HAL open science

Quelques ponts entre sémantique opérationnelle et modèles dénotationnels des langages de programmation

Tom Hirschowitz

To cite this version:

Tom Hirschowitz. Quelques ponts entre sémantique opérationnelle et modèles dénotationnels des langages de programmation. Logique en informatique [cs.LO]. Université Grenoble Alpes, 2016. tel-01400666

HAL Id: tel-01400666

https://hal.science/tel-01400666v1

Submitted on 22 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Communauté UNIVERSITÉ Grenoble Alpes

Habilitation à diriger les recherches
de l'Université de Grenoble
Spécialité : Mathématiques et Informatique

Présentée par
Tom Hirschowitz

préparée au sein du LAMA, Université Savoie Mont Blanc
et de l'École doctorale MSTII

Quelques ponts entre sémantique opérationnelle et modèles dénotationnels des langages de programmation

Habilitation soutenue publiquement le 17 novembre 2016
devant un jury composé de

Jean-Bernard Stefani (président)
Inria

Pierre-Louis Curien (rapporteur)
University Paris - Diderot

Martin Hyland (rapporteur)
University of Cambridge

Glynn Winskel (rapporteur)
University of Cambridge

Krzysztof Worytkiewicz (examinateur)
Université Savoie Mont Blanc

Paul-André Melliès (invité)
University Paris - Diderot

Résumé

Le fil directeur de ce manuscrit réside dans la volonté de mathématiser l'activité des chercheurs en programmation. Exactement comme la définition de groupe a émergé d'une quantité suffisante de cas particuliers, on y tente d'abstraire et de généraliser certains résultats et constructions en sémantique des langages de programmation, avec des motivations concrètes mais relativement diverses.

En partie 3, on esquisse une approche générale de la méthode traditionnelle de définition de langages de programmation par une syntaxe munie d'une relation de réduction modélisant l'exécution. Relativement élémentaire, cette approche présente deux avantages importants : d'une part elle rend pleinement compte du processus d'engendrement du langage à partir d'opérations de base et de règles de réduction ; d'autre part elle fournit automatiquement une notion de modèle dénotationnel du langage considéré.

En partie 4, on propose une théorie des calculs dits « graphiques », une méthode relativement récente de définition de langages de programmation — et en fait aussi de structures mathématiques — dans laquelle la syntaxe est remplacée par une structure voisine des graphes. Dans les bons cas, non seulement cette approche permet une définition rigoureuse et efficace du calcul graphique en question, mais elle fournit aussi une notion de modèle dénotationnel. Ce dernier point découle d'une théorie des présentations de foncteurs analytiques entre catégories de préfaisceaux, qui généralisent les foncteurs analytiques de Joyal [40].

En partie 5, on propose une version abstraite d'une pratique en sémantique dénotationnelle, la sémantique dite des jeux [61, 38, 1]. L'approche décrite a permis d'étendre la sémantique des jeux d'un cadre strictement séquentiel aux cadres non-déterministe et concurrent [35, 15, 16, 73]. Elle fournit aussi une explication générale du lien entre sémantique opérationnelle et modèles de jeux.

La politique touche à tout et tout touche à la politique. Dire que l'on ne fait pas de politique, c'est avouer que l'on n'a pas le désir de vivre.

Ruben Um Nyobè (1913-1958)

Remerciements

Pour commencer, merci aux membres du jury : à Pierre-Louis Curien, Martin Hyland et Glynn Winskel pour leur immense expérience, leur relecture attentive et leurs commentaires bienveillants, ainsi qu'à Paul-André Melliès, Jean-Bernard Stefani et Kris Worytkiewicz, au-delà de leur présent rôle, pour leurs encouragements depuis de longues années.

Merci à mes coauteurs et étudiants pour les travaux concernés et leurs ancêtres, à savoir Bel, Clovis Eberhart, Richard Garner, Florian Hatat, Dédou, Damien Pous, Thomas Seiller.

Merci à mes collègues proches et à ceux qui ont encouragé mes travaux : Filippo Bonchi, Simon Castellan, Pierre Clairambault, Richard Garner, Daniel Hirschko, Dédou, Paul-André Melliès, Damien Pous, Paweł Sobociński, Alexandra Silva, Alex Simpson, Nicolas Tabareau, Fabio Zanasi. Il faut leur ajouter (de manière peut-être redondante) les deux relecteurs magiques de mon long article sur CCS et les terrains de jeux [31].

Merci à mes mentors catégoriques, Dédou (encore) pour les topos, Mark Weber pour les génériques et les Carboni-Johnstoneries, Kris Worytkiewicz pour les ω-catégories et les systèmes de factorisation, ainsi (encore) que Richard Garner.

