and the right hand side  $R$  of the partial embedding  $h$ . Then, the group  $\{\sigma_L \in \mathbb{G}_L \mid f.\sigma_L = \varepsilon_D\}$  is the kernel of morphism  $h_f$  that is induced by the embedding  $f$  and denotes the set of transformations over the left hand side  $L$  of the partial embedding  $h$  that do not modify the domain  $D$  of the partial embedding  $h$  (when we restrict this transformation to the domain  $D$  of the embedding  $f$ ). The same way, the group  $\{\sigma_R \in \mathbb{G}_R \mid g.\sigma_R = \varepsilon_D\}$  is the kernel of morphism  $h_g$  that is induced by the embedding  $g$  and denotes the set of transformations over the right hand side  $R$  of the partial embedding  $h$  that do not modify the domain  $D$  of the partial embedding  $h$  (when we restrict this transformation to the domain  $D$  of the embedding  $g$ ).

We illustrate the use of Thm. 6.2.2 by decomposing the group of the pairs of heterogeneous permutations of sites and the group of the pairs of homogeneous permutations of sites that can be applied to the partial embedding that is given in Fig. 6.22(a), in the following two examples.

**Example 6.2.23** We take the same signature as in Exa. 6.2.1 and we consider the following partial embedding (e.g. see Fig. 6.22(a)):

$$h : L[f]D[g]R$$

with:

$$\begin{aligned} D &:= A_1(x), A_2(z^-) \\ L &:= A_3(x), A_4(y, z^{A_5 \otimes z}), A_5(x, z^{A_4 \otimes z}) \\ R &:= A_6(x, z^{A_7 \otimes z}), A_7(y, z^{A_6 \otimes z}), A_8(y) \\ f &:= [(A, 1) \mapsto 3, (A, 2) \mapsto 4] \\ g &:= [(A, 1) \mapsto 6, (A, 2) \mapsto 7]. \end{aligned}$$

The group of the pairs of heterogeneous permutations  $\mathbb{G}_h^{\text{AGENT}}$  of sites that can be applied to the partial embedding  $h$  can be decomposed into three groups.

1. The group of the heterogeneous permutations of sites that can be applied to the domain  $D$  of the partial embedding  $h$ , and extended both into an heterogeneous permutation of sites that can be applied to the left hand side  $L$  of the partial embedding  $h$  and into an heterogeneous permutation of sites that can be applied to the right hand side  $R$  of the partial embedding  $h$ , that is to say the group  $\wp(\{1, 2\})$  that contains exactly four elements.
2. The group of the heterogeneous permutations of sites that can be applied to the left hand side  $L$  of the partial embedding  $h$ , and that are restricted to the identity permutation when restricted to the domain  $D$  of the partial embedding  $h$ , that is to say the set  $\{\emptyset, \{5\}\}$  that contains exactly two elements.
3. The group of the heterogeneous permutations of sites that can be applied to the right hand side  $R$  of the partial embedding  $h$ , and that are restricted to the identity permutation when restricted to the domain  $D$  of the partial embedding  $h$ , that is to say the set  $\{\emptyset, \{8\}\}$  that contains exactly two elements.

We get back the  $2^4$  elements of the group  $\mathbb{G}_h^{\text{AGENT}}$ .

**Example 6.2.24** We take the same signature as in Exa. 6.2.1 and the same partial embedding as in Exa. 6.2.23.

The group of the pairs of homogeneous permutations  $\mathbb{G}_h^{\text{SPECIES}}$  of sites that can be applied to the partial embedding  $h$  can be decomposed into three groups.

1. The group of the homogeneous permutations of sites that can be applied to the domain  $D$  of the partial embedding  $h$ , and extended both into an heterogeneous permutation of sites that can be applied to the left hand side  $L$  of the partial embedding  $h$  and into an heterogeneous permutation of sites that can be applied to the right hand side  $R$  of the partial embedding  $h$ . Since the agents with label 6 and 7 are bound together in  $R$ , such a permutation should swap the sites  $x$  and  $y$  in the agent with the label 1 if and only if it swaps also the sites  $x$  and  $y$  of the agent with the label 2 in the site graph  $D$ . Thus this group is the group  $\{\emptyset, \{1, 2\}\}$  and it contains exactly two elements.
2. The group of the homogeneous permutations of sites that can be applied to the left hand side  $L$  of the partial embedding  $h$ , and that gives the identity permutation when restricted to the domain  $D$  of the partial embedding  $h$ . There is only one such permutation of sites (the identity transformation  $\emptyset$ ). Indeed, if it were containing another permutation, this permutation would have to swap the two sites  $x$  and  $y$  in

the agent with the label 5 in the site graph  $L$ , then since the agents with the labels 4 and 5 are bound together in the site graph  $L$ , such a permutation would have to swap also the sites  $x$  and  $y$  of the agent 4 as well, which is not possible, since its restriction to the domain  $D$  of the partial embedding would then not be the identity permutation of sites. Thus, this group is the group  $\{\emptyset\}$  and it has only one element.

3. The group of the homogeneous permutations of sites that can be applied to the right hand side  $R$  of the partial embedding  $h$ , and that gives the identity permutation when restricted to the domain  $D$  of the partial embedding  $h$ , that is to say the set  $\{\emptyset, \{8\}\}$  that contains exactly two elements.

We get back the 4 elements of the group  $\mathbb{G}_h^{\text{SPECIES}}$ .

**Compatibility between the application of pairs of transformations and the composition of weak partial embeddings** So as to check the coherence of our algebraic construction, we check some sanity properties about the application of pairs of transformations to weak partial embeddings.

The domain of a composition of weak partial embeddings on the first hand, and the domain of rules in the second hand are defined only modulo isomorphism. Thus, it is important to check to the image of a weak partial embedding by a pair of transformations over site graphs, is independent from the choice, modulo isomorphism, of the domain of this weak partial embedding. This sanity property is formalised in the following proposition.

**Propriété 6.2.21** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs. Let  $L$ ,  $R$ , and  $D$  be three site graphs,  $f$  be a weak embedding between the site graphs  $D$  and  $L$ , and  $g$  be a weak embedding between the site graph  $D$  and  $R$ . We denote by  $h$  the following weak partial embedding:*

$$h : L[f]D[g]R.$$

*Let  $\phi$  an isomorphism between a site graph  $D'$  and the site graph  $D$ . We denote by  $h'$  the following weak partial embedding:*

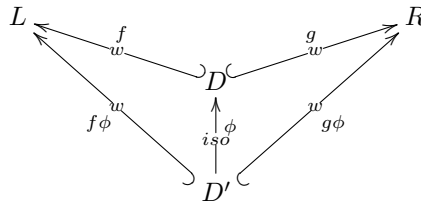
$$h' : L[f\phi]D'[g\phi]R.$$

*Under these assumptions, both following properties are satisfied:*

1.  $\mathbb{G}_h = \mathbb{G}_{h'}$ ;
2. *For any  $(\sigma_L, \sigma_R) \in \mathbb{G}_h$ , there exists an isomorphism  $\psi$  between a site graph  $D''$  and the site graph  $(f.\sigma_L).D$  such that the image  $\sigma_L\sigma_R h'$  of the weak embedding  $h'$  by the pair of transformations  $(\sigma_L, \sigma_R)$  is equal to the following weak partial embedding:*

$$\sigma_L\sigma_R h' : \sigma_L.L[(\sigma_L.''f)\psi]D''[(\sigma_R.''g)\psi]\sigma_R.R.$$

**Preuve 6.2.3** *Let us take the notations of Prop. 6.2.21.*



1. (a) Let  $(\sigma_L, \sigma_R) \in \mathbb{G}_h$ .

We have:

$$\begin{aligned} (f\phi).\sigma_L &= \phi.(f.\sigma_L) && \text{(by Def. 6.2.6.(3))} \\ (f\phi).\sigma_L &= \phi.(g.\sigma_R) && \text{(by Def. ??)} \\ (f\phi).\sigma_L &= (g\phi).\sigma_R && \text{(by Def. 6.2.6.(3))} \end{aligned}$$

By Def. ??, we get that:  $(\sigma_L, \sigma_R) \in \mathbb{G}_{h'}$ .

Thus,  $\mathbb{G}_r \subseteq \mathbb{G}_{h'}$ .

- (b) We get that  $\mathbb{G}_h \supseteq \mathbb{G}_{h'}$  by replacing  $h$  with  $h'$ ,  $f$  with  $f\phi$ ,  $g$  with  $g\phi$ , and  $\phi$  with  $\phi^{-1}$  in the proof of the direct inclusion.

2. Let us apply the pair of transformations  $(\sigma_L, \sigma_R) \in \mathbb{G}_h$  to the weak partial embedding  $h$ .

By Def. 6.2.6.(4), we have:

$$\sigma_L.''(f\phi) = (\sigma_L.''f)((f.'\sigma_L)''\phi)$$

and:

$$\sigma_R.''(g\phi) = (\sigma_R.''g)((g.'\sigma_R)''\phi)$$

Then, by Def. 6.2.12, we get that:

$$\sigma_R.''(g\phi) = (\sigma_L.''f)((f.'\sigma_L)''\phi)$$

By Def. 6.2.1, the image  $(f.'\sigma_L)''\phi$  of the isomorphism  $\phi$  by the transformation  $f.'\sigma_L$  is a partial embedding between the site graph  $(\phi.'(f.'\sigma_L)).D'$  and the site graph  $(f.'\sigma_L).D$ . Moreover, by Prop. 6.2.8, this partial embedding is an isomorphism.

Let us denote by  $D''$  the image  $(\phi.'(f.'\sigma_L)).D'$  of the site graph  $D'$  by the transformation  $\phi.'(f.'\sigma_L)$ , and  $\psi$  the image  $(f.'\sigma_L)''\phi$  of the isomorphism  $\phi$  by the transformation  $f.'\sigma_L$ . We get the following commutating diagram:

$$\begin{array}{ccccc}
 \sigma_L.L & & \xleftarrow{\sigma_L.''f} & & \sigma_R.R \\
 & \searrow \scriptstyle w & & \nwarrow \scriptstyle w & \\
 & & (f.'\sigma_L).D & & \\
 & \swarrow \scriptstyle w & \uparrow \scriptstyle \psi & \searrow \scriptstyle w & \\
 \sigma_L.''(f\phi) & & D'' & & \sigma_R.''(g\phi)
 \end{array}$$

which concludes the proof.

In order to define the composition between weak partial embeddings and partial embeddings, we had to consider each partial embedding as a weak partial embedding in which the left partial embedding is an identity embedding. So as to check the coherence of this convention, we check that whenever a weak partial embedding is indeed a weak embedding (because its left weak embedding is an identity embedding), applying a pair of transformations to the weak partial embedding comes down to apply a transformation to the corresponding weak embedding.

**Propriété 6.2.22** Let  $\mathbb{G}$  be a valid set of transformations over site graphs,  $E$  and  $F$  be two site graphs,  $f$  be a weak embedding between the site graph  $E$  and the site graph  $F$  and  $\sigma_F$  be a transformation that can be applied to the site graph  $F$ .

Both following assertions hold:

1.  $(f.'\sigma_F, \sigma_F) \in \mathbb{G}_{(E,f)}$ ;
2.  $(f.'\sigma_F)\sigma_F(E, f) = ((f.'\sigma_F).E, \sigma_F.''f)$ .

**Preuve 6.2.4** Let  $\mathbb{G}$  be a valid set of transformations over site graphs,  $E$  and  $F$  be two site graphs,  $f$  be a weak embedding between the site graph  $E$  and the site graph  $F$  and  $\sigma_F$  be a transformation that can be applied to the site graph  $F$ .

1. We have:

$$E(f.'\sigma_F) = f.'\sigma_F \quad (\text{by Def. 6.2.6.(1)})$$

Thus, by Prop. 6.2.12, we have:  $(f.'\sigma_F, \sigma_F) \in \mathbb{G}_{(E,f)}$ .

2. We have:

$$\begin{aligned}
 & (f.'\sigma_F)\sigma_F(E, f) \\
 &= ((f.'\sigma_F)''E, \sigma_F.''f) \quad (\text{by Def. ??}) \\
 &= ((f.'\sigma_F), \sigma_F.''f) \quad (\text{by Def. 6.2.6.(2)})
 \end{aligned}$$

**Propriété 6.2.23** Let  $\mathbb{G}$  be a valid set of transformations over site graphs,  $G, H, K$  be three site graphs, and  $h$  and  $h'$  be two partial weak embedding respectively between the site graphs  $G$  and  $H$ , and between the site graphs  $H$  and  $K$ . Let  $\sigma_G \in \mathbb{G}_G$ ,  $\sigma_H \in \mathbb{G}_H$ ,  $\sigma_K \in \mathbb{G}_K$  be three transformations that can be applied respectively to the site graphs  $G, H$ , and  $K$ . We assume that  $(\sigma_G, \sigma_H) \in \mathbb{G}_h$  and  $(\sigma_H, \sigma_K) \in \mathbb{G}_{h'}$ .

Under this assumption, both following assertions hold:

1.  $(\sigma_G, \sigma_K) \in \mathbb{G}_{(h'h)}$ ;
2.  $\sigma_G \sigma_K (h'h) = (\sigma_H \sigma_K h')(\sigma_G \sigma_H h)$ .

**Preuve 6.2.5** Let  $\mathbb{G}$  be a valid set of transformations over site graphs,  $G, H, K$  be three site graphs, and  $h$  and  $h'$  be two partial weak embedding respectively between the site graphs  $G$  and  $H$ , and between the site graphs  $H$  and  $K$ . Let  $\sigma_G \in \mathbb{G}_G$ ,  $\sigma_H \in \mathbb{G}_H$ ,  $\sigma_K \in \mathbb{G}_K$  be three transformations that can be applied respectively to the site graphs  $G, H$ , and  $K$ . We assume that  $(\sigma_G, \sigma_H) \in \mathbb{G}_h$  and  $(\sigma_H, \sigma_K) \in \mathbb{G}_{h'}$ .

We denote as:

$$h : G[h_l]D[h_r]H,$$

the weak partial embedding  $h$  and as:

$$h' : H[h'_l]D'[h'_r]K,$$

the weak partial embedding  $h'$ .

As seen in Sect. ??, each span of weak embeddings has pullback (that is uniquely defined modulo isomorphism) which is a partial embedding. Let us introduce  $D''$  a site graph and  $h''_l$  and  $h''_r$  be two embeddings between the site graphs  $D''$  and  $D$ , and the site graphs  $D''$  and  $D'$ , such that the partial embedding:

$$D[h_l'']D''[h_r'']D'$$

is a pullback of the span:

$$D[h_r]H[h'_l]D'.$$

Let us summarise our assumptions in the following commutative diagram:

$$\begin{array}{ccccc}
 G & & & & H \\
 & \swarrow h_l & & \searrow h_r & \\
 & D & & & D' \\
 & \swarrow h''_l & & \searrow h''_r & \\
 & D''[d] & & & 
 \end{array}$$

By Def. ??, the image  $\sigma_G \sigma_K (h'h)$  of the weak partial embedding  $h'h$  by the pair of transformation  $(\sigma_G, \sigma_K)$  is defined as the following weak partial embedding:

$$\sigma_G.G[\sigma_G.''(h_l h''_l)]((h_l h''_l)'. \sigma_G).D''[\sigma_K.''(h'_r h''_r)]\sigma_K.K.$$

Furthermore:

1. by Lem. 6.2.1, we have:

$$\sigma_H h_r h''_l = \sigma_H h'_l h''_r;$$

- 2.

$$\begin{aligned}
 (h_l h''_l)'. \sigma_G &= h''_l'. h_l'. \sigma_G & (\text{by Def. 6.2.6.(3)}) \\
 (h_l h''_l)'. \sigma_G &= h''_l'. h_r'. \sigma_H & (\text{by Def. ??})
 \end{aligned}$$

3. we also have:

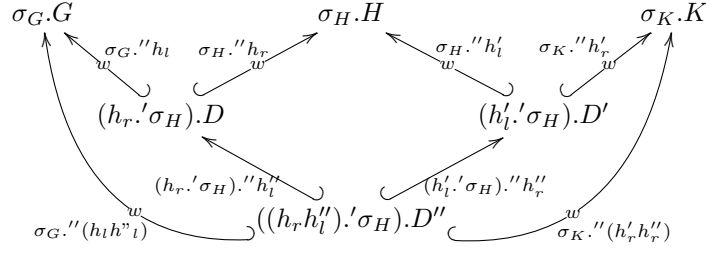
$$\begin{aligned}
 \sigma_G.''(h_l h''_l) &= \sigma_G h_l h''_l & (\text{by Def. 6.2.6.(4)}) \\
 \sigma_G.''(h_l h''_l) &= (\sigma_G.''h_l)((h_r'. \sigma_H)'. h''_l) & (\text{by Def. ??})
 \end{aligned}$$

4. and:

$$\begin{aligned}
 \sigma_K.''(h'_r h''_r) &= \sigma_K h'_r h''_r & (\text{by Def. 6.2.6.(4)}) \\
 \sigma_K.''(h'_r h''_r) &= (\sigma_K.''h'_r)((h_r'. \sigma_H)'. h''_l) & (\text{by Def. ??})
 \end{aligned}$$

5. by Def. 6.2.8, the weak embeddings  $(h_r'. \sigma_H)'. h''_l$  and  $(h'_l'. \sigma_H)'. h''_r$  are both embeddings.