Merci à Wolfram Kahl pour m'avoir convaincu que les catégories, c'est juste un nouveau langage à apprendre, ainsi qu'à John Baez pour m'avoir fait croire un instant que ce langage était facile. Ça n'a duré que le temps de me lancer dedans, mais ça valait le coup. Merci à Ingo Waschkies pour un coup de pouce décisif et inattendu.

Merci aux participants du groupe de lecture sur les catégories organisé au Lama en 2015 et en particulier à Pierre Villemot, à qui la partie 2 devrait sembler familière.

Merci aux développeurs de Patoline, Pierre-Etienne Meunier, Christophe Raffalli, Rodolphe Lepigre, Florian Hatat, Pierre Hyvernat, Guillaume Theyssier, pour avoir fait de l'habilitation un vrai défi (à titre indicatif, il m'a fallu environ 36 minutes pour souligner le « vrai » précédent), mais aussi un vrai plaisir.

Merci à Frédéric Suffert pour la motiv.

Spéciale dédicace à Pascal Brutal, mon modèle de virilité, sans qui j'aurais pu manquer de courage dans les moments critiques.

Merci à Sissou, Karel, Lila, Lise, Mido, Dédou, Bel, Douf, Floti, El-H'Zout et Digdig, aux blairos, à mes amis, à Marc Voinchet, au Diplo, au Débat et à Sepp Holzer de me ramener à la réalité.

Table des matières

1 Introduction *9*

1.1 Résumé détaillé *9*

1.1.1 Récriture d'ordre supérieur *9*

1.1.2 Calculs graphiques *9*

1.1.3 Modèles de jeux *10*

1.2 Méthode *12*

1.2.1 Du « vrai » au « comment » *12*

1.2.1.1 Une tendance de fond *12*

1.2.1.2 Manifestation dans ce manuscrit *13*

1.2.2 Mathématiques et informatique *13*

2 Préliminaires *15*

2.1 Notations de base *15*

2.2 Catégories doubles *16*

2.3 Systèmes de transitions et bisimulations *18*

2.4 Fins, cofins et extensions de Kan *24*

2.4.1 Motivation *24*

2.4.2 Définition et premiers exemples *25*

2.4.3 Expression en termes de (co)limites ordinaires *28*

2.4.4 Lien avec les extensions de Kan *32*

2.4.5 Une application classique : réalisation géométrique et foncteur singulier *35*

2.4.6 Lien avec les (co)limites pondérées *36*

3 De la récriture d'ordre supérieur à la sémantique 2-catégorique *41*

3.1 Un lieur pour les gouverner tous *41*

3.2 Signatures *44*

3.3 Termes de réduction *47*

3.4 Equations entre réductions *50*

3.5 Une monade sur les signatures *53*

3.6 Algèbres et 2-catégories *53*

3.7 Lien avec la récriture d'ordre supérieur *57*

3.8 Une application et des perspectives *58*
4 De la syntaxe graphique à la sémantique catégorique *61*
4.1 Motivation *61*
4.2 Foncteurs familiaux sur les ensembles *65*
4.3 Foncteurs familiaux entre catégories de préfaisceaux *71*
4.4 Monades familiales libres et monades morphologiques *74*
4.5 Foncteurs analytiques *80*
4.6 Foncteurs analytiques cellulaires *94*
5 De la sémantique des jeux à la syntaxe *101*
5.1 Introduction *101*
5.1.1 Corpus couvert et historique *101*
5.1.2 Les traces multi-joueurs comme pont entre modèles de jeux et syntaxe *101*
5.2 Parties multi-joueurs *103*
5.2.1 Structure globale : premières intuitions sur les terrains de jeux *103*
5.2.2 Exemple détaillé : représentation des parties en CCS *105*
5.2.2.1 Générateurs *105*
5.2.2.2 Coups et parties *110*
5.3 Stratégies *112*
5.3.1 Stratégies naïves *112*
5.3.2 L'idée des stratégies innocentes *117*
5.3.3 Interlude : un exemple d'application des colimites pondérées *120*
5.3.4 Comportements innocents et lien avec stratégies naïves *122*
5.4 Adéquation *128*
5.4.1 Adéquation intentionnelle pleine et équivalences de test *128*
5.4.2 Processus *131*
5.4.3 Un système de transitions pour les comportements innocents *133*
5.4.4 Interprétation des processus et adéquation *136*
5.5 Perspectives *140*
6 Bibliographie *143*

1 Introduction

1.1 Résumé détaillé

Comme annoncé dans le résumé, ce texte présente trois approches pour mathématiser certaines pratiques en sémantique des langages de programmation. Il commence en partie 2 par poser les notations et rappeler quelques notions de base, ainsi que par introduire certains outils cruciaux pour les parties 4 et 5, les fins et cofins d'une part, et les extensions de Kan d'autre part.

1.1.1 Récriture d'ordre supérieur

En partie 3, on examine une pratique répétée des dizaines de fois chaque année chez les sémanticiens, qui consiste à définir un nouveau langage de programmation (ou du moins son modèle mathématique). Ceci inclut sa syntaxe, ainsi que sa sémantique *opérationnelle*, typiquement une relation binaire entre programmes, censée décrire les étapes de leur évaluation. Un point important est que la syntaxe en question comporte en général des variables *liées*, ou *muettes*, celles qu'on peut renommer sans changer le sens du programme. Un peu moins fréquemment, lorsque le langage en question atteint un statut important, se pose la question de sa sémantique *catégorique*. Ceci signifie qu'on se demande ce qu'est un *modèle* de ce langage, dans un sens voisin de celui de la théorie des modèles en logique. Cette question a en général des applications très concrètes sur le langage lui-même, d'où son importance.

La partie 3 présente une réponse au problème, qui s'applique lorsque le langage considéré entre dans le cadre limité de la récriture d'ordre supérieur. On y montre que les étapes de récriture d'ordre supérieur [62] définissent la 2-catégorie cartésienne fermée librement engendrée par le système de récriture considéré. On obtient ainsi une adjonction entre systèmes de récriture d'ordre supérieur et 2-catégories cartésiennes fermées, qu'on peut comprendre comme définissant automatiquement la catégorie des modèles dénotationnels de chaque système de récriture d'ordre supérieur.

1.1.2 Calculs graphiques

En partie 4, on considère une pratique voisine de celle analysée en partie 3, sauf qu'au lieu d'une syntaxe standard le sémanticien définit une syntaxe dite « graphique », ce qui signifie informellement qu'on dispose d'une définition très touffue, qui se comprend instantanément grâce à quelques dessins. Dans la plupart des cas, on s'empresse alors

d'oublier la définition pour raisonner sur les dessins. Comme en partie précédente, dans certains cas, la définition du langage s'accompagne d'une notion de modèle dénotationnel.

Le premier exemple de tel langage graphique est fourni par les réseaux de preuve en logique linéaire [27], dont les modèles dénotationnels ont donné lieu à d'intenses recherches [5, 12]. D'autres exemples, au-delà des réseaux de preuves et de leurs variantes, sont les *réseaux d'interaction* [47] et les *bigraphes* [39]. Dans la même veine, certaines notions mathématiques bien établies et graphiques dans le sens ci-dessus ont été mises en correspondance avec des concepts informatiques. C'est le cas par exemple des *polycatégories* [71], qui correspondent à un fragment des réseaux de preuves, ainsi que des PROPs, récemment mis à l'honneur en informatique théorique [77].

La partie 4 élabore une approche générale des calculs graphiques. Le point de départ en est qu'on peut donner un sens formel aux dessins utilisés pour expliquer ces calculs graphiques. On peut inférer à partir de ce sens formel une définition alternative du calcul graphique considéré, grâce à une théorie des présentations de foncteurs analytiques entre catégories de préfaisceaux, qui généralisent les foncteurs analytiques de Joyal [40]. Le calcul graphique engendre un foncteur analytique, qu'on peut munir, sous certaines hypothèses, d'une structure de monade : les modèles dénotationnels du calcul graphique peuvent alors être définis comme les algèbres pour cette monade. L'aspect le plus prometteur de cette théorie est sans doute qu'elle permet d'engendrer de nouvelles structures algébriques simplement à partir de leur calcul graphique.

1.1.3 Modèles de jeux

En partie 5, on prend le problème à l'envers : on part d'une pratique en sémantique dénotationnelle, la sémantique dite des jeux [61, 38, 1], dont on tente de proposer une version abstraite. Bien que j'en sois convaincu, il n'est pas encore établi à ce jour que cette version abstraite rende exactement compte des modèles de jeux visés. Cependant, elle a permis d'étendre la sémantique des jeux d'un cadre strictement séquentiel aux cadres non-déterministe et concurrent [35, 15, 16, 73].

L'observation cruciale à la base de cette approche est que l'*innocence* des stratégies dans les modèles de jeux [38] peut s'exprimer comme une condition de faisceaux [56]. En effet, il est connu que les stratégies peuvent être vues comme des préfaisceaux booléens sur les parties du jeu munies de l'ordre préfixe, i.e. des foncteurs $\mathscr{P}^{\mathrm{op}} \longrightarrow \mathbf{2}$, où le caractère gras $\mathbf{2}$ désigne l'ordinal fini à 2 éléments vu comme une catégorie et $\mathscr{P}$ désigne l'ordre partiel des parties. En enrichissant l'ordre préfixe avec de nouveaux morphismes entre parties prenant mieux en compte leur structure intrinsèquement multi-joueurs, on peut définir une topologie de Grothendieck selon laquelle toute partie est couverte par ses *vues* (c'est-à-dire ses parties mono-joueur). Les faisceaux booléens pour cette topologie sont des stratégies multi-joueurs innocentes, c'est-à-dire dont le comportement de chaque joueur ne dépend que de sa vue. En passant des faisceaux booléens aux faisceaux

arbitraires, on obtient une notion de stratégie innocente et concurrente qui s'avère stable par composition.