Thus, we get the following commutative diagram:



We want to prove that the weak partial embedding:

$$\sigma_G.G[\sigma_G.''(h_l h''_l)]((h_l h_l'').'\sigma_G).D''[\sigma_K.''(h_r' h_r'')] \sigma_K.K,$$

is the composition of the weak partial embedding:

$$\sigma_G.G[\sigma_G.''h_l](h_r.'\sigma_H).D[\sigma_H.''h_r]\sigma_H.H$$

and the weak partial embedding:

$$\sigma_H.H[\sigma_H.''h_l'](h_l'.'\sigma_H).D'[\sigma_K.''h_r']\sigma_K.K.$$

By Sec. ?? and the previous commuting diagram, it is enough to prove that the following weak partial embedding:

$$(h_r.'\sigma_H).D[(h_r.'\sigma_H)''h_l']((h_r h_l'').'\sigma_H).D''[(h_l'.'\sigma_H)''h_r'](h_l'.'\sigma_H).D'$$

is a pullback of the following cospan:

$$(h_r.'\sigma_H).D[\sigma_H.''h_r]\sigma_H.H[\sigma_H.''h_l'](h_l'.'\sigma_H).D'.$$

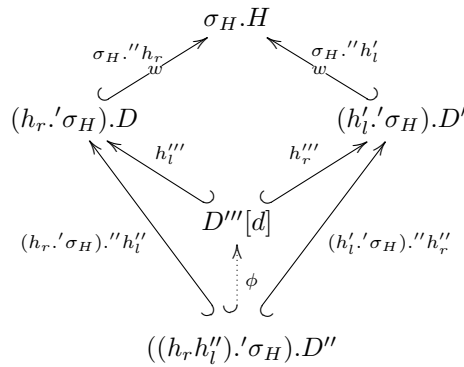
Let us consider:

$$(h_r.'\sigma_H).D[h_l''']D'''[h_r'''](h_l'.'\sigma_H).D'$$

a pullback of the cospan:

$$(h_r.'\sigma_H).D[\sigma_H.''h_r]\sigma_H.H[\sigma_H.''h_l'](h_l'.'\sigma_H).D'.$$

By definition of a pullback, there exists a (unique) embedding  $\phi$  between  $((h_r h_l'').'\sigma_H).D''$  and  $D'''$  such that  $(h_r.'\sigma_H)''h_l' = h_l''' \phi$  and  $(h_l'.'\sigma_H)''h_r' = h_r''' \phi$ . Thus we get the following commutative diagram commutes:



Now we apply the transformation  $\sigma_H^{-1}$  on the top of this commutative diagram, and we propagate it downwards the commutative square and both commutative triangles by applying successively Lem. 6.2.1, and Def. 6.2.6.(4) (twice), and Def. 6.2.8 (six times).

Since, the seven following equalities:

1.

$$\begin{aligned} \sigma_H^{-1}(\sigma_H.H) &= (\sigma_H^{-1} \circ_H \sigma_H).H && \text{(by Def. 6.2.1.(2))} \\ \sigma_H^{-1}(\sigma_H.H) &= \varepsilon_H.H && \text{(by Def. ??)} \\ \sigma_H^{-1}(\sigma_H.H) &= H && \text{(by Def. 6.2.1.(2))} \end{aligned}$$



2.

$$\begin{aligned}\sigma_H^{-1} \cdot "(\sigma_H \cdot "h_r) &= (\sigma_H^{-1} \circ_H \sigma_H) \cdot "h_r && (\text{by Def. 6.2.5.(1)}) \\ \sigma_H^{-1} \cdot "(\sigma_H \cdot "h_r) &= \varepsilon_H \cdot "h_r && (\text{by Def. ??}) \\ \sigma_H^{-1} \cdot "(\sigma_H \cdot "h_r) &= h_r && (\text{by Def. 6.2.5.(1)})\end{aligned}$$

3.

$$\begin{aligned}\sigma_H^{-1} \cdot "(\sigma_H \cdot "h'_l) &= (\sigma_H^{-1} \circ_H \sigma_H) \cdot "h'_l && (\text{by Def. 6.2.5.(1)}) \\ \sigma_H^{-1} \cdot "(\sigma_H \cdot "h'_l) &= \varepsilon_H \cdot "h'_l && (\text{by Def. ??}) \\ \sigma_H^{-1} \cdot "(\sigma_H \cdot "h'_l) &= h'_l && (\text{by Def. 6.2.5.(1)})\end{aligned}$$

4.

$$\begin{aligned}& ((\sigma_H \cdot "h_r) \cdot ' \sigma_H^{-1}) \cdot ((h_r \cdot ' \sigma_H) \cdot D) \\ &= (((\sigma_H \cdot "h_r) \cdot ' \sigma_H^{-1}) \circ_D (h_r \cdot ' \sigma_H)) \cdot D && (\text{by Def. 6.2.1.(2)}) \\ &= (h_r \cdot ' (\sigma_H^{-1} \circ_H \sigma_H)) \cdot D && (\text{by Def. 6.2.5.(3)}) \\ &= (h_r \cdot ' \varepsilon_H) \cdot D && (\text{by Def. ??}) \\ &= \varepsilon_D \cdot D && (\text{by Def. 6.2.5.(2)}) \\ &= D && (\text{by Def. 6.2.1.(2)})\end{aligned}$$

5.

$$\begin{aligned}& ((\sigma_H \cdot "h'_l) \cdot ' \sigma_H^{-1}) \cdot ((h'_l \cdot ' \sigma_H) \cdot D') \\ &= (((\sigma_H \cdot "h'_l) \cdot ' \sigma_H^{-1}) \circ_{D'} (h'_l \cdot ' \sigma_H)) \cdot D' && (\text{by Def. 6.2.1.(2)}) \\ &= (h'_l \cdot ' (\sigma_H^{-1} \circ_H \sigma_H)) \cdot D' && (\text{by Def. 6.2.5.(3)}) \\ &= (h'_l \cdot ' \varepsilon_H) \cdot D' && (\text{by Def. ??}) \\ &= \varepsilon_{D'} \cdot D' && (\text{by Def. 6.2.5.(2)}) \\ &= D' && (\text{by Def. 6.2.1.(2)})\end{aligned}$$

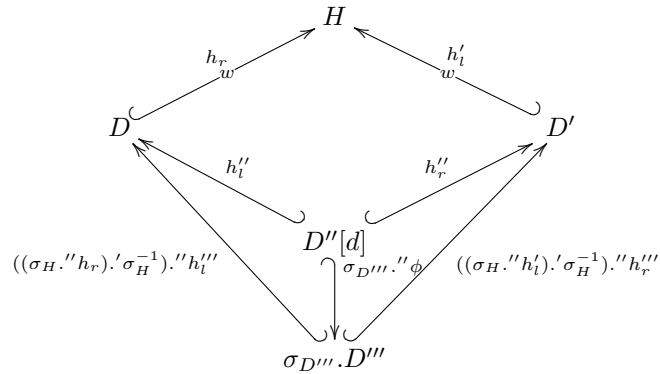
6.

$$\begin{aligned}& ((\sigma_H \cdot "h_r) \cdot ' \sigma_H^{-1}) \cdot "((h_r \cdot ' \sigma_H) \cdot "h'_l) \\ &= (((\sigma_H \cdot "h_r) \cdot ' \sigma_H^{-1}) \circ_D (h_r \cdot ' \sigma_H)) \cdot "h'_l && (\text{by Def. 6.2.5.(1)}) \\ &= (h_r \cdot ' (\sigma_H^{-1} \circ_H \sigma_H)) \cdot "h'_l && (\text{by Def. 6.2.5.(3)}) \\ &= (h_r \cdot ' \varepsilon_H) \cdot "h'_l && (\text{by Def. ??}) \\ &= \varepsilon_D \cdot "h'_l && (\text{by Def. 6.2.5.(2)}) \\ &= h'_l && (\text{by Def. 6.2.5.(1)})\end{aligned}$$

7.

$$\begin{aligned}& ((\sigma_H \cdot "h'_l) \cdot ' \sigma_H^{-1}) \cdot "((h'_l \cdot ' \sigma_H) \cdot "h_r) \\ &= (((\sigma_H \cdot "h'_l) \cdot ' \sigma_H^{-1}) \circ_{D'} (h'_l \cdot ' \sigma_H)) \cdot "h_r && (\text{by Def. 6.2.5.(1)}) \\ &= (h'_l \cdot ' (\sigma_H^{-1} \circ_H \sigma_H)) \cdot "h_r && (\text{by Def. 6.2.5.(3)}) \\ &= (h'_l \cdot ' \varepsilon_H) \cdot "h_r && (\text{by Def. ??}) \\ &= \varepsilon_{D'} \cdot "h_r && (\text{by Def. 6.2.5.(2)}) \\ &= h_r && (\text{by Def. 6.2.5.(1)})\end{aligned}$$

hold, we obtain the following commutative diagram:



where  $\sigma_{D'''} = ((\sigma_H \cdot "h_r) h'_l) \cdot ' \sigma_H^{-1}$ .

Since the partial embedding:

$$D[h_l'']D''[h_r'']D'$$

is a pullback of the span:

$$D[h_r]H[h_l']D',$$

there exists a (unique) embedding between the site graph  $\sigma_{D'''}D'''$  and the site graph  $D''$ . It follows that the embedding  $\sigma_{D'''}\phi$  is an isomorphism.

Thus, by Prop. 6.2.8, the embedding  $\phi$  is an isomorphism as well, which ensures that  $D''$  and the span

$$(h_r.\sigma_H).D[(h_r.\sigma_H)''h_l'']((h_rh_l'')\sigma_H).D''[(h_l'\sigma_H)''h_r'']((h_l'\sigma_H).D'$$

is a pullback of the cospan:

$$(h_r.\sigma_H).D[\sigma_H''h_r]\sigma_H.H[\sigma_H''h_l'] (h_l'\sigma_H).D'.$$

**Propriété 6.2.24** Let  $\mathbb{G}$  be a valid set of transformations over site graphs. Let  $L, R, D, L', R'$ , and  $D'$  be six site graphs. Let  $h$  be a partial embedding between the site graphs  $L$  and  $R$  with the domain  $D$ ,  $h'$  be a partial embedding between the site graphs  $L'$  and  $R'$  with the domain  $D'$ ,  $f$  be an embedding between the site graph  $L$  and the site graph  $L'$ , and  $g$  be an embedding between the site graph  $R$  and the site graph  $R'$ , such that:  $h'f = gh$ .

Let  $(\sigma_{L'}, \sigma_{R'}) \in \mathbb{G}_{h'}$  be a pair of transformations that can be applied to the partial embedding  $h'$ , such that the pair of transformation  $(f.\sigma_{L'}, g.\sigma_{R'})$  can be applied to the partial embedding  $h$  (i.e. such that  $(f.\sigma_{L'}, g.\sigma_{R'}) \in \mathbb{G}_h$ ).

Then the following assertion holds:

$$(\sigma_{L'}\sigma_{R'}h')(\sigma_{L'}''f) = (\sigma_{R'}''g)(f.\sigma_{L'}g.\sigma_{R'}h).$$

**Preuve 6.2.6** Let  $\mathbb{G}$  be a valid set of transformations over site graphs. Let  $L, R, D, L', R'$ , and  $D'$  be six site graphs. Let  $h$  be a partial embedding between the site graphs  $L$  and  $R$  with the domain  $D$ ,  $h'$  be a partial embedding between the site graphs  $L'$  and  $R'$  with the domain  $D'$ ,  $f$  be an embedding between the site graph  $L$  and the site graph  $L'$ , and  $g$  be an embedding between the site graph  $R$  and the site graph  $R'$ , such that:  $h'f = gh$ .

Let  $(\sigma_{L'}, \sigma_{R'}) \in \mathbb{G}_{h'}$  be a pair of transformations that can be applied to the partial embedding  $h'$ , such that  $(f.\sigma_{L'}, g.\sigma_{R'}) \in \mathbb{G}_h$ .

We have:

1.  $(\sigma_{L'}, \sigma_{R'}) \in \mathbb{G}_{h'}$ ;
2.  $(f.\sigma_{L'}, g.\sigma_{R'}) \in \mathbb{G}_{h'}$ ;
3.  $(f.\sigma_{L'}, \sigma_{L'}) \in \mathbb{G}_{([L, f])}$  (by Prop. 6.2.22);
4.  $(g.\sigma_{R'}, \sigma_{L'}) \in \mathbb{G}_{([L, f])}$  (by Prop. 6.2.22).

It follows that:

$$\begin{aligned} & (\sigma_{L'}\sigma_{R'}h')(\sigma_{L'}''f) \\ &= (\sigma_{L'}\sigma_{R'}h')((f.\sigma_{L'})\sigma_{L'}(L, f)) && \text{(by Prop. 6.2.22)} \\ &= (f.\sigma_{L'})\sigma_{R'}(h'f) && \text{(by Prop. 6.2.23)} \\ &= (g.\sigma_{R'})\sigma_{R'}(h'f) && \text{(by Def. 6.2.12)} \\ &= (g.\sigma_{R'})\sigma_{R'}(gh) && \text{(since } h'f = fh) \\ &= ((g.\sigma_{R'})\sigma_{R'}(R, g))(f.\sigma_{L'}g.\sigma_{R'}h) && \text{(by Prop. 6.2.23)} \\ &= (\sigma_{R'}''g)(f.\sigma_{L'}g.\sigma_{R'}h) && \text{(by Prop. 6.2.22)} \end{aligned}$$

### Induced group action over rules

Now we focus of the case when the partial embedding is a rule. By Def. 6.2.12, the image of a rule by a pair of transformations in a valid set of transformations over site graphs, is a rule. Thus the structure of group action can be lifted to rules, as expressed in the following proposition.

**Propriété 6.2.25 (action of pair of transformations over rules)** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs and  $r$  be a rule.*

*Then, the function which maps any pair  $((\sigma_L, \sigma_R), r')$  in the set  $\mathbb{G}_r \times \{\sigma_r.''r \mid \sigma_r \in \mathbb{G}_r\}$ , to the rule  $\sigma_L \sigma_R r$ , is a group action.*

As it was the case for site graphs, the number of automorphisms of a rule  $r$  and the number of rules in its orbit that are isomorphic to the rule  $r$  are related by the following proposition.

**Propriété 6.2.26** *Let  $\mathbb{G}$  be a valid set of transformations over site graphs. For any rule  $r$  and for any pair of permutations  $\sigma \in \mathbb{G}_r$ ,*

$$\frac{r}{\sigma.''r} = \frac{\text{card}([\sigma.''r]_{iso} \cap [\sigma.''r])}{\text{card}([r]_{iso} \cap [r])},$$

*for any rule  $r'$ , the value  $r'$  denotes the number of automorphisms of the rule  $r'$  (that is say the number of pairs of automorphisms  $(h_L, h_R)$  such that the triple  $(r', h_L, h_R)$  is a refinement of the rule  $r'$ ).*

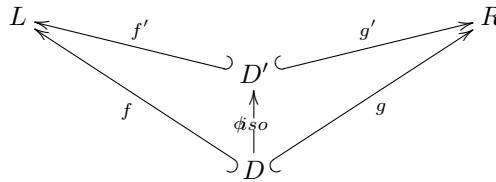
**Preuve 6.2.2** *The proof is similar as the proof of Prop. 6.2.10.*

We notice that some constraints in Def. 6.2.12 are redundant. Indeed, it would have been enough to assume that the image of a rule by a pair of transformations in a valid set of transformations satisfies the constraints 2, 3 and 4 of Def. ???. For any set of transformations  $\mathbb{G}$  such that the restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings distributes both over the composition of transformations (e.g. see def. 6.2.5) and over the composition of weak embeddings (e.g. see def. 6.2.6), and such that the image of an embedding by a transformation in  $\mathbb{G}$  is an embedding (e.g. see def. 6.2.8), the image of a partial embedding  $h$  the domain of which is maximal, by a pair of transformations in  $\mathbb{G}_h$ , is also a partial embedding the domain of which is maximal. This is formalised in the following proposition.

**Propriété 6.2.27** *Let  $\mathbb{G}$  be a set of transformations over site graphs such that the restriction of the transformations in  $\mathbb{G}$  to the domain of weak embeddings distributes both over the composition of transformations (e.g. see def. 6.2.5) and over the composition of weak embeddings (e.g. see def. 6.2.6), and such that the image of an embedding by a transformation in  $\mathbb{G}$  is an embedding (e.g. see def. 6.2.8). Let  $L$ ,  $R$ , and  $D$  be three site graphs. Let  $f$  be an embedding between the site graphs  $D$  and  $L$ , and  $g$  be an embedding between the site graphs  $D$  and  $R$ . We denote by  $h$  the following partial embedding:*

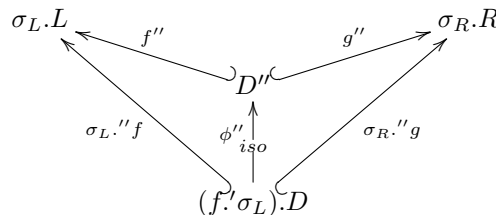
$$h : L[f]D[g]R.$$

*We assume that the site graph  $D$  is a maximal common site graph between the site graph  $L$  and the site graph  $R$ , that is to say that we assume that for any site graph  $D'$ , any embedding  $\phi$  between  $D$  and  $D'$ , any embedding  $f'$  between the site graphs  $D'$  and  $L$ , and any embedding  $g'$  between the site graphs  $D'$  and  $R$ , such that  $f = f'\phi$  and  $g = g'\phi$ , the embedding  $\phi$  is necessarily an isomorphism.*

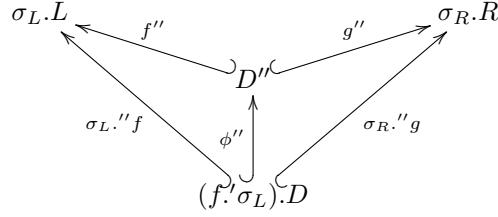


*We consider  $(\sigma_L, \sigma_R) \in \mathbb{G}_h$  a pair of transformations in  $\mathbb{G}_h$  that can be applied to the partial embedding  $h$ .*

*Then, for any site graph  $D''$ , any embedding  $f''$  between the site graph  $D''$  and the site graph  $\sigma_L.L$ , and any embedding  $g''$  between the site graph  $D''$  and the site graph  $\sigma_R.R$ , and any embedding  $\phi''$  between the site graph  $(f'.\sigma_L).D$  and the site graph  $D''$ , such that  $\sigma_L.''f = f''\phi''$  and  $\sigma_R.''g = g''\phi''$ , the embedding  $\phi''$  is necessarily an isomorphism.*



**Preuve 6.2.7** We sketch the proof of Prop. 6.2.27. We take the same notations as in Prop. 6.2.27.  
By assumption, the following diagram:



commutes.

Then, we apply the pair of transformations  $(\sigma_L^{-1}, \sigma_R^{-1})$  to the diagram above, and propagate both transformations  $\sigma_L^{-1}$  and  $\sigma_R^{-1}$  downwards by applying Def. 6.2.6.(4) (twice), and Def. 6.2.8 (five times). Since the five following equalities:

1.

$$\begin{aligned}
 \sigma_L^{-1}.(\sigma_L.L) &= (\sigma_L^{-1} \circ_L \sigma_L).L && \text{(by Def. 6.2.1.(2))} \\
 \sigma_L^{-1}.(\sigma_L.L) &= \varepsilon_L.L && \text{(by Def. ??)} \\
 \sigma_L^{-1}.(\sigma_L.L) &= L && \text{(by Def. 6.2.1.(2))}
 \end{aligned}$$

2.

$$\begin{aligned}
 \sigma_R^{-1}.(\sigma_R.R) &= (\sigma_R^{-1} \circ_R \sigma_R).R && \text{(by Def. 6.2.1.(2))} \\
 \sigma_R^{-1}.(\sigma_R.R) &= \varepsilon_R.R && \text{(by Def. ??)} \\
 \sigma_R^{-1}.(\sigma_R.R) &= R && \text{(by Def. 6.2.1.(2))}
 \end{aligned}$$

3.

$$\begin{aligned}
 &((\sigma_L.''f)'.\sigma_L^{-1}).((f'.\sigma_L).D) \\
 &= (((\sigma_L.''f)'.\sigma_L^{-1}) \circ_D (f'.\sigma_L)).D && \text{(by Def. 6.2.1.(2))} \\
 &= (f'.(\sigma_L^{-1} \circ_L \sigma_L)).D && \text{(by Def. 6.2.5.(3))} \\
 &= (f'.\varepsilon_L).D && \text{(by Def. ??)} \\
 &= \varepsilon_D.D && \text{(by Def. 6.2.5.(2))} \\
 &= D && \text{(by Def. 6.2.1.(2))}
 \end{aligned}$$

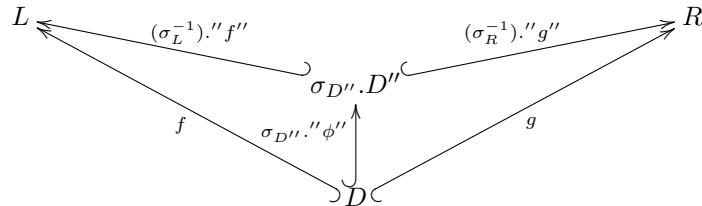
4.

$$\begin{aligned}
 \sigma_L^{-1}.''(\sigma_L.''f) &= (\sigma_L^{-1} \circ_L \sigma_L).''f && \text{(by Def. 6.2.5.(1))} \\
 \sigma_L^{-1}.''(\sigma_L.''f) &= \varepsilon_L.''f && \text{(by Def. ??)} \\
 \sigma_L^{-1}.''(\sigma_L.''f) &= f && \text{(by Def. 6.2.5.(1))}
 \end{aligned}$$

5.

$$\begin{aligned}
 \sigma_R^{-1}.''(\sigma_R.''g) &= (\sigma_R^{-1} \circ_R \sigma_R).''g && \text{(by Def. 6.2.5.(1))} \\
 \sigma_R^{-1}.''(\sigma_R.''g) &= \varepsilon_R.''g && \text{(by Def. ??)} \\
 \sigma_R^{-1}.''(\sigma_R.''g) &= g && \text{(by Def. 6.2.5.(1))}
 \end{aligned}$$

the following diagram:



where the transformation  $\sigma_{D''} \in \mathbb{G}_{D''}$  is defined as  $f''.(\sigma_L^{-1})$ , commutes.

Since  $D$  is maximal, it follows that  $\sigma_{D''}.'' \phi''$  is an isomorphism. So, by Prop. 6.2.8,  $\phi''$  is an isomorphism (and thus the domain  $(f'.\sigma_L).D$  of the partial embedding  $\sigma_L \sigma_R h$  is maximal as well).

### Induced group action over refinements

We are now almost ready to apply pairs of transformations over refinements. Before proving that the image of refinements by a pair of transformations is indeed a refinement, we prove two intermediary lemmas.

**Lemme 6.2.2** *Let  $L$ ,  $R$ , and  $R'$  be three site graphs. Let  $r$  be a rule between the site graphs  $L$  and  $R$  and  $r'$  be a rule between the site graphs  $L$  and  $R'$ . We assume that there exists a weak embedding  $\phi$  between the site graph  $R$  and  $R'$  and  $\phi'$  be a weak embedding  $\phi'$  between the site graph  $R'$  and  $R$  such that we have  $r' = \phi r$  and  $r = \phi' r'$ . Under these assumptions, the weak embeddings  $\phi$  and  $\phi'$  are two isomorphisms.*

**Preuve 6.2.8** *Let  $L$ ,  $R$ , and  $R'$  be three site graphs. Let  $r$  and  $r'$  be two rules between the site graphs  $L$  and  $R$ , and  $L$  and  $R'$ . Let  $\phi$  and  $\phi'$  be two weak embeddings between the site graphs  $R$  and  $R'$ , and  $R'$  and  $R$  such that both  $r' = \phi r$  and  $r = \phi' r'$ .*

*Now we consider the set:*

$$\mathcal{B} := \{s \in [R] \mid \exists x \in [R] \cup \{-\} \text{telque}(s, x) \in [R]\}$$

*and the set:*

$$\mathcal{B}' := \{s \in [R'] \mid \exists x \in [R'] \cup \{-\} \text{telque}(s, x) \in [R']\}.$$

*Intuitively,  $\mathcal{B}$  (resp.  $\mathcal{B}'$ ) denotes the set of the sites of the site graph  $R$  (resp.  $R'$ ) that are necessarily bound. Since site graphs are finite. Both sets  $\mathcal{B}$  and  $\mathcal{B}'$  are finite. Since  $\phi$  is a weak embedding between the site graphs  $R$  and  $R'$ , it follows that:*

$$\text{card}(\mathcal{B}) \leq \text{card}(\mathcal{B}').$$

*Then, since  $\phi'$  is a weak embedding between the site graphs  $R'$  and  $R$ , it follows that:*

$$\text{card}(\mathcal{B}') \leq \text{card}(\mathcal{B}).$$

*Thus,*

$$\text{card}(\mathcal{B}) = \text{card}(\mathcal{B}'),$$

*and for any site  $(n, i) \in [R]$ , we have  $(n, i) \in \mathcal{B}$  is and only if  $(\phi(n), i) \in \mathcal{B}'$ .*

*Now we can prove that the weak embedding  $\phi$  is an embedding. It is indeed sufficient to prove that for any site  $(n, i) \in [R]$  such that  $((n, i), -) \in [R]$ , we have  $((\phi(n), i), -) \in [R']$  (that is to say that the image, by  $\phi$  of any site that is in free in the site graph  $R$ , is free as well in the site graph  $R'$ ).*

*Let  $(n, i) \in [R]$  be a site in the site graph  $R$ , such that  $((n, i), -) \in [R]$ .*

*Let us write the rule  $r$  as a partial embedding:*

$$r : L[f]D[g]R.$$

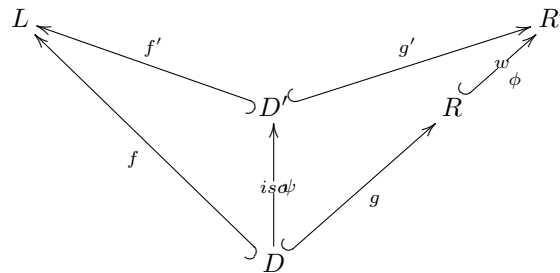
*By Sect. ??, since  $r' = \phi r$ , the following partial embedding:*

$$L[f]D[\phi g]R'$$

*is a valid representation of the rule  $r'$ .*

*Thus, there exists an isomorphism  $\psi$  between the site graphs  $D$  and  $D'$  such that  $f = f'\psi$  and  $\phi g = g'\psi$ .*

*It follows that the following diagram*



*commutes.*

*Let us consider two cases, according to whether or not, the agent  $n$  has been created by the rule  $r$ , or not.*

1. If  $n \notin (g)$ . (i.e. if the agent  $n$  has been created by the rule  $r$ )

Then  $\phi(n) \notin (g')$ .

Otherwise,

there would exists an agent  $m \in [D']$  such that:

$$g'(m) = \phi(n);$$

then, we would have:

$$\phi(g(\psi^{-1}(m))) = \phi(n);$$

and, since the function  $\phi$  is into, it would follow that:

$$g(\psi^{-1}(m)) = n,$$

which would be absurd.

By Def. ??.(4), it follows that there exists a binding state  $x \in [R'] \cup \{-\}$  such that  $((\phi(n), i), x) \in [R']$ .

2. If  $n \in (g)$ .

Let  $m \in [D]$  such that  $g(m) = n$ .

We have,  $((g(m), i), -) \in [R]$ .

Thus, by Def. ??.(3a), there exists a binding state  $x \in [L] \cup$  such that  $((f(m), i), x) \in [L]$ .

Since  $f = f'\psi$ , it follows that  $((f'(\psi(m)), i), x) \in [L]$ .

So, by Def. ??.(3a), there exists a binding state  $x' \in [R'] \cup$  such that  $((g'(\psi(m)), i), x') \in [R']$ .

Since  $g'\psi = \phi g$ , it follows that  $((\phi(g(m)), i), x') \in [R']$ .

That is to say that  $((\phi(n), i), x') \in [R']$ .

Thus, in both cases, there exists a binding state  $x' \in [R'] \cup$  such that  $((\phi(n), i), x') \in [R']$ .

Then, since  $((n, i), -) \in [R]$ , we have  $(n, i) \notin \mathcal{B}$ .

It follows that  $(\phi(n), i) \notin \mathcal{B}'$ .

So  $((\phi(n), i), -) \in \mathcal{B}'$ .

This way, the partial embedding  $\phi$  is an embedding.

With the same reasoning, up to replacing the site graph  $R$  by the site graph  $R'$  and vice versa, the rule  $r$  by the rule  $r'$  and vice versa, and the partial embedding  $\phi$  by the partial embedding  $\phi'$  and vice versa, we can conclude that the partial embedding  $\phi'$  is an embedding as well.

Thus,  $\phi$  is an embedding between the site graph  $R$  and  $R'$  and  $\phi'$  is an embedding between the site graph  $R'$  and the site graph  $R$ , so  $R$  and  $R'$  are two isomorphic site graphs, and the embeddings  $\phi$  and  $\phi'$  are both isomorphisms.

**Lemme 6.2.3** Let  $\mathbb{G}$  be a set of valid set of transformations over site graphs. Let  $r$  be a rule and  $(r', h_L, h_R)$  be a refinement of  $r$ . Under these assumptions, for any pair  $(\sigma_{L'}, \sigma_{R'}) \in \mathbb{G}_r$  of transformations that can be applied to the rule  $r'$ , both following assertions hold:

1.  $(h_L.'\sigma_{L'}, h_R.'\sigma_{R'}) \in \mathbb{G}_h$ ;
2.  $(\sigma_{L'}\sigma_{R'}r')(\sigma_{L'}.''h_L)$   
 $= (\sigma_{R'}.''h_R)(h_L.'\sigma_{L'}h_R.'\sigma_{R'}r).$

**Preuve 6.2.9** Let  $\mathbb{G}$  be a set of valid set of transformations over site graphs. Let  $r$  be a rule and  $(r', h_L, h_R)$  be a refinement of  $r$ . Let  $(\sigma_{L'}, \sigma_{R'}) \in \mathbb{G}_r$  be a pair of transformations which can be applied to the rule  $r'$ .

1. We write the rule  $r$  as a pair of two embedding  $f$  and  $g$ , and the rule  $r'$  as a pair of two embedding  $f'$  and  $g'$ . We denote by  $h_D$  the embedding between the domain of the rule  $r$  and the domain of the rule  $r'$  which makes the following diagram commutes:

We have:

$$\begin{aligned}
f.'(h_L.'\sigma_{L'}) &= (h_L f).'\sigma_{L'} && (\text{by Def. 6.2.6.(3)}) \\
f.'(h_L.'\sigma_{L'}) &= (f' h_D).'\sigma_{L'} && (\text{since } h_L f = f' h_D) \\
f.'(h_L.'\sigma_{L'}) &= h_D.'(f'.'\sigma_{L'}) && (\text{by Def. 6.2.6.(3)}) \\
f.'(h_L.'\sigma_{L'}) &= h_D.'(g'.'\sigma_{R'}) && (\text{since } (\sigma_{L'}, \sigma_{R'}) \in \mathbb{G}_{r'}) \\
f.'(h_L.'\sigma_{L'}) &= (g' h_D).'\sigma_{R'} && (\text{by Def. 6.2.6.(3)}) \\
f.'(h_L.'\sigma_{L'}) &= (h_R g).'\sigma_{R'} && (\text{since } g' h_D = h_R g) \\
f.'(h_L.'\sigma_{L'}) &= g.'(h_R.'\sigma_{R'}) && (\text{by Def. 6.2.6.(3)})
\end{aligned}$$

By Def. ??, it follows that:  $(h_L.'\sigma_{L'}, h_R.'\sigma_{R'}) \in \mathbb{G}_r$ .

2. By Prop. 6.2.24,

$$(\sigma_{L'}\sigma_{R'}r')(\sigma_{L'}.''h_L) = (\sigma_{R'}.''h_R)(h_L.'\sigma_{L'}h_R.'\sigma_{R'}r').$$

**Propriété 6.2.28** Let  $\mathbb{G}$  be a set of valid set of transformations over site graphs. Let  $r$  and  $r'$  be two rules,  $h_L$  an embedding between the left hand side of the rule  $r$  and the left hand side of the rule  $r'$ , and  $h_R$  an embedding between the right hand side of the rule  $r$  and the right hand side of the rule  $r'$ . Let  $(\sigma_{L'}, \sigma_{R'}) \in \mathbb{G}_{r'}$  be a pair of transformations that can be applied to the rule  $r'$ .

Then, the following two assertions are equivalent:

1. the triple  $(r', h_L, h_R)$  is a refinement of the rule  $r$ ;
2. the triple  $(\sigma_{L'}\sigma_{R'}r', \sigma_{L'}.''h_L, \sigma_{R'}.''h_R)$  is a refinement of the rule  $h_L.'\sigma_{L'}h_R.'\sigma_{R'}r$ .

We can use Prop. prop:carry:refinements to define a group action between any valid set of transformations and refinements.

**Definition 6.2.19 (image of a refinement by a pair of transformations)** Let  $\mathbb{G}$  be a set of transformations over site graphs. We say that the refinement  $(s, h_L, h_R)$  of a rule  $r$  and the refinement  $(s', h'_L, h'_R)$  of the rule  $r'$  are symmetric to one another with respect to the set of transformations  $\mathbb{G}$  if and only if there exists a pair of transformations  $(\sigma_L, \sigma_R) \in \mathbb{G}_r$  such that:

- $s' = \sigma_L \sigma_R s$ ;
- $h'_L = \sigma_L.''h_L$ ;
- $h'_R = \sigma_R.''h_R$ ;
- $r' = (h_L.'\sigma_L)(h_R.'\sigma_R)r$ .

In such a case, we write  $((s, h_L, h_R), r)((s', h'_L, h'_R), r')$ .

**Propriété 6.2.29** Let  $\mathbb{G}$  be a set of transformations over site graphs, then the relation  $\square$  is an equivalence relation over the pairs  $((r', h_L, h_R), r)$  such that  $r$  is a rule and  $(r', h_L, h_R)$  is a refinement of the rule  $r$ .

**Propriété 6.2.30 (action of pairs of transformations over refinements)** let  $\mathbb{G}$  be a set of permutations over site graphs and  $(s, h_L, h_R)$  be a refinement of a rule  $r$ .

The function mapping any pair  $((\sigma_L, \sigma_R), ((s', h'_L, h'_R), r'))$  such that  $(\sigma_L, \sigma_R) \in \mathbb{G}_s$ ,  $r'$  is a rule,  $(s', h'_L, h'_R)$  is a refinement of the rule  $r'$ , and  $((s, h_L, h_R), r)((s', h'_L, h'_R), r')$ , is a group action.

**Preuve 6.2.1**

## 6.3 symetries in rule-based models

So far, we have defined the notion of transformations over site graphs and we have shown how to apply these transformations over site graphs, embeddings, rules, and refinements of rules.

Now we will introduce the notion of symetries which are, intuitively, transformations which leave a set of rules unchanged and investigate under which conditions it is possible to use these symetries so as to simplify the individual or the population semantics by considering states modulo symetries.

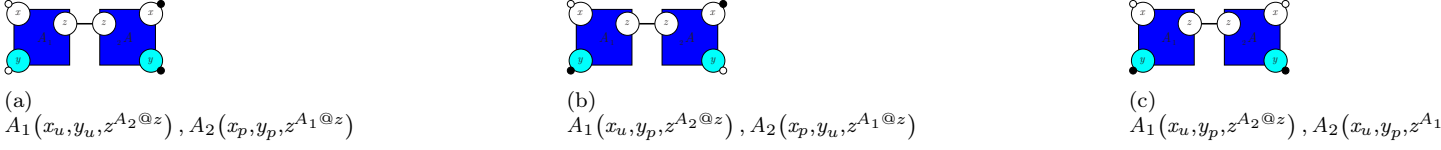


Figure 6.23: Three sites graphs. The first one  $A_1(x_u, y_u, z^{A_2 @ z}), A_2(x_p, y_p, z^{A_1 @ z})$  is both  $\mathbb{G}^{\text{AGENT}}$ -symetric and  $\mathbb{G}^{\text{SPECIES}}$ -symetric; its *iso*-equivalence class is both  $\mathbb{G}^{\text{AGENT}}$ -symetric and  $\mathbb{G}^{\text{SPECIES}}$ -symetric as well. The second one  $A_1(x_u, y_p, z^{A_2 @ z}), A_2(x_p, y_u, z^{A_1 @ z})$  is neither  $\mathbb{G}^{\text{AGENT}}$ -symetric, nor  $\mathbb{G}^{\text{SPECIES}}$ -symetric; but its *iso*-equivalence class is  $\mathbb{G}^{\text{SPECIES}}$ -symetric (but not  $\mathbb{G}^{\text{AGENT}}$ -symetric). The third one  $A_1(x_u, y_p, z^{A_2 @ z}), A_2(x_u, y_p, z^{A_1 @ z})$  is neither  $\mathbb{G}^{\text{AGENT}}$ -symetric, nor  $\mathbb{G}^{\text{SPECIES}}$ -symetric; its *iso*-equivalence class is neither  $\mathbb{G}^{\text{AGENT}}$ -symetric, nor  $\mathbb{G}^{\text{SPECIES}}$ -symetric too.

### 6.3.1 symetries among a site graph

First we introduce the notion of symetries among a site graph. The notion of symetries depends on the level of observation. In the individual semantics, we make the distinction between isomorphic site graphs. Thus, we say that a site graph is symetric with respect to a set of transformations over site graphs if, and only if, it is equal to the image of itself by any transformation (which can be applied with it) in this set of transformations.

**Definition 6.3.1** *Let  $\mathbb{G}$  be a set of transformations over site graphs. A site graph  $E$  is said to be symetric with respect to  $\mathbb{G}$  if and only if for any transformation  $\sigma_E \in \mathbb{G}_E$  that can be applied to the site graph  $E$ , the site graphs  $\sigma_E.E$  and  $E$  are the same.*

*In such a case, we write that the site graph  $E$  is  $\mathbb{G}$ -symetric.*

Obviously, since a subset of transformations contains less transformations than the overlying set of transformations. Whenever a site graph is symetric with respect to a given set of transformations over site graphs, it is necessarily symetric with respect to any subset of this set of transformations, as stated in the following proposition.

**Propriété 6.3.1** *Let  $\mathbb{G}$  and  $\mathbb{H}$  be two sets of transformations over site graphs, such that the set  $\mathbb{H}$  is a subset of transformations of the set of transformations  $\mathbb{G}$ . Let  $E$  be a site graph.*

*Under these assumptions, if the site graph  $E$  is  $\mathbb{G}$ -symetric, then the site graph  $E$  is  $\mathbb{H}$ -symetric as well.*

**Exemple 6.3.1** *We take the same signature as in Exa. 6.2.1. We consider both the set  $\mathbb{G}^{\text{AGENT}}$  of heterogeneous permutations of sites (as defined in Exa. 6.2.1) and the set  $\mathbb{G}^{\text{SPECIES}}$  of the homogeneous permutations of sites (as defined in Exa. 6.2.2).*

*We consider the three site graphs that are drawn in Fig. 6.23.*

1. *The following site graph:*

$$A_1(x_u, y_u, z^{A_2 @ z}), A_2(x_p, y_p, z^{A_1 @ z})$$

*(e.g. see Fig. 6.23(a)) is  $\mathbb{G}^{\text{AGENT}}$ -symetric (thus, by Prop. 6.3.1, it is  $\mathbb{G}^{\text{SPECIES}}$ -symetric as well). Indeed, in both of its agents, the site  $x$  and the site  $y$  have the same internal state, thus it does not change anything to swap the state of these two sites in a given agent.*

2. *The following site graph:*

$$A_1(x_u, y_p, z^{A_2 @ z}), A_2(x_p, y_u, z^{A_1 @ z})$$

*(e.g. see Fig. 6.23(b)) is not  $\mathbb{G}^{\text{SPECIES}}$ -symetric (and thus, by Prop. 6.3.1, it is not  $\mathbb{G}^{\text{AGENT}}$ -symetric as well). In particular, if we apply the homogeneous permutation of sites  $\{1, 2\}$  which consists in swapping the state of the sites  $x$  and  $y$  both in the agent with the label 1, and in the agent with label 2, we get the following site graph:*

$$A_1(x_p, y_u, z^{A_2 @ z}), A_2(x_u, y_p, z^{A_1 @ z})$$

*which is isomorphic to the initial one, but not equal.*



3. The following site graph:

$$A_1(x_u, y_p, z^{A_2 \otimes z}), A_2(x_u, y_p, z^{A_1 \otimes z})$$

(e.g. see Fig. 6.23(c)) is not  $\mathbb{G}^{\text{SPECIES}}$ -symmetric (and thus, by Prop. 6.3.1, it is not  $\mathbb{G}^{\text{AGENT}}$ -symmetric as well). In particular, if we apply the homogeneous permutation of sites  $\{1, 2\}$  which consists in swapping the state of the sites  $x$  and  $y$  both in the agent with the label 1, and in the agent with label 2, we get the following site graph:

$$A_1(x_p, y_u, z^{A_2 \otimes z}), A_2(x_p, y_u, z^{A_1 \otimes z})$$

which is not even isomorphic to the initial one.

In the population semantics, site graphs are seen modulo isomorphisms. As a consequence, we would like to relax the notion of symmetry and say that a site graph is symmetric with respect to a set of transformations over site graphs, whenever this site graph is isomorphic to the image of itself by each transformation (which can be applied to it) in this set of transformations. So as to make sure that this definition makes sense, we have to check that whenever a site graph  $E$  is such that the image of this site graph  $E$  by each transformation that can be applied to it, is symmetric to the site graph  $E$  itself, then, any site graph that is isomorphic to the site graph  $E$  also satisfies this property. This is the purpose of the following proposition, which holds only for valid sets of transformations.

**Propriété 6.3.2** Let  $\mathbb{G}$  be a valid set of transformations over site graphs and  $E$  and  $F$  be two isomorphic site graphs.

If  $[E] \subseteq [E]_{\text{iso}}$ , then  $[F] \subseteq [F]_{\text{iso}}$ .

Now we can safely introduce the notion of symmetric *iso*-equivalence class of site graphs, as follows:

**Definition 6.3.2** Let  $\mathbb{G}$  be a valid set of transformations over site graphs. A class  $[E]_{\text{iso}}$  of *iso*-equivalence of site graphs is said to be symmetric if and only if for any transformation  $\sigma_E \in \mathbb{G}_E$  that can be applied to the site graph  $E$ , the site graphs  $\sigma_E.E$  and  $E$  are isomorphic (i.e.  $[E] \in [E]_{\text{iso}}$ ).

In the next propositions, we state that whenever a site graph is symmetric with respect to a set of valid transformations, then its *iso*-equivalence class is also symmetric with respect to the same set of valid transformations, and that whenever the *iso*-equivalence class of a site graph is symmetric with respect to a set of valid transformations over site graphs, then it is also symmetric with respect to any subset of this set of valid transformations.

**Propriété 6.3.3** Let  $\mathbb{G}$  be a valid set of transformations over site graphs, and  $E$  be a site graph.

If the site graph  $E$  is  $\mathbb{G}$ -symmetric, then the *iso*-equivalence class  $[E]_{\text{iso}}$  of the site graph  $E$  is  $\mathbb{G}$ -symmetric as well.

**Propriété 6.3.4** Let  $\mathbb{G}$  and  $\mathbb{H}$  be two valid sets of transformations over site graphs, such that the set  $\mathbb{H}$  is a subset of transformations of the set of transformations  $\mathbb{G}$ .

Let  $E$  be a site graph. Under these assumptions, if the *iso*-equivalence class  $[E]_{\text{iso}}$  of the site graph  $E$  is  $\mathbb{G}$ -symmetric, then the *iso*-equivalence class  $[E]_{\text{iso}}$  of the site graph  $E$  is  $\mathbb{H}$ -symmetric as well.

**Exemple 6.3.2** We take the same signature as in Exa. 6.2.1. We consider both the set  $\mathbb{G}^{\text{AGENT}}$  of heterogeneous permutations of sites (as defined in Exa. 6.2.1) and the set  $\mathbb{G}^{\text{SPECIES}}$  of the homogeneous permutations of sites (as defined in Exa. 6.2.2).

We consider the *iso*-equivalence class of the three site graphs that are drawn in Fig. 6.23.

1. The *iso*-equivalence class of the site graph:

$$A_1(x_u, y_u, z^{A_2 \otimes z}), A_2(x_p, y_p, z^{A_1 \otimes z})$$

(e.g. see Fig. 6.23(a)) is both  $\mathbb{G}^{\text{AGENT}}$ -symmetric and  $\mathbb{G}^{\text{SPECIES}}$ -symmetric, since this site graph is both  $\mathbb{G}^{\text{AGENT}}$ -symmetric and  $\mathbb{G}^{\text{SPECIES}}$ -symmetric (e.g. see Exa. 6.3.1).

2. The iso-equivalence class of the site graph:

$$A_1(x_u, y_p, z^{A_2 \otimes z}), A_2(x_p, y_u, z^{A_1 \otimes z})$$

(e.g. see Fig. 6.23(b)) is  $\mathbb{G}^{\text{SPECIES}}$ -symmetric, but not  $\mathbb{G}^{\text{AGENT}}$ -symmetric. There are only two homogeneous permutations of sites that can be applied to this site graph, the identity permutation  $\emptyset$  and the permutation  $\{1, 2\}$ . The former one does not change the site graph (by definition), whereas the second one consists in swapping the state of the sites  $x$  and  $y$  in both agents of the site graph, which transforms it into the following site graph:

$$A_1(x_p, y_u, z^{A_2 \otimes z}), A_2(x_u, y_p, z^{A_1 \otimes z}).$$

We notice that the so-obtained site graph is isomorphic to the initial one. Now if we consider the set of heterogeneous permutations of sites, we can apply the permutation  $\{1\}$ , which consists in swapping the state of the sites  $x$  and  $y$  in the agent with the label 1,. As a result, we get the following site graph:

$$A_1(x_p, y_u, z^{A_2 \otimes z}), A_2(x_p, y_u, z^{A_1 \otimes z})$$

which is not isomorphic to the initial one.

3. The iso-equivalence class of the site graph:

$$A_1(x_u, y_p, z^{A_2 \otimes z}), A_2(x_u, y_p, z^{A_1 \otimes z})$$

(e.g. see Fig. 6.23(c)) is not  $\mathbb{G}^{\text{SPECIES}}$ -symmetric (and thus, by Prop. 6.3.4, it is not  $\mathbb{G}^{\text{AGENT}}$ -symmetric as well). In particular, if we apply the homogeneous permutation of sites  $\{1, 2\}$  which consists in swapping the state of the sites  $x$  and  $y$  both in the agent with the label 1, and in the agent with label 2, we get the following site graph:

$$A_1(x_p, y_u, z^{A_2 \otimes z}), A_2(x_p, y_u, z^{A_1 \otimes z})$$

which is not isomorphic to the initial one.

### 6.3.2 symetries among a set of rules

Defining the notion of symetries among a set of rules is more difficult. One expect ta set of rules  $X$  to be symmetric with respect to a set of transformations over site graphs, if and only if, the image of a rule by a pair of transformations which can be applied with it, is also a rule in the set  $X$  with the same corrected rate. Yet, this definition is too restrictive, because rules can be defined modulo isomorphisms. As a consequence, in a symmetric set of rules should remain symmetric whenever we gather two isomorphic rules into a single one (while summing their rates) or whenever we split a rule into two (while preserving the overall rate). We notice that it is easier to use families of rules instead of set of rules, since a given rule can occur twice in a family, unlike in a set.

We are going to proceed as follows. Firstly we will define a strong notion of symmetry for family of rules, in which we do not see rules up to isomorphisms. Then, we will define a notion of equivalence among rule families, and we will define symmetric families of rules are the ones which are equivalent to strongly symmetric ones. Lastly we will provide a more convenient characterisation of symmetric families of rules.

Firstly, we define a model as a family of rules.

**Definition 6.3.3** A model is a family of pairs  $(r_i, k_i)_{i \in I}$  indexed over a finite set  $I$ , where for each element  $i \in I$ ,  $r_i$  is a rule and  $k_i$  is a rate constant in  $\mathbb{R}_{\geq 0}$ .

We define, as follows, strongly symmetric models as the ones in which the image of each rule by each pair of transformations which can be applied to it, is another rule with the same corrected rate:

**Definition 6.3.4 (strongly symmetric model)** Let  $M := (r_i, k_i)_{i \in I}$  be a model and  $\mathbb{G}$  be a valid set of transformations over site graphs.

We say that the model  $M$  is strongly symmetric with respect to the set of transformations  $\mathbb{G}$  if and only if:

1. for any element  $i, j \in I$  such that  $i \neq j$ , we have  $r_i \neq r_j$ ;
2. for any element  $i \in I$  and any pair of permutation  $\sigma \in \mathbb{G}_{r_i}$ , there exists an element  $j \in I$ , such that:

$$(a) \sigma.r_i = r_j.$$

$$(b) \frac{k_i}{[L_i, L_i]} = \frac{k_j}{[L_j, L_j]},$$

where:

- i.  $L_i$  is the left hand side of the rule  $r_i$ ,
- ii.  $L_j$  is the left hand side of the rule  $r_j$ ,
- iii. and for any site graph  $[E, E]$  denotes the number of automorphisms in the site graph  $E$ .

According to Def. 6.3.4, strongly symetric models are those in which (constraint 1) no rule can occur twice (which does not prevent from having several distinct isomorphic rules), (constraint 2a) the image of a rule by a pair of transformations is always a rule and (constraint 2b) the corrected rates of a rule and its image is the same.

Now we define the equivalence relations between models, we allow the replacement of a rule by an isomorphic one, the merging of two isomorphic rules, or the splitting of a rule into two ones.

**Definition 6.3.5 (model equivalence)** We define the equivalence relation over models as the least equivalence relation such that:

- 1.  $(r_i, k_i)_{i \in I} (r_{\sigma(j)}, k_{\sigma(j)})_{j \in J}$  for any one to one function  $\sigma$  between  $J$  and  $I$ ;
- 2.  $(r_i, k_i)_{i \in I} (s_i, k_i)_{i \in I}$  whenever  $r_i$  isos  $s_i$  for any element  $i$  in the set  $I$ ;
- 3.  $(r_i, k_i)_{i \in I} (r_i, k'_i)_{i \in I \setminus \{i_1\}}$  with

$$k'_i = \begin{cases} k_{i_0} + k_{i_1} & \text{whenever } i = i_0 \\ k_i & \text{otherwise;} \end{cases}$$

for any distinct elements  $i_0, i_1$  in the set  $I$ , such that  $r_{i_0} = r_{i_1}$ .

Intuitively, in Def. 6.3.5, the constraint 1 allows the re-indexing of rules; the constraint 2 allows the replacement of a rule by an isomorphic one; the constraint 3 allows, in the direct way, the merging of identical (or isomorphic) rules into a single one while summing their rates; the constraint 3 allows, in the other way, the splitting of a rule into several (isomorphic) ones while distributing its rate among the new ones.

Now we can define a more general notion of symetric models.

**Definition 6.3.6 (symetric model)** A model is said to be symetric with respect to a valid set of transformations  $\mathbb{G}$  over site graphs if and only if it is -equivalent to a model that is strongly symetric with respect to the set of transformations  $\mathbb{G}$ .

Def. 6.3.6 does not provide an effective decision procedure to decide wether or not a model is symetric with respect to a set of transformations.

Now, we give a direct characterisation of symetric models.

**Propriété 6.3.5** Let  $M := (r_i, k_i)_{i \in I}$  be a model and  $\mathbb{G}$  be a valid set of transformations over site graphs. Then, the following assertions are equivalent:

- 1. the model  $M$  is symetric with respect to the set of transformations  $\mathbb{G}$ ;
- 2. for two rules  $r' : L' R'$  and  $r'' : L'' R''$  such that  $r' r''$ , the following equation is satisfied:

$$\sum_{\substack{i \in \text{Itelque} \\ r_i \text{isor}'}} \frac{r' \cdot k_i}{[L', L']} = \sum_{\substack{j \in \text{Itelque} \\ r_j \text{isor}''}} \frac{r'' \cdot k_j}{[L'', L'']},$$

where:

- (a) for any rule  $r$ , the value  $r$  denotes the number of automorphisms of the rule  $r$  (that is say the number of pairs of automorphisms  $(h_L, h_R)$  such that the triple  $(r, h_L, h_R)$  is a refinement of the rule  $r$ );
- (b) and, for any site graph  $E$ , the value  $[E, E]$  denotes the number of automorphisms of the site graph  $E$ ,

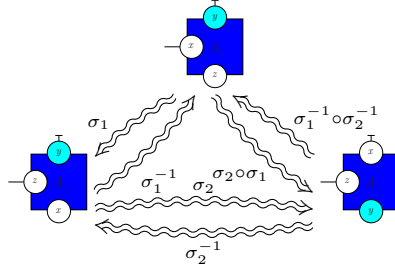


Figure 6.24: Examples of permutations of the sites of an agent. The permutation  $\sigma_1$  swaps the states of the site  $x$  and the site  $z$  in the agent 1. The permutation  $\sigma_2$  swaps the states of the site  $x$  and the site  $y$  in the agent 1.

The checking whether, or not, a model is symmetric with respect to a given (valid) set of transformations between site graph can be done in two steps. Firstly, we gather rules into *iso*-equivalence classes. Then, we scan over the *iso*-equivalence classes of rules of the model in the following way. We pick an *iso*-equivalence class of rules in the model among the ones that we have not already dealt with. We pick an arbitrary rule in that *iso*-equivalence class and we compute the image of this rule by each pair of transformations that can be applied to this rule. For each so-obtained image, we check whether the condition about the rates of the *iso*-equivalence classes of the initial rules and its image is satisfied. If it is satisfied for each image, we mark the isomorphic class of each image as dealt with, and we keep on iterating, otherwise we have proved that the model is not symmetric. If we can deal with each rule without failing a condition over the rates of *iso*-equivalence classes of rules, then we have proved that the model is symmetric.

Notice, that in Prop. 6.3.5, thanks to Prop. 6.2.26, we could have multiply the rate of each rule  $r$  by the number  $\text{card}([r]_{iso} \cap [r])$  of the rules which are both symmetric and isomorphic to this rule, instead of multiplying by the number of automorphisms of this rule.

## 6.4 Permutations of sites in site graphs

Before giving the main properties about the semantic of symmetric models, we formalise further the notion of permutations of sites that we have used in our running example. In particular, in our running examples, we have only swapped pairs of property sites, and we have wisely avoided permuting pairs of binding sites. In this section, we propose to instantiate the algebraic framework that we have introduced so as to model permutations of sites in general.

In this section, we disallow the use of binding types in site graphs. We make this assumption because whenever an embedding maps a site with a binding type to a site that is bound to another site, it is quite tricky to restrict a permutation of sites that can be applied to the image of this embedding, into a permutation of sites that can be applied to the domain of this embedding, while ensuring the group structure of the sets of permutations and the distributivity of the restriction of the permutations of sites over the product between these permutations (e.g. see Def. 6.2.6).

### 6.4.1 Formalisation

We start with some examples of permutations of the sites of a site graph.

**Exemple 6.4.1** In Fig. 6.24, we consider an agent and some permutations of the state of its sites. More precisely, We consider two kinds of agents  $A$  and  $B$ , and six site identifiers  $u, v, w, x, y$ , and  $z$ . We assume that agents of type  $A$  have the three following binding sites:  $x, y$  and  $z$ , and that agents of type  $B$  have the three following binding sites  $u, v$ , and  $w$ . We assume that agents have no property site. The agent type  $B$  will be used later in Exa. 6.4.5.

More formally, the signature of our model is defined by:

- $= \{A, B\};$
- $= \{u, v, w, x, y, z\};$

- $= \emptyset$ ;
- $(A) = \{x, y, z\}$ ;
- $(B) = \{u, v, w\}$ ;
- $(A) = \emptyset$ ;
- and  $(B) = \emptyset$ .

We consider an agent of type  $A$  with the site  $x$  bound (without any further information about to which site this site is bound to), the site  $y$  free, and with no information about the binding state of the site  $z$ . This agent is denoted by the following expression:

$$A_1(x^-, y, z^?).$$

We apply a first permutation,  $\sigma_1$ , which consists in swapping the binding state of the sites  $x$  and  $z$ . Thus, we get the following agent:

$$A_1(x^?, y, z^-),$$

where the binding state of the site  $x$  is unknown, the site  $y$  is free, and the site  $z$  is bound (without any information about the about to which site this site is bound to).

Then we apply another permutation,  $\sigma_2$ , which consists in by swapping the binding state of the sites  $x$  and  $y$ . Thus, we get the following agent

$$A_1(x, y^?, z^-),$$

where the site  $x$  is free, the binding state of the site  $y$  is unknown, and the site  $z$  is bound (without any information about to which site this site is bound to).

**Definition 6.4.1 (permutations among the sites of a site graph)** Let  $G$  be a site graph. We call a permutation of the sites of the site graph  $G$ , any family  $(f_n)_{n \in [G]}$  of one to one functions, indexed over the agents of the site graph  $G$ , such that, for any agent  $n \in [G]$  of the site graph  $G$ ,  $f_n$  is a one to one function over the set  $([G](n))$  that preserves the set  $([G](n))$  of the binding sites<sup>2</sup> of the agents of type  $[G](n)$  (that is to say that for any site identified  $i \in$ , if  $i \in ([G](n))$ , then  $f_n(i) \in ([G](n))$ ).

The set of all the permutations of the sites of the site graph  $G$  is denoted by  $[G]$ .

In a permutation  $(f_n)_{n \in [G]} \in [G]$  of the sites of a given graph  $G$ , the one to one function  $f_n$  denotes the permutation to be applied to the sites of the agent  $n$  in the site graph  $G$ . Such a permutation is applied by replacing each site  $i$  by the site  $f_n(i)$ , so that the site  $f_n(i)$  of the agent  $n$ , takes the former state of the site  $i$  in the agent  $n$ . This is formalised in the following definition.

**Definition 6.4.2 (action of a permutation of sites on a site graph)** Let  $G$  be a site graph. Let  $(f_n)_{n \in [G]} \in [G]$  be a permutation of the sites of the site graph  $G$ .

Then, the image  $(f_n)_{n \in [G]}$  of the site graph by the permutation  $(f_n)_{n \in [G]}$  of the sites of the site graph, is given by the site graph  $H$  that is defined as follows:

1.  $[H] = [G]$ ;
2.  $[H] = [G]$ ;
3.  $[H] = \{(n, f_n(i)) \mid (n, i) \in [G]\}$ ;
4.  $[H] = \{(p(s), p(s')) \mid (s, s') \in [G]\}$   
where, for any  $s \in [G] \cup \{\neg, -\}$ ,

$$p(s) := \begin{cases} (n, f_n(i)) & \text{if } s \in [G] \text{ and } s = (n, i); \\ s & \text{if } s = \neg \text{ or } s = -; \end{cases}$$

5.  $[H](n, i) = [G](n, f_n^{-1}(i))$ .

---

<sup>2</sup>Since,  $\{([G](n)), ([G](n))\}$  forms a partition of the set  $([G](n))$  and  $f_n$  is one to one, this is equivalent to say that  $f_n$  preserves the set  $([G](n))$  of the property sites of the agents of type  $[G](n)$ .

In Def. 6.4.2, we notice that each site  $(n, i)$  that occurs in  $\square$  and in  $\square$  is replaced by the site  $(n, f_n(i))$ . We deal with the internal states in another way: instead of replacing each site  $(n, i)$  by the site  $(n, f_n(i))$ , we associate each site  $(n, i)$  with the previous internal state of the site  $(n, f_n^{-1}(i))$ , so that, after the permutation, the site  $(n, f_n(i))$  is associated with previous the internal state of the site  $(n, i)$ .

**Example 6.4.2** We consider the same signature, the same site graphs, and the same permutations of sites as Exa. 6.4.1 (e.g. see Fig. 6.24).

The agent that is denoted by the following expression:

$$A_1(x^-, y, z^?),$$

is described by the site graph that is defined by:

- $[G] = \{1\};$
- $[G] = [1 \mapsto A];$
- $[G] = \{(1, x), (1, y), (1, z)\};$
- $[G] = \Delta(\{((1, x), -), ((1, y), \neg)\});$
- $[G] = [(1, x) \mapsto \emptyset, (1, y) \mapsto \emptyset, (1, z) \mapsto \emptyset].$

Now we give the encoding of the permutations of sites  $\sigma_1$ . We denote by  $(f_n)_{n \in \{1\}}$  the permutation of sites  $\sigma_1$ . This permutation swaps the sites  $x$  and  $z$ . Thus, we have:

$$f_1 = [x \mapsto z, y \mapsto y, z \mapsto x].$$

Let us apply the permutation of sites  $\sigma_1$  to the site graph  $G$ . By Def. 6.4.2, we have:

- $[\sigma_1.G] = [G];$
- $[\sigma_1.G] = [G];$
- $[\sigma_1.G] = \{(n, f_n(i)) \mid (n, i) \in [G]\};$
- $[\sigma_1.G] = \{(p(s), (s')) \mid (s, s') \in [G]\}$   
where, for any  $s \in [G] \cup \{\neg, -\},$

$$p(s) := \begin{cases} (n, f_n(i)) & \text{if } s \in [G] \text{ and } s = (n, i); \\ s & \text{if } s \in \{\neg, -\}. \end{cases}$$

- $[\sigma_1.G] = [(n, i) \mapsto [G](n, f_n^{-1}(i))].$

Thus, we have:

- $[\sigma_1.G] = \{1\};$
- $[\sigma_1.G] = [1 \mapsto A];$
- $[\sigma_1.G] = \{(1, f_1(x)), (1, f_1(y)), (1, f_1(z))\};$
- $[\sigma_1.G] = \Delta(\{((1, f_1(x)), -), ((1, f_1(y)), \neg)\});$
- $[\sigma_1.G] = \left[ \begin{array}{l} (1, x) \mapsto [G]((1, f_1^{-1}(x)), \\ (1, y) \mapsto [G]((1, f_1^{-1}(y)), \\ (1, z) \mapsto [G]((1, f_1^{-1}(z)) \end{array} \right].$

And:

- $[\sigma_1.G] = \{1\};$
- $[\sigma_1.G] = [1 \mapsto A];$

- $[\sigma_1.G] = \{(1, z), (1, y), (1, x)\};$
- $[\sigma_1.G] = \Delta(\{((1, z), -), ((1, y), \neg)\});$
- $[\sigma_1.G] = [(1, x) \mapsto \emptyset, (1, y) \mapsto \emptyset, (1, z) \mapsto \emptyset].$

We can check that the site graph  $\sigma_1.G$  denotes the following agent:

$$A_1(x^?, y, z^-),$$

The permutation  $\sigma_2$  of sites swaps the sites  $x$  and  $y$ . We denote by  $(f'_n)_{n \in \{1\}}$  the permutation of sites  $\sigma_2$ . We have:

$$f'_1 = [x \mapsto y, y \mapsto x, z \mapsto z].$$

Permutations of sites can be composed, as formalised in the following definition:

**Definition 6.4.3 (composition)** Let  $\square$  be a site graph. Let  $(f_n)_{n \in [G]}$  and  $(f'_n)_{n \in [G]}$  be two permutations of sites in  $\square$ . We define the product of the permutation of sites  $(f_n)_{n \in [G]}$  and the permutation of sites  $(f'_n)_{n \in [G]}$  as the family  $(f_n f'_n)_{n \in [G]}$  of one to one functions.

The product between  $(f_n)_{n \in [G]}$  and  $(f'_n)_{n \in [G]}$  belongs to the set  $\square$ , we denote it by  $(f_n)_{n \in [G]} \circ (f'_n)_{n \in [G]}$ .

This way, the composition of permutations of sites is obtained by composing the one to one functions component-wise.

**Example 6.4.3** We consider the same signature, the same site graphs, and the same permutations of sites as Exa. 6.4.1 (e.g. see Fig. 6.24).

We compute the composition of the permutations of sites  $\sigma_2$  and  $\sigma_1$ , and we check that we get the expected result.

We denote by  $(f_n)_{n \in \{1\}}$  the permutation  $\sigma_1$ , by  $(f'_n)_{n \in \{1\}}$  the permutation  $\sigma_2$ , and by  $(f''_n)_{n \in \{1\}}$  the product  $\sigma_2 \circ_G \sigma_1$ .

By Def. 6.4.3, the one to one function  $f''_1$  is defined by  $f''_1 \circ f_1$ . Thus,

$$f''_1 = [x \mapsto y, y \mapsto x, z \mapsto z][x \mapsto z, y \mapsto y, z \mapsto x].$$

It follows that:

$$f''_1 = [x \mapsto z, y \mapsto x, z \mapsto y].$$

Thus the permutation of sites  $\sigma_2 \circ \sigma_1$  substitutes the site identifier  $x$  with the site identifier  $z$ , the site identifier  $y$  with the site identifier  $x$ , and the site identifier  $z$  with the site identifier  $y$  (so that the new state of the site  $z$  is the former state of the site  $x$ , the new state of the site  $x$  is the former state of the site  $y$ , and the new state of the site  $y$  is the former state of the site  $z$ ).

For any site graph  $G$ , the permutations of sites that can be applied to the site graph  $G$  form a group for composition, as established in the following proposition.

**Propriété 6.4.1** Let  $G$  be a site graph. The set  $[G]$  forms a group for the composition  $\circ$ . More precisely,

1. the identity element is the family  $(Id_{(\square(n))})_{n \in [G]}$  made, for each agent  $n$  of the site graph  $G$ , of the identity function over the set  $(\square(n))$ ;
2. the inverse of an element  $(f_n)_{n \in [G]}$  is the family  $(f_n^{-1})_{n \in [G]}$  made, for each agent  $n$  of the site graph  $G$ , of the inverse of the function  $f_n$ .

**Example 6.4.4** We consider the same signature, the same site graphs, and the same permutations of sites as Exa. 6.4.1 (e.g. see Fig. 6.24).

Now we compute the inverse of  $\sigma_1$ . We denote by  $(f_n)_{n \in \{1\}}$  the permutation  $\sigma_1$  and by  $(f'''_n)_{n \in \{1\}}$  the permutation  $\sigma_1^{-1}$ .

By applying Prop. 6.4.1.(2), we get that:

$$f'''_1 = (f_1)^{-1}.$$

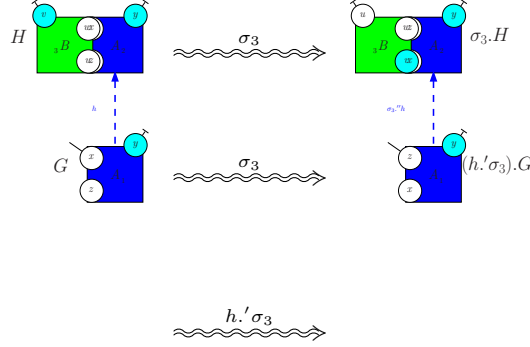


Figure 6.25: Restriction of a permutation of sites to the domain of an embedding.

It follows that:

$$f_1''' = [x \mapsto z, y \mapsto y, z \mapsto x].$$

That is to say that the permutation of sites  $\sigma_1$  is its own inverse. The same way, the permutation of sites  $\sigma_2$  is in own inverse.

Now we compute the permutation of sites  $(\sigma_2 \circ \sigma_1)^{-1}$  in two different ways (as the inverse of the permutation of sites  $\sigma_2 \circ \sigma_1$ , and as the combination of the permutations of sites  $\sigma_1^{-1}$  and  $\sigma_2^{-1}$ , and check that we get the same result.

Let us denote by  $(f'_n)_{n \in \{1\}}$  the permutation of sites  $\sigma_2$  and by  $(f_n''')_{n \in \{1\}}$  the permutation of sites  $(\sigma_2 \circ \sigma_1)^{-1}$ . As the inverse of the permutation of sites  $(\sigma_2 \circ \sigma_1)$ , we get, by applying Prop. 6.4.1.(2), that:

$$f_1'''' = (f_1'')^{-1}.$$

It follows that:

$$f_1'''' = [x \mapsto z, y \mapsto x, z \mapsto y]^{-1}.$$

Thus,

$$f_1'''' = [x \mapsto y, y \mapsto z, z \mapsto x].$$

That is to say that the permutation of sites  $(\sigma_2 \circ \sigma_1)^{-1}$  consists in substituting the site identifier  $x$  with the site identifier  $y$ , the site identifier  $y$  with the site identifier  $z$ , and the site identifier  $z$  with the site identifier  $x$ .

As the composition of the permutation of sites  $\sigma_1^{-1}$  and the permutation of sites  $\sigma_2^{-1}$ , we get by Def. 6.4.3 that:

$$f_1'''' = (f_1)(f_1').$$

It follows that:

$$f_1'''' = [x \mapsto z, y \mapsto y, z \mapsto x][x \mapsto y, y \mapsto x, z \mapsto z].$$

Thus:

$$f_1'''' = [x \mapsto y, y \mapsto z, z \mapsto x].$$

Thus, the two computations are consistent with each other.

Now we define the restriction of a permutation of sites that can be applied to the image of an embedding, into a permutation of sites that can be applied to the domain of this embedding.

**Definition 6.4.4 (permutation propagation)** Let  $h$  be an embedding between two site graphs  $G$  and  $H$ . Let  $\sigma_H \in [H]$  be a permutation of sites that can be applied to the site graph  $H$ .

We define  $h.'\sigma_H$  as the permutation of sites  $(f_{h_n})_{n \in [G]}$ , which can be applied to the site graph  $G$ .



We are left to define the action of a permutation of sites over an embedding. Since permutations of sites preserve the set of agents of the site graphs to which they are applied (only the set of sites and the states of these sites may change), we can define the image of an embedding between two site graphs, as the embedding between the image of these two site graphs, that is induced by the same function over agents.

**Propriété 6.4.2** *Let  $h$  be an embedding between two site graphs  $G$  and  $H$ . Let  $\sigma_H \in [H]$  be a permutation of the sites of the site graph  $H$ .*

*The function between  $[G]$  and  $[H]$  that induces the embedding  $h$ , also induces an embedding between the site graph  $(h.'\sigma_H).G$  and the site graph  $\sigma_H.H$ .*

**Definition 6.4.5 (action of a permutation over an embedding)** *Let  $h$  be an embedding between two site graphs  $G$  and  $H$ . Let  $\sigma_H \in [[H]]$  be a permutation of the sites of the graph  $H$ .*

*We define  $\sigma_H.''h$  as the embedding between the site graph  $(h.'\sigma_H).G$  and  $\sigma_H.H$  that is induced by the same function between  $[G]$  and the site graph  $[H]$  as  $h$ .*

Now we give more details about the example in Fig. 6.25.

**Exemple 6.4.5** *We illustrate Defs. 6.4.4 and 6.4.5 in Fig. 6.25. The signature is the same as in Ex. 6.4.1: there are two kinds of agents  $A$  and  $B$ . The agents of type  $A$  have three binding sites  $x$ ,  $y$  and  $z$ , while the agents of type  $B$  have three binding sites  $u$ ,  $v$ ,  $w$ .*

*We consider an embedding  $h$  between two site graphs  $G$  and  $H$ , where:*

1. *the site graph  $G$  is made of an agent 1 of type  $A$  in which the site  $x$  is bound (with no further information), the site  $y$  is free, and we have no information about the binding state of the site  $z$ .*
2. *the site graph  $H$  is made of two agents: the agent 2 of type  $A$  and the agent 3 of type  $B$ . The site  $y$  of the agent 2 and the site  $v$  of the agent 3 are free. There is a link between the site  $x$  of the agent 2 and the site  $w$  of the agent 1 and another link between the site  $z$  of the agent 2 and the site  $u$  of the agent 3.*
3. *the embedding  $h$  maps the agent of the site graph  $G$  to the agent 2 of the site graph  $H$ .*

*We consider the permutation of sites  $\sigma_3 \in [H]$  that is defined as  $\sigma_3 := (f_n)_{n \in \{2,3\}}$  where:*

- $f_2 = [x \mapsto z, y \mapsto y, z \mapsto x];$
- $f_3 = [u \mapsto v, v \mapsto u, w \mapsto w].$

*Thus the action of the permutation of sites  $\sigma_3$  on the site graph  $G$  swaps the states of the sites  $x$  and  $z$  in agent 2 and the states of the sites  $u$  and  $v$  in agent 3. As a result the site  $y$  of the agent 2 and the site  $u$  of the agent 3 become free. The site  $x$  of agent 2 and the site  $v$  of the agent 3 are now bound together, as well as the site  $z$  of the agent 2 and the site  $w$  of the agent 3.*

*The permutation of sites  $\sigma_3$  can be restricted to the domain of the embedding  $h$ . By Def. 6.4.5, the resulting permutation of sites  $h.'\sigma_3$  is the permutation in  $[G]$  that is defined by  $h.'\sigma_3 = (f'_1)$  where:*

$$f'_1 = f_{h(1)}.$$

*It follows that:*

$$f'_1 = [x \mapsto z, y \mapsto y, z \mapsto x].$$

*Thus, the permutation of sites  $h.'\sigma_3$  swaps the states of the site  $x$  and  $z$  in the unique agent of the site graph  $G$ . It follows that in the unique agent of the site graph  $(h.'\sigma_3).G$ , the binding state of the site  $x$  is not specified, the site  $y$  is free, and the site  $z$  is bound (with no further information). We can check that the embedding  $\sigma_3.''h$  embeds the site graph  $(h.'\sigma_3).G$  into the site graph  $\sigma_3.H$ .*

**Propriété 6.4.3** *The tuple  $\square = ([G], \dots, ', '')$  is a valid set of transformations over site graphs (as defined in Defs. 6.2.1 and 6.2.14).*

*Moreover each embedding is  $\square$ -compatible (as defined in Def. 6.2.15).*

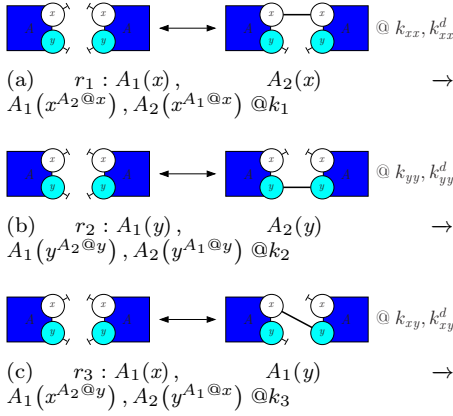


Figure 6.26: Under which conditions are the sites  $x$  and  $y$  equivalent in each occurrence of the protein  $A$  ?

## 6.4.2 Retour sur le cas d'étude

Pour conclure ce chapitre, cette approche est appliquée sur le modèle jouet qui avait été introduit Sec. 6.1.1 page 54.

Dans ce modèle, quitte à intervertir les sites  $x$  et  $y$  dans une ou plusieurs occurrences de la protéine  $A$ , les règles se répartissent en deux catégories : les règles d'association et les règles de dissociation. Ces deux catégories sont dessinées en Fig. 6.27, ainsi que les permutations de sites qui permettent de passer d'une règle à une autre au sein de chaque catégorie. Dans chacune des ces deux catégories, deux règles correspondent à la même, quitte à échanger le rôle des deux occurrences de la protéine  $A$ . Il s'agit des deux règles d'association asymétrique pour lier le site  $x$  d'une des occurrences de la protéine  $A$  au site  $y$  de l'autre occurrence de la protéine  $A$  (les deux règles sur la ligne du milieu en Fig. 6.27(a)) et des deux règles de dissociation asymétriques (les deux règles sur la ligne du milieu en Fig. 6.27(b)). Ainsi, pour que les sites  $x$  et  $y$  aient un rôle équivalent dans le modèle, il suffit que les constantes corrigées des quatre règles d'association soient égales et que celles des quatre règles de dissociation le soient également. Ainsi, en tenant compte, du coefficient de correction des constantes de réaction et des répétitions des règles au sein de chaque catégorie de règles, cela donne les contraintes suivantes :

$$\frac{k_{xx}}{2 \cdot 1} = \frac{k_{yy}}{2 \cdot 1} = \frac{k_{xy}}{2 \cdot 2}$$

et

$$\frac{k_{xx}}{2 \cdot 1} = \frac{k_{yy}}{2 \cdot 1} = \frac{k_{xy}}{1 \cdot 2}.$$

Dans ces fractions, le premier facteur des dénominateurs représente la correction des constantes de réaction alors que le deuxième facteur correspond au nombre de formes de la règle (modulo permutation des agents) qui apparaissent dans les catégories de règles.

Il s'en suit les contraintes suivantes:

$$\begin{cases} k_{xx} = k_{yy} \\ k_{xy} = 2k_{xx} \\ k_{xx}^d = k_{yy}^d \\ k_{xx}^d = k_{xy}^d \end{cases}$$

Sous ces conditions, les sites  $x$  et  $y$  sont équivalents, ce qui induit une bisimulation dans les deux sens sur à la fois sur le système stochastique sous-jacent et le système différentiel sous-jacent.

Cette bisimulation permet de réduire ces systèmes en oubliant la différence entre les sites  $x$  et  $y$ , et ce quel que soit la distribution initiale des états (dans le cadre stochastique) ou les concentrations initiales (dans le cadre différentiel).

Par ailleurs, cette bisimulation permet de caractériser des sous-espaces de distributions d'états (dans le cadre stochastique) et des sous-espaces de concentrations (dans le cas différentiel) stables.

L'examen des différents complexes et de leur relation vis à vis des permutations de sites équivalents permet de trouver un sous-espace stable de concentrations. Quitte à permuter l'état des sites  $x$  et  $y$  dans une ou

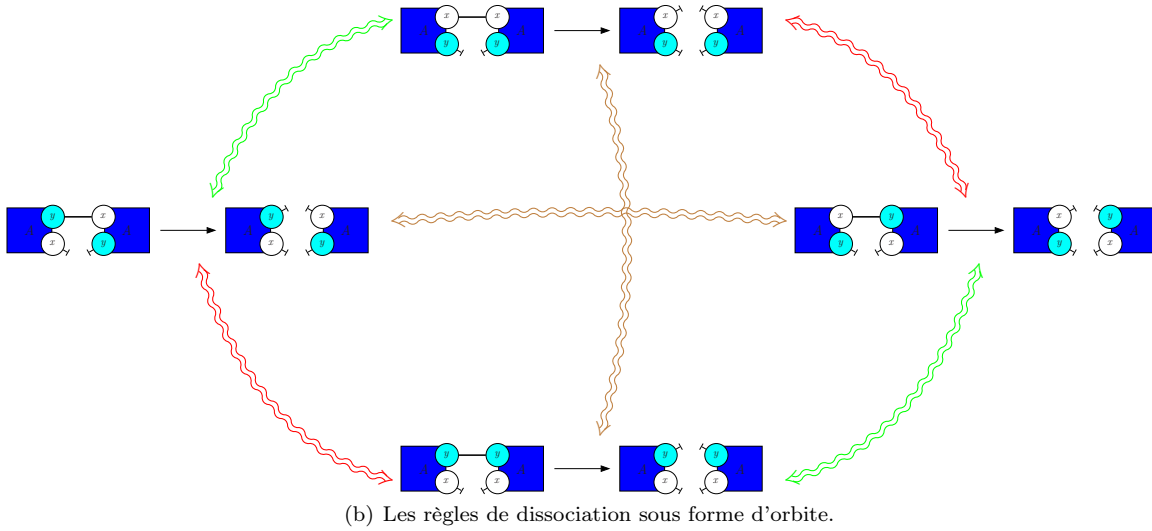
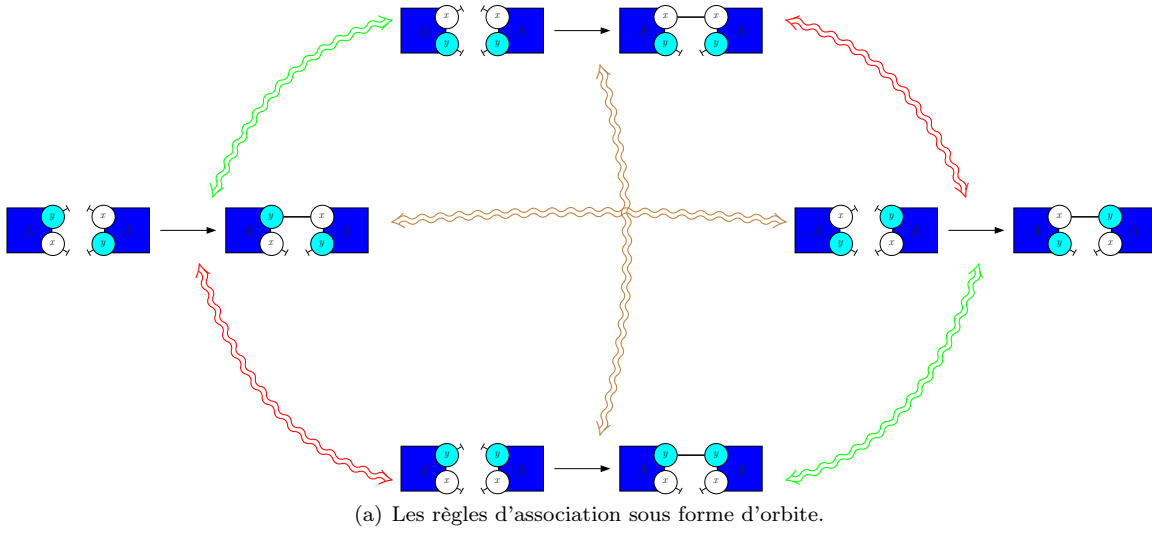


Figure 6.27: Quitte à permuter les sites  $x$  et  $y$  dans certaines occurrences de la protéine  $A$ , les règles peuvent être regroupées en deux catégories. Les règles d'association sont représentées en Fig. 6.27(a), alors que les règles de dissociation sont représentées en Fig. 6.27(b). L'effet des permutations de sites est représenté par des doubles flèches ondulées. Les permutations triviales, qui consiste à ne rien changer ne sont pas représentées. La permutation des sites  $x$  et  $y$  dans l'occurrence gauche de la protéine est représentée en vert ; celle dans l'occurrence de droite est représentée en rouge ; la permutation simultanée des sites  $x$  et  $y$  dans les deux occurrences de la protéine  $A$  est représentée en marron.

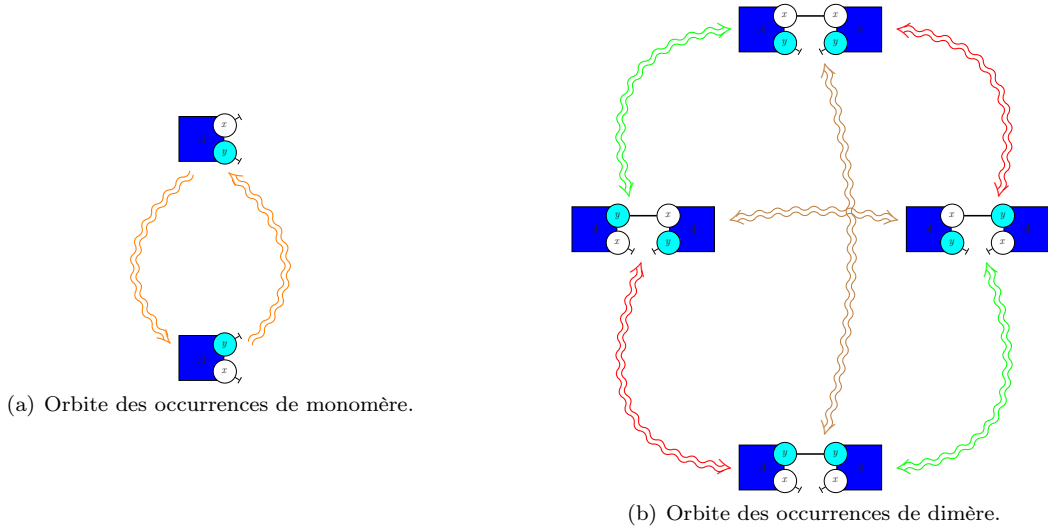


Figure 6.28: Quitte à permuter les sites  $x$  et  $y$  dans les occurrences de la protéine  $A$ , les complexes biochimiques se classent en deux catégories, les monomères (voir en Fig. 6.28(a)) et les dimères (voir en Fig. 6.28(b)). Pour les monomères, la seule transformation possible consiste à permuter les sites de l'unique instance de la protéine (ce qui est dessiné avec une double flèche ondulée orange). Ceci ne change par la conformation de la protéine puisque l'ordre des sites n'a pas d'importance dans le langage Kappa. Pour les dimères, le rôle des sites peuvent être échangés dans l'occurrence gauche (ce qui est dessiné avec une double flèche ondulée verte), dans l'occurrence droite (rouge) ou dans les deux simultanément (noire). Quitte à changer l'ordre des agents et des sites de liaisons, les deux représentations graphiques des dimères asymétriques sont équivalentes du point de vue de la sémantique du langage Kappa.

plusieurs instances de la protéine  $A$ , l'ensemble de tous les complexes biochimiques du modèle jouet se classe en deux catégories : les monomères et les dimères. Celles-ci sont ressinées en Fig. 6.28 sous la forme de deux orbites. Les permutations qui permettent de passer d'un complexe biochimique à un autre au sein de chaque catégorie sont représentées par des doubles flèches ondulées. Plutôt que d'intervertir l'état des sites, l'action de ses permutations a été représenté en échangeant, de manière équivalente, les sites. Étant donné que l'ordre des sites dans une occurrence de protéine n'a pas de signification particulière en Kappa, il existe une seule forme de monomère. Leur concentration n'est donc pas contrainte dans le sous-espace stable de concentrations qui découle de l'équivalence entre les sites  $x$  et  $y$  dans les occurrences de la protéine  $A$ . Il existe en revanche trois formes de dimères : les dimères symétriques avec une liaison entre deux sites  $x$  ; les dimères symétriques avec une liaison entre deux sites  $y$  et les dimères asymétriques avec une liaison entre un site  $x$  d'une occurrence de la protéine  $A$  et le site  $y$  d'une autre occurrence de la protéine  $A$ . Quitte à échanger le rôle des deux occurrences des protéines  $A$ , cette dernière forme de dimère apparaît deux fois. En conséquence, le sous-espace stable de concentrations qui découle de l'équivalence entre les sites  $x$  et  $y$  doit satisfaire les contraintes suivantes : les dimères asymétriques doivent constituer la moitié de la concentration totale en dimère et chacune des deux formes de dimère symétrique doit constituer un quart de la concentration totale en dimère. Il s'agit des mêmes proportions que pour le jeu de "pile" ou "face".

Dans le cadre stochastique, il faut regarder, pour déduire un sous-espace de distributions stable, les catégories d'états du système. Quitte à permuter l'état des sites  $x$  et  $y$ , dans une ou plusieurs occurrences de la protéine  $A$ , deux états sont équivalents s'ils contiennent le même nombre d'occurrence de monomère et le même nombre d'occurrence de dimère.

Reprenant la notation  $i, j, k, l$  pour désigner un état avec  $i$  occurrences de monomère,  $j$  occurrences du dimère formé d'une liaison entre deux sites  $x$ ,  $k$  occurrences du dimère formé d'une liaison entre deux sites  $y$  et  $l$  occurrences du dimère formé d'une liaison entre le site  $x$  d'une occurrence de la protéine  $A$  et le site  $y$  d'une autre occurrence de la protéine  $A$ , deux états  $i', j', k', l'$  et  $i'', j'', k'', l''$  sont donc équivalents lorsque  $i' = i''$  et  $j' + k' + l' = j'' + k'' + l''$ . Reste à calculer les rapports de proportions entre les probabilités de deux tels états dans le sous espace de distributions stable défini par l'équivalence entre les sites  $x$  et  $y$ .

## Chapter 7

# Conclusion

Après un bref passage en revue de l'écosystème Kappa et de l'utilisation de l'interprétation abstraite pour extraire les propriétés des réseaux d'interactions biomoléculaires, le langage Kappa a été présenté plus en détail, ainsi qu'une analyse statique pour détecter parmi un ensemble de motifs d'intérêt lesquels peuvent potentiellement apparaître dans des complexes biochimiques dans une trace 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 complexes 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, ce qui le rend 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 ou ajoutées à 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 (Harmer et collab. 2019 ; Harmer et Oshurko 2019). 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 (Husson et Krivine 2019 ; Husson 2019). 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.

Améliorer l'interactivité des outils (Boutillier 2019 ; Boutillier et collab. 2018b) et un travail sur le rendu visuel des propriétés (Forbes et collab. 2018) sont des enjeux cruciaux pour créer des outils utilisables pour des modélisateurs non experts en langage formel. Il est important d'intéresser un spectre plus large d'utilisateurs. D'une part, c'est une source inépuisable de défis scientifiques qui permettent l'amélioration des outils. D'autre

part, c'est nécessaire pour construire un nombre satisfaisant de modèles.

Les modèles sont de plus en plus grands, que ce soit en nombre de complexes biochimiques différents ou en nombre d'instances des complexes biochimiques. 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 et stochastiques des modèles qui permettront de trouver un encadrement garanti de l'évolution du nombre d'instances ou de la concentration, selon le choix de la sémantique, de motifs d'intérêt au cours du temps, sous la forme de paires de fonctions, elles-mêmes définies comme la solution d'un système différentiel ou comme les trajectoires d'un système stochastique. Des travaux préliminaires ont permis d'intégrer dans un cadre formel des méthodes de troncation de développement formel (Saint-Germain et Feret 2016) ou des méthodes inspirées de la physique comme la tropicalisation (Beica et collab.), 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. 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 complexes biochimiques géants rares et des petits complexes présents en très grand nombre.

## 7.1 Bibliography

- Abou-Jaoudé, W., J. Feret et D. Thieffry. 2015, «Derivation of qualitative dynamical models from biochemical networks», dans *Computational Methods in Systems Biology - 13th International Conference, CMSB 2015, Nantes, France, September 16-18, 2015, Proceedings, Lecture Notes in Computer Science*, vol. 9308, édité par O. F. Roux et J. Bourdon, Springer, p. 195–207. URL [https://doi.org/10.1007/978-3-319-23401-4\\_17](https://doi.org/10.1007/978-3-319-23401-4_17).
- Abou-Jaoudé, W., D. Thieffry et J. Feret. 2016, «Formal derivation of qualitative dynamical models from biochemical networks», *Biosystems*, vol. 149, p. 70–112. URL <https://doi.org/10.1016/j.biosystems.2016.09.001>.
- Allart, E., J. Niehren et C. Versari. 2019, «Computing difference abstractions of metabolic networks under kinetic constraints», dans *Computational Methods in Systems Biology - 17th International Conference, CMSB 2019, Trieste, Italy, September 18-20, 2019, Proceedings, Lecture Notes in Computer Science*, vol. 11773, édité par L. Bortolussi et G. Sanguinetti, Springer, p. 266–285. URL [https://doi.org/10.1007/978-3-030-31304-3\\_14](https://doi.org/10.1007/978-3-030-31304-3_14).
- Andersen, J. L., C. Flamm, D. Merkle et P. F. Stadler. 2016, «A software package for chemically inspired graph transformation», dans *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, Lecture Notes in Computer Science*, vol. 9761, édité par R. Echahed et M. Minas, Springer, p. 73–88. URL [https://doi.org/10.1007/978-3-319-40530-8\\_5](https://doi.org/10.1007/978-3-319-40530-8_5).
- Andrei, O. et H. Kirchner. 2008, «A rewriting calculus for multigraphs with ports», *Electr. Notes Theor. Comput. Sci.*, vol. 219, p. 67–82. URL <https://doi.org/10.1016/j.entcs.2008.10.035>.
- Behr, N. et J. Krivine. 2019, «Compositionality of rewriting rules with conditions», *CoRR*, vol. abs/1904.09322. URL <http://arxiv.org/abs/1904.09322>.
- Beica, A., J. Feret et T. Petrov. «Tropical abstraction of biochemical reaction networks with guarantees», *Electr. Notes Theor. Comput. Sci.*
- Blanchet, B., P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Miné, D. Monniaux et X. Rival. 2003, «A static analyzer for large safety-critical software», dans *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation 2003, San Diego, California, USA, June 9-11, 2003*, édité par R. Cytron et R. Gupta, p. 196–207.
- Blinov, M. L., J. R. Faeder, B. Goldstein et W. S. Hlavacek. 2004a, «Bionetgen: software for rule-based modeling of signal transduction based on the interactions of molecular domains», *Bioinformatics*, vol. 20, 17.
- Blinov, M. L., J. R. Faeder, B. Goldstein et W. S. Hlavacek. 2004b, «Bionetgen: software for rule-based modeling of signal transduction based on the interactions of molecular domains», *Bioinformatics*, vol. 20, 17, p. 3289–3291.

- Blinov, M. L., J. R. Faeder, B. Goldstein et W. S. Hlavacek. 2006, «A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity», *BioSystems*, vol. 83, p. 136–151.
- Bodei, C., L. Brodo, R. Gori, D. Hermith et F. Levi. 2015, «A global occurrence counting analysis for brane calculi», dans *Logic-Based Program Synthesis and Transformation - 25th International Symposium, LOPSTR 2015, Siena, Italy, July 13-15, 2015. Revised Selected Papers, Lecture Notes in Computer Science*, vol. 9527, édité par M. Falaschi, Springer, p. 179–200. URL [https://doi.org/10.1007/978-3-319-27436-2\\_11](https://doi.org/10.1007/978-3-319-27436-2_11).
- Bodei, C., P. Degano, F. Nielson et H. R. Nielson. 1998, «Control flow analysis for the pi-calculus», dans *CONCUR '98: Concurrency Theory, 9th International Conference, Nice, France, September 8-11, 1998, Proceedings, Lecture Notes in Computer Science*, vol. 1466, édité par D. Sangiorgi et R. de Simone, Springer, p. 84–98. URL <https://doi.org/10.1007/BFb0055617>.
- Borisov, N. M., N. I. Markevich, B. N. Kholodenko et E. D. Gilles. 2005, «Signaling through receptors and scaffolds: Independent interactions reduce combinatorial complexity», *Biophysical Journal*, vol. 89.
- Boutillier, P. 2019, «The kappa simulator made interactive», dans *Computational Methods in Systems Biology - 17th International Conference, CMSB 2019, Trieste, Italy, September 18-20, 2019, Proceedings, Lecture Notes in Computer Science*, vol. 11773, édité par L. Bortolussi et G. Sanguinetti, Springer, p. 296–301. URL [https://doi.org/10.1007/978-3-030-31304-3\\_16](https://doi.org/10.1007/978-3-030-31304-3_16).
- Boutillier, P., F. Camporesi, J. Coquet, J. Feret, K. Q. Lý, N. Théret et P. Vignat. 2018a, «Kasa: A static analyzer for kappa», dans *Computational Methods in Systems Biology - 16th International Conference, CMSB 2018, Brno, Czech Republic, September 12-14, 2018, Proceedings, Lecture Notes in Computer Science*, vol. 11095, édité par M. Ceska et D. Safránek, Springer, p. 285–291. URL [https://doi.org/10.1007/978-3-319-99429-1\\_17](https://doi.org/10.1007/978-3-319-99429-1_17).
- Boutillier, P., T. Ehrhard et J. Krivine. 2017, «Incremental update for graph rewriting», dans *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, Lecture Notes in Computer Science*, vol. 10201, édité par H. Yang, Springer, p. 201–228. URL [https://doi.org/10.1007/978-3-662-54434-1\\_8](https://doi.org/10.1007/978-3-662-54434-1_8).
- Boutillier, P., A. Faure de Pebeyre et J. Feret. «Proving the absence of unbounded polymers in rule-based models», dans *Nine International Workshop on Static Analysis and Systems Biology (SASB'18)*, ENTCS, elsevier. To appear.
- Boutillier, P., M. Maasha, X. Li, H. F. Medina-Abarca, J. Krivine, J. Feret, I. Cristescu, A. G. Forbes et W. Fontana. 2018b, «The kappa platform for rule-based modeling», *Bioinformatics*, vol. 34, 13, p. i583–i592. URL <https://doi.org/10.1093/bioinformatics/bty272>.
- Brightman, F. A. et D. A. Fell. 2000, «Differential feedback regulation of the mapk cascade underlies the quantitative differences in egf and ngf signalling in pc12 cells», *FEBS Letters*, vol. 482, 3, p. 169–174.
- Buchholz, P. 1994, «Exact and ordinary lumpability in finite markov chains», *Journal of Applied Probability*, vol. 31, 1, p. 59–75.
- Buchholz, P. 2008, «Bisimulation relations for weighted automata», *TCS*, vol. 393, 1-3, p. 109–123.
- Camporesi, F., J. Feret, H. Koepl et T. Petrov. 2010, «Combining model reductions», dans *MFPSXXVI: Post-proceedings of the 26th Conference on the Mathematical Foundations of Programming Semantics, Electronic Notes in Theoretical Computer Science*, vol. 265, Elsevier Science Publishers, p. 73–96.
- Camporesi, F., J. Feret et K. Q. Lý. 2017, «Kade: A tool to compile kappa rules into (reduced) ODE models», dans *Computational Methods in Systems Biology - 15th International Conference, CMSB 2017, Darmstadt, Germany, September 27-29, 2017, Proceedings, Lecture Notes in Computer Science*, vol. 10545, édité par J. Feret et H. Koepl, Springer, p. 291–299. URL [https://doi.org/10.1007/978-3-319-67471-1\\_18](https://doi.org/10.1007/978-3-319-67471-1_18).
- Cardelli, L. 2004, «Brane calculi», dans *Computational Methods in Systems Biology, International Conference, CMSB 2004, Paris, France, May 26-28, 2004, Revised Selected Papers, Lecture Notes in Computer Science*, vol. 3082, édité par V. Danos et V. Schächter, Springer, p. 257–278. URL [https://doi.org/10.1007/978-3-540-25974-9\\_24](https://doi.org/10.1007/978-3-540-25974-9_24).
- Cardelli, L. et A. D. Gordon. 1998, «Mobile ambients», dans *Foundations of Software Science and Computation Structure, First International Conference, FoSSaCS'98, Held as Part of the European Joint Conferences on the Theory and Practice of Software, ETAPS'98, Lisbon, Portugal, March 28 - April 4, 1998, Proceedings, Lecture Notes in Computer Science*, vol. 1378, édité par M. Nivat, Springer, p. 140–155. URL [https://doi.org/10.1007/978-3-540-60871-1\\_11](https://doi.org/10.1007/978-3-540-60871-1_11).

org/10.1007/BFb0053547.

- Cardelli, L. et A. D. Gordon. 2000, «Mobile ambients», *Theor. Comput. Sci.*, vol. 240, 1, doi:10.1016/S0304-3975(99)00231-5, p. 177–213. URL [https://doi.org/10.1016/S0304-3975\(99\)00231-5](https://doi.org/10.1016/S0304-3975(99)00231-5).
- Cardelli, L., M. Tribastone, M. Tschaikowski et A. Vandin. 2015, «Forward and backward bisimulations for chemical reaction networks», dans *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1-4, 2015*, p. 226–239.
- Cardelli, L., M. Tribastone, M. Tschaikowski et A. Vandin. 2019, «Symbolic computation of differential equivalences», *Theor. Comput. Sci.*, vol. 777, doi:10.1016/j.tcs.2019.03.018, p. 132–154. URL <https://doi.org/10.1016/j.tcs.2019.03.018>.
- Ciocchetta, F. et J. Hillston. 2009, «Bio-PEPA: A framework for the modelling and analysis of biological systems», *Theoretical Computer Science*, vol. 410, 33 – 34, p. 3065 – 3084. Concurrent Systems Biology: To Nadia Busi (1968–2007).
- Cohen, P. R. 2015, «DARPA's big mechanism program», *Physical Biology*, vol. 12, 4, doi:10.1088/1478-3975/12/4/045008, p. 045 008. URL <https://doi.org/10.1088/1478-3975/12/4/045008>.
- Conzelmann, H., J. Saez-Rodriguez, T. Sauter, B. N. Kholodenko et E. D. Gilles. 2006, «A domain-oriented approach to the reduction of combinatorial complexity in signal transduction networks», *BMC Bioinformatics*, vol. 7.
- Cook, B., J. Fisher, E. Krepska et N. Piterman. 2011, «Proving stabilization of biological systems», dans *Verification, Model Checking, and Abstract Interpretation - 12th International Conference, VMCAI 2011, Austin, TX, USA, January 23-25, 2011. Proceedings*, vol. 6538, édité par R. Jhala et D. A. Schmidt, Springer, p. 134–149. URL [https://doi.org/10.1007/978-3-642-18275-4\\_11](https://doi.org/10.1007/978-3-642-18275-4_11).
- Corradini, A., T. Heindel, F. Hermann et B. König. 2006, «Sesqui-pushout rewriting», dans *Graph Transformations, Third International Conference, ICGT 2006, Natal, Rio Grande do Norte, Brazil, September 17-23, 2006, Proceedings, Lecture Notes in Computer Science*, vol. 4178, édité par A. Corradini, H. Ehrig, U. Montanari, L. Ribeiro et G. Rozenberg, Springer, p. 30–45. URL [https://doi.org/10.1007/11841883\\_4](https://doi.org/10.1007/11841883_4).
- Corradini, A., U. Montanari, F. Rossi, H. Ehrig, R. Heckel et M. Löwe. 1997, «Algebraic approaches to graph transformation - part I: basic concepts and double pushout approach», dans *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*, édité par G. Rozenberg, World Scientific, p. 163–246.
- Cousot, P. 1999, «The calculational design of a generic abstract interpreter», dans *Calculational System Design*, édité par M. Broy et R. Steinbrüggen, NATO ASI Series F. IOS Press, Amsterdam.
- Cousot, P. 2002, «Constructive design of a hierarchy of semantics of a transition system by abstract interpretation», *Theoretical Computer Science*, vol. 277, 1–2, p. 47–103.
- Cousot, P. et R. Cousot. 1977, «Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints», dans *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*, édité par R. M. Graham, M. A. Harrison et R. Sethi, ACM, p. 238–252. URL <https://doi.org/10.1145/512950.512973>.
- Cousot, P. et R. Cousot. 1979, «Systematic design of program analysis frameworks», dans *Conference Record of the Sixth Annual ACM Symposium on Principles of Programming Languages, San Antonio, Texas, USA, January 1979*, édité par A. V. Aho, S. N. Zilles et B. K. Rosen, ACM Press, p. 269–282. URL <https://doi.org/10.1145/567752.567778>.
- Damgaard, T. C., E. Højsgaard et J. Krivine. 2012, «Formal cellular machinery», *Electr. Notes Theor. Comput. Sci.*, vol. 284, doi:10.1016/j.entcs.2012.05.015, p. 55–74. URL <https://doi.org/10.1016/j.entcs.2012.05.015>.
- Damm, W. et D. Harel. 2001, «LSCs: Breathing life into message sequence charts», *Formal Methods in System Design*, vol. 19, 1, doi:10.1023/A:1011227529550, p. 45–80, ISSN 1572-8102. URL <http://dx.doi.org/10.1023/A:1011227529550>.
- Danos, V., J. Feret, W. Fontana, R. Harmer, J. Hayman, J. Krivine, C. D. Thompson-Walsh et G. Winskel. 2012, «Graphs, rewriting and pathway reconstruction for rule-based models», dans *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India, LIPIcs*, vol. 18, édité par D. D'Souza, T. Kavitha et J. Radhakrishnan, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, p. 276–288. URL <https://doi.org/10.4230/LIPIcs.FSTTCS>.



2012.276.

- Danos, V., J. Feret, W. Fontana, R. Harmer et J. Krivine. 2007a, «Rule-based modelling of cellular signalling», dans *CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings, Lecture Notes in Computer Science*, vol. 4703, édité par L. Caires et V. T. Vasconcelos, Springer, p. 17–41. URL [https://doi.org/10.1007/978-3-540-74407-8\\_3](https://doi.org/10.1007/978-3-540-74407-8_3).
- Danos, V., J. Feret, W. Fontana, R. Harmer et J. Krivine. 2010, «Abstracting the differential semantics of rule-based models: Exact and automated model reduction», dans *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, IEEE Computer Society, ISBN 978-0-7695-4114-3, p. 362–381. URL <https://doi.org/10.1109/LICS.2010.44>.
- Danos, V., J. Feret, W. Fontana et J. Krivine. 2007b, «Scalable simulation of cellular signaling networks», dans *Programming Languages and Systems, 5th Asian Symposium, APLAS 2007, Singapore, November 29-December 1, 2007, Proceedings, Lecture Notes in Computer Science*, vol. 4807, édité par Z. Shao, Springer, p. 139–157. URL [https://doi.org/10.1007/978-3-540-76637-7\\_10](https://doi.org/10.1007/978-3-540-76637-7_10).
- Danos, V., J. Feret, W. Fontana et J. Krivine. 2007c, «Scalable simulation of cellular signaling networks, invited paper», dans *APLAS'07: Proceedings of the Fifth Asian Symposium on Programming Systems, Lecture Notes in Computer Science*, vol. 4807, Springer, Berlin, Germany, p. 139–157.
- Danos, V., J. Feret, W. Fontana et J. Krivine. 2008, «Abstract interpretation of cellular signalling networks», dans *Verification, Model Checking, and Abstract Interpretation, 9th International Conference, VMCAI 2008, San Francisco, USA, January 7-9, 2008, Proceedings, Lecture Notes in Computer Science*, vol. 4905, édité par F. Logozzo, D. A. Peled et L. D. Zuck, Springer, p. 83–97. URL <https://doi.org/10.1007/978-3-540-78163-9>.
- Danos, V., R. Honorato-Zimmer, S. Jaramillo-Riveri et S. Stucki. 2015, «Rigid geometric constraints for Kappa models», dans *SASB'12: PostProceedings of the 3rd International Workshop on Static Analysis and Systems Biology, ENTCS*, vol. 313, Elsevier, p. 23–46.
- Danos, V. et C. Laneve. 2003, «Graphs for core molecular biology», dans *Computational Methods in Systems Biology, First International Workshop, CMSB 2003, Roverto, Italy, February 24-26, 2003, Proceedings, Lecture Notes in Computer Science*, vol. 2602, édité par C. Priami, Springer, p. 34–46. URL [https://doi.org/10.1007/3-540-36481-1\\_4](https://doi.org/10.1007/3-540-36481-1_4).
- Danos, V. et C. Laneve. 2004, «Formal molecular biology», *Theoretical Computer Science*, vol. 325, 1, doi: <http://dx.doi.org/10.1016/j.tcs.2004.03.065>, p. 69 – 110, ISSN 0304-3975. URL <http://www.sciencedirect.com/science/article/pii/S0304397504002336>, computational Systems Biology.
- Danos, V. et S. Pradalier. 2004, «Projective brane calculus», dans *Computational Methods in Systems Biology, International Conference, CMSB 2004, Paris, France, May 26-28, 2004, Revised Selected Papers, Lecture Notes in Computer Science*, vol. 3082, édité par V. Danos et V. Schächter, Springer, p. 134–148. URL [https://doi.org/10.1007/978-3-540-25974-9\\_11](https://doi.org/10.1007/978-3-540-25974-9_11).
- Ded, T., D. Safránek, M. Troják, M. Klement, J. Salagovic et L. Brim. 2016, «Formal biochemical space with semantics in kappa and BNGL», *Electr. Notes Theor. Comput. Sci.*, vol. 326, p. 27–49. URL <https://doi.org/10.1016/j.entcs.2016.09.017>.
- Dematté, L., C. Priami et A. Romanel. 2008, «The blenx language: A tutorial», dans *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*, édité par M. Bernardo, P. Degano et G. Zavattaro, Springer Berlin Heidelberg, Berlin, Heidelberg, p. 313–365. URL [http://dx.doi.org/10.1007/978-3-540-68894-5\\$9](http://dx.doi.org/10.1007/978-3-540-68894-5$9).
- Faeder, J. R., M. L. Blinov, B. Goldstein et W. S. Hlavacek. 2005, «Rule-based modeling of biochemical networks», *Complexity*, vol. 10, 4, doi:10.1002/cplx.20074, p. 22–41, ISSN 1099-0526. URL <http://dx.doi.org/10.1002/cplx.20074>.
- Fähndrich, M. et F. Logozzo. 2010, «Static contract checking with abstract interpretation», dans *Formal Verification of Object-Oriented Software - International Conference, FoVeOOS 2010, Paris, France, June 28-30, 2010, Revised Selected Papers, LNCS*, vol. 6528, édité par B. Beckert et C. Marché, Springer, p. 10–30.
- Feinberg, M. «Lectures on chemical reaction networks», Notes of lectures given at the Mathematics Research Centre, University of Wisconsin, in 1979.
- Feret, J. 2000, «Confidentiality analysis of mobile systems», dans *Static Analysis, 7th International Sympo-*

- sium, *SAS 2000*, Santa Barbara, CA, USA, June 29 - July 1, 2000, *Proceedings, Lecture Notes in Computer Science*, vol. 1824, édité par J. Palsberg, Springer, p. 135–154. URL [https://doi.org/10.1007/978-3-540-45099-3\\_8](https://doi.org/10.1007/978-3-540-45099-3_8).
- Feret, J. 2001, «Occurrence counting analysis for the pi-calculus», *Electr. Notes Theor. Comput. Sci.*, vol. 39, 2, doi:10.1016/S1571-0661(05)01155-2, p. 1–18. URL [https://doi.org/10.1016/S1571-0661\(05\)01155-2](https://doi.org/10.1016/S1571-0661(05)01155-2).
- Feret, J. 2014, «An algebraic approach for inferring and using symmetries in rule-based models.», dans *5th International Workshop on Static Analysis and Systems Biology (SASB 2014)*, édité par L. Paulevé et H. Koepl, ENTCS, Elsevier. To appear.
- Feret, J. 2015, «An algebraic approach for inferring and using symmetries in rule-based models», *Electr. Notes Theor. Comput. Sci.*, vol. 316, p. 45–65. URL <https://doi.org/10.1016/j.entcs.2015.06.010>.
- Feret, J., V. Danos, J. Krivine, R. Harmer et W. Fontana. 2009, «Internal coarse-graining of molecular systems», *PNAS*.
- Feret, J., H. Koepl et T. Petrov. 2013, «Stochastic fragments: A framework for the exact reduction of the stochastic semantics of rule-based models», *Int. J. Software and Informatics*, vol. 7, 4, p. 527–604. URL [http://www.ijsi.org/ch/reader/view\\_abstract.aspx?file\\_no=i173](http://www.ijsi.org/ch/reader/view_abstract.aspx?file_no=i173).
- Feret, J. et K. Q. Lý. 2016, «Local traces: An over-approximation of the behaviour of the proteins in rule-based models», dans *Computational Methods in Systems Biology - 14th International Conference, CMSB 2016, Cambridge, UK, September 21-23, 2016, Proceedings, Lecture Notes in Computer Science*, vol. 9859, édité par E. Bartocci, P. Liò et N. Paoletti, Springer, p. 116–131. URL [https://doi.org/10.1007/978-3-319-45177-0\\_8](https://doi.org/10.1007/978-3-319-45177-0_8).
- Feret, J. et K. Q. Ly. 2018, «Local traces: An over-approximation of the behavior of the proteins in rule-based models», *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, 4, doi:10.1109/TCBB.2018.2812195, p. 1124–1137, ISSN 1545-5963. URL [doi.ieeecomputersociety.org/10.1109/TCBB.2018.2812195](https://doi.ieeecomputersociety.org/10.1109/TCBB.2018.2812195).
- Feret, J. et K. Q. Lý. 2018, «Reachability analysis via orthogonal sets of patterns», *Electr. Notes Theor. Comput. Sci.*, vol. 335, p. 27–48. URL <https://doi.org/10.1016/j.entcs.2018.03.007>.
- Folschette, M., L. Paulevé, M. Magnin et O. F. Roux. 2013, «Under-approximation of reachability in multivalued asynchronous networks», *Electr. Notes Theor. Comput. Sci.*, vol. 299, doi:10.1016/j.entcs.2013.11.004, p. 33–51. URL <https://doi.org/10.1016/j.entcs.2013.11.004>.
- Forbes, A. G., A. Burks, K. Lee, X. Li, P. Boutillier, J. Krivine et W. Fontana. 2018, «Dynamic influence networks for rule-based models», *IEEE Trans. Vis. Comput. Graph.*, vol. 24, 1, p. 184–194. URL <https://doi.org/10.1109/TVCG.2017.2745280>.
- Gao, Q., F. Liu, D. Gilbert, M. Heiner et D. Tree. 2011, «A multiscale approach to modelling planar cell polarity in drosophila wing using hierarchically coloured petri nets», dans *Proceedings of the 9th International Conference on Computational Methods in Systems Biology, CMSB '11*, ACM, New York, NY, USA, ISBN 978-1-4503-0817-5, p. 209–218, doi:10.1145/2037509.2037538. URL <http://doi.acm.org/10.1145/2037509.2037538>.
- Gay, S., F. Fages, T. Martinez, S. Soliman et C. Solnon. 2014, «On the subgraph epimorphism problem», *Discrete Applied Mathematics*, vol. 162, doi:10.1016/j.dam.2013.08.008, p. 214–228. URL <http://dx.doi.org/10.1016/j.dam.2013.08.008>.
- Gillespie, C. S. 2009, «Moment-closure approximations for mass-action models», *IET systems biology*, vol. 3, 1, p. 52–58.
- Gillespie, D. T. 1977, «Exact stochastic simulation of coupled chemical reactions», *The Journal of Physical Chemistry*, vol. 81, 25, p. 2340–2361.
- Gori, R. et F. Levi. 2006, «An analysis for proving temporal properties of biological systems», dans *Programming Languages and Systems, 4th Asian Symposium, APLAS 2006, Sydney, Australia, November 8-10, 2006, Proceedings, Lecture Notes in Computer Science*, vol. 4279, édité par N. Kobayashi, Springer, p. 234–252. URL [https://doi.org/10.1007/11924661\\_15](https://doi.org/10.1007/11924661_15).
- Grosu, R., G. Batt, F. H. Fenton, J. Glimm, C. L. Guernic, S. A. Smolka et E. Bartocci. 2011, «From cardiac cells to genetic regulatory networks», dans *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings, Lecture Notes in Computer Science*, vol. 6806, édité par G. Gopalakrishnan et S. Qadeer, Springer, p. 396–411. URL [https://doi.org/10.1007/978-3-642-22279-0\\_27](https://doi.org/10.1007/978-3-642-22279-0_27).

978-3-642-22110-1\_31.

- Gyori, B., J. Bachman, K. Subramanian, J. Muhlich, L. Galescu et P. Sorger. 2017, «From word models to executable models of signaling networks using automated assembly», *Molecular Systems Biology*, vol. 13.
- Halpern, J. Y. et J. Pearl. 2013, «Causes and explanations: A structural-model approach — part 1: Causes», *CoRR*, vol. abs/1301.2275. URL <http://arxiv.org/abs/1301.2275>.
- Harmer, R. 2009, «Rule-based modelling and tunable resolution», dans *DCM'09: Proceedings Fifth Workshop on Developments in Computational Models—Computational Models From Nature, EPTCS*, vol. 9, p. 65–72.
- Harmer, R., Y. L. Cornec, S. Légaré et E. Oshurko. 2019, «Bio-curation for cellular signalling: The KAMI project», *IEEE/ACM Trans. Comput. Biology Bioinform.*, vol. 16, 5, p. 1562–1573. URL <https://doi.org/10.1109/TCBB.2019.2906164>.
- Harmer, R., V. Danos, J. Feret, J. Krivine et W. Fontana. 2010, «Intrinsic information carriers in combinatorial dynamical systems», *Chaos*, vol. 20. URL <http://link.aip.org/link/CHAOEH/v20/i3/p037108/s1>.
- Harmer, R. et E. Oshurko. 2019, «Kamistudio: An environment for biocuration of cellular signalling knowledge», dans *Computational Methods in Systems Biology - 17th International Conference, CMSB 2019, Trieste, Italy, September 18-20, 2019, Proceedings, Lecture Notes in Computer Science*, vol. 11773, édité par L. Bortolussi et G. Sanguinetti, Springer, p. 322–328. URL [https://doi.org/10.1007/978-3-030-31304-3\\_20](https://doi.org/10.1007/978-3-030-31304-3_20).
- Heiner, M. et I. Koch. 2004, «Petri net based model validation in systems biology», dans *Applications and Theory of Petri Nets 2004: 25th International Conference, ICATPN 2004, Bologna, Italy, June 21–25, 2004. Proceedings*, édité par J. Cortadella et W. Reisig, Springer Berlin Heidelberg, Berlin, Heidelberg, p. 216–237. URL [http://dx.doi.org/10.1007/978-3-540-27793-4\\_13](http://dx.doi.org/10.1007/978-3-540-27793-4_13).
- Helms, T., T. Warnke, C. Maus et A. M. Uhrmacher. 2017, «Semantics and efficient simulation algorithms of an expressive multilevel modeling language», *ACM Trans. Model. Comput. Simul.*, vol. 27, 2, doi:10.1145/2998499, p. 8:1–8:25. URL <https://doi.org/10.1145/2998499>.
- Husson, A. 2019, *Logical foundations of a modelling assistant for molecular biology*, thèse de doctorat, Université de Paris, France. URL [http://ahusson.fr/husson\\_manuscript\\_electronic.pdf](http://ahusson.fr/husson_manuscript_electronic.pdf).
- Husson, A. et J. Krivine. 2019, «A tractable logic for molecular biology», dans *Proceedings 35th International Conference on Logic Programming (Technical Communications), ICLP 2019 Technical Communications, Las Cruces, NM, USA, September 20-25, 2019., EPTCS*, vol. 306, édité par B. Bogaerts, E. Erdem, P. Fodor, A. Formisano, G. Ianni, D. Inclezan, G. Vidal, A. Villanueva, M. D. Vos et F. Yang, p. 101–113. URL <https://doi.org/10.4204/EPTCS.306.17>.
- John, M., C. Lhoussaine, J. Niehren et C. Versari. 2011a, «Biochemical reaction rules with constraints», dans *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, Lecture Notes in Computer Science*, vol. 6602, Springer, p. 338–357. URL <http://dx.doi.org/10.1007/978-3-642-19718-5>.
- John, M., C. Lhoussaine, J. Niehren et C. Versari. 2011b, «Biochemical reaction rules with constraints», dans *ESOP'11: Proceedings of the 20th European Symposium on Programming, Lecture Notes in Computer Science*, vol. 6602, p. 338–357.
- John, M., M. Nebut et J. Niehren. 2013, «Knockout prediction for reaction networks with partial kinetic information», dans *Verification, Model Checking, and Abstract Interpretation, 14th International Conference, VMCAI 2013, Rome, Italy, January 20-22, 2013. Proceedings, Lecture Notes in Computer Science*, vol. 7737, édité par R. Giacobazzi, J. Berdine et I. Mastroeni, Springer, p. 355–374. URL [https://doi.org/10.1007/978-3-642-35873-9\\_22](https://doi.org/10.1007/978-3-642-35873-9_22).
- Kahramanogullari, O. et L. Cardelli. 2013, «An intuitive modelling interface for systems biology», *Int. J. Software and Informatics*, vol. 7, 4, p. 655–674. URL [http://www.ijsi.org/ch/reader/view\\_abstract.aspx?file\\_no=i177](http://www.ijsi.org/ch/reader/view_abstract.aspx?file_no=i177).
- Klärner, H., A. Bockmayr et H. Siebert. 2015, «Computing maximal and minimal trap spaces of boolean networks», *Natural Computing*, vol. 14, 4, doi:10.1007/s11047-015-9520-7, p. 535–544. URL <https://doi.org/10.1007/s11047-015-9520-7>.
- Köhler, A., J. Krivine et J. Vidmar. 2014, «A rule-based model of base excision repair», dans *Computational Methods in Systems Biology - 12th International Conference, CMSB 2014, Manchester, UK, November 17-19, 2014, Proceedings, Lecture Notes in Computer Science*, vol. 8859, édité par P. Mendes, J. O. Dada et

- K. Smallbone, Springer, p. 173–195. URL [https://doi.org/10.1007/978-3-319-12982-2\\_13](https://doi.org/10.1007/978-3-319-12982-2_13).
- Kolcák, J., D. Safránek, S. Haar et L. Paulevé. 2019, «Parameter space abstraction and unfolding semantics of discrete regulatory networks», *Theor. Comput. Sci.*, vol. 765, doi:10.1016/j.tcs.2018.03.009, p. 120–144. URL <https://doi.org/10.1016/j.tcs.2018.03.009>.
- Kwiatkowska, M. Z., G. Norman et D. Parker. 2011, «PRISM 4.0: Verification of probabilistic real-time systems», dans *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings, Lecture Notes in Computer Science*, vol. 6806, édité par G. Gopalakrishnan et S. Qadeer, Springer, p. 585–591. URL [https://doi.org/10.1007/978-3-642-22110-1\\_47](https://doi.org/10.1007/978-3-642-22110-1_47).
- Laurent, J., J. Yang et W. Fontana. 2018, «Counterfactual resimulation for causal analysis of rule-based models», dans *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, édité par J. Lang, ijcai.org, p. 1882–1890. URL <https://doi.org/10.24963/ijcai.2018/260>.
- Löwe, M. 1993, «Algebraic approach to single-pushout graph transformation», *Theor. Comput. Sci.*, vol. 109, 1&2, p. 181–224. URL [https://doi.org/10.1016/0304-3975\(93\)90068-5](https://doi.org/10.1016/0304-3975(93)90068-5).
- Mazurkiewicz, A. W. 1984, «Traces, histories, graphs: Instances of a process monoid», dans *Mathematical Foundations of Computer Science 1984, Praha, Czechoslovakia, September 3-7, 1984, Proceedings, Lecture Notes in Computer Science*, vol. 176, édité par M. Chytil et V. Koubek, Springer, p. 115–133. URL <https://doi.org/10.1007/BFb0030293>.
- McQuarrie, D. A. 1967, «Stochastic approach to chemical kinetics», *Journal of Applied Probability*, vol. 4, 3, p. pp. 413–478, ISSN 00219002. URL <http://www.jstor.org/stable/3212214>.
- Nielson, H. R. et F. Nielson. 2000, «Shape analysis for mobile ambients», dans *POPL 2000, Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Boston, Massachusetts, USA, January 19-21, 2000*, édité par M. N. Wegman et T. W. Reps, ACM, p. 142–154. URL <https://doi.org/10.1145/325694.325711>.
- Paulevé, L., M. Magnin et O. F. Roux. 2011, «Abstract interpretation of dynamics of biological regulatory networks», *Electr. Notes Theor. Comput. Sci.*, vol. 272, p. 43–56. URL <https://doi.org/10.1016/j.entcs.2011.04.004>.
- Petrov, T., J. Feret et H. Koepl. 2012, «Reconstructing species-based dynamics from reduced stochastic rule-based models», dans *Winter Simulation Conference, WSC '12, Berlin, Germany, December 9-12, 2012*, édité par O. Rose et A. M. Uhrmacher, WSC, ISBN 978-1-4673-4779-2, p. 225:1–225:15. URL <https://doi.org/10.1109/WSC.2012.6465241>.
- Plateau, B. 1985, «On the stochastic structure of parallelism and synchronization models for distributed algorithms», *SIGMETRICS Perform. Eval. Rev.*, vol. 13, 2, doi:10.1145/317786.317819, p. 147–154, ISSN 0163-5999. URL <http://doi.acm.org/10.1145/317786.317819>.
- Radulescu, O., A. N. Gorban, A. Zinovyev et V. Noel. 2012, «Reduction of dynamical biochemical reactions networks in computational biology», *Frontiers in Genetics*, vol. 3, p. 131.
- Radulescu, O., S. Vakulenko et D. Grigoriev. 2015, «Model reduction of biochemical reactions networks by tropical analysis methods», *Mathematical Modelling of Natural Phenomena*, vol. 10, 3, p. 124–138.
- Regev, A., E. M. Panina, W. Silverman, L. Cardelli et E. Y. Shapiro. 2004, «Bioambients: An abstraction for biological compartments», *TCS*, vol. 325, 1, p. 141–167.
- Regev, A., W. Silverman et E. Shapiro. 2001, «Representation and simulation of biochemical processes using the pi-calculus process algebra», dans *Pacific Symposium on Biocomputing, Volume 6*, édité par R. B. Altman, A. K. Dunker, L. Hunter et T. E. Klein, Singapore, p. 459–470.
- Saint-Germain, K. C. et J. Feret. 2016, «Conservative numerical approximations of the differential semantics in biological rule-based models», Master thesis.
- Schoeberl, B., C. Eichler-Jonsson, E. D. Gilles et G. Müller. 2002, «Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized egf receptors.», *Nat Biotechnol.*, vol. 20, 4, p. 370–375.
- Stewart, D. 2010, «Spatial biomodelling», Master thesis, School of Informatics, University of Edinburgh.
- Suderman, R. et E. J. Deeds. 2013, «Machines vs. ensembles: effective mapk signaling through heterogeneous sets of protein complexes.», *PLoS Computational Biology*, vol. 9.

Tarski, A. 1955, «A lattice-theoretical fixpoint theorem and its applications.», *Pacific J. Math.*, vol. 5, 2.



# Index

état (réseau réactionnel), 25  
état d'activation, 13  
état de liaison, 13

carte de contacts, 13  
chevauchements (de motifs), 31  
complexe biochimique, 14  
concrétisation, 28  
concrétisation (fonction de), 28  
contre-partie abstraite, 30  
correspondance de Galois, 28

ensemble de motifs orthogonaux, 33

homomorphisme, 15

lemme de raffinement, 35

meilleure approximation, 28  
motif, 15

plongement, 16

règle d'interaction, 19  
règle morte, 25  
règle-réaction, 21  
rigidité, 17

site d'interaction, 13  
sorte de protéines, 13

théorème de Tarski, 26  
trace locale, 38  
transition (réseau réactionnel), 25

vue locale, 27