Cette observation nous a permis de définir une notion générale de présentation, les *terrains de jeux*, dont on a montré qu'elle permet de mettre en correspondance la syntaxe et le modèle de jeux associés. Intuitivement, un terrain de jeux représente une notion de *partie* (au sens des jeux), ou de *trace* (au sens des processus). A partir d'un terrain de jeux arbitraire $\mathscr{T}$, on construit deux systèmes de transitions étiquetées dans le même graphe $\mathscr{M}_{\mathscr{T}}$, ayant pour sommets les positions du jeu et pour arêtes les coups entre eux :

- un système syntaxique $\mathscr{O}_{\mathscr{T}}$ (pour « opérationnel ») dont les états sont les termes de la syntaxe associée à $\mathscr{T}$
- et un système sémantique $\mathscr{S}_{\mathscr{T}}$, dont les états sont les stratégies du modèle de jeux associé à $\mathscr{T}$.

On construit ensuite une fonction de traduction $[\![-]\!] : \mathscr{O}_{\mathscr{T}} \longrightarrow \mathscr{S}_{\mathscr{T}}$ dont on montre qu'elle relie des états fortement bisimilaires.

Bien que la théorie des terrains de jeux n'englobe encore pas cet aspect, on a montré dans deux cas significatifs, CCS [35] et le π-calcul [15, 16], que ce résultat entraîne un résultat peut-être plus pertinent de pleine abstraction intentionnelle vis-à-vis d'équivalences de test [13]. Nous soupçonnons que le travail d'Ong et Tsukada [73] peut aussi être présenté de cette manière.

L'idée fondamentale sous-tendant ce développement est que le pont entre la syntaxe et les jeux repose sur une notion de *trace d'exécution*, ou *partie*, multi-joueurs :

- les branches des arbres de syntaxe sont simplement des parties mono-joueur, qu'on appelle des *vues* en référence à Hyland et Ong; cela revient à dire que les arbres de syntaxe sont des préfaisceaux sur les parties mono-joueurs;
- les stratégies sont des préfaisceaux sur les parties multi-joueurs, satisfaisant une condition de localité qu'on appelle *innocence* toujours en référence à Hyland et Ong et qui peut s'exprimer comme une condition de faisceau;
- dans ce cas particulier, la faisceautisation, c'est-à-dire la construction du faisceau canoniquement associé à un préfaisceau donné, se fait par une simple extension de Kan à droite.

Un point crucial est que les parties doivent être munies d'une notion de morphisme plus riche que le simple ordre préfixe. Cette notion de morphisme met en avant le caractère multi-joueurs des parties puisqu'elle fournit en particulier pour chaque partie P un morphisme $V \longrightarrow P$ pour chaque vue V de P. Elle permet de plus d'incorporer une notion de symétrie sur les parties qui rend compte sans effort des questions de renommage de variables liées, de manière relativement proche des ensembles nominaux [23]. Les premières prémisses de cette idée sont apparues vers 2008 dans mes travaux en famille sur un modèle de jeux multi-joueurs pour la logique linéaire [33, 34]. Les outils pour la traiter efficacement datent du premier article sur CCS [35].

La situation catégorique est aussi moins habituelle que dans les parties précédentes puisqu'on a un diagramme

$$\begin{tikzcd}
 & \text{Termes} \arrow[dd, Rightarrow, "\text{Traduction}"] \arrow[dr, bend left] & \\
\text{Terrains de jeux} \arrow[ur, bend left] \arrow[dr, bend right] & & \text{Systèmes de transitions étiquetés} \\
 & \text{Stratégies} \arrow[ur, bend right] &
\end{tikzcd}$$

Comme aucun des deux sommets en préserce n'est canoniquement muni d'une structure de catégorie, il faut préciser. On a des fonctions entre les sommets comme indiqué et la « 2-cellule » représente, pour chaque terrain de jeux $\mathscr{T}$, la bisimulation fonctionnelle $[\![-]\!] : \mathscr{O}_{\mathscr{T}} \longrightarrow \mathscr{S}_{\mathscr{T}}$.

Ainsi, l'approche répond seulement partiellement au cahier des charges énoncé en début de partie : elle fournit une notion de présentation pour les langages, mais n'associe pas à chaque langage une catégorie de modèles. Pour améliorer la situation, il faudrait sans doute répondre formellement à la question : qu'est-ce qu'un modèle de jeux ?

1.2 Méthode

Ce texte, comme la plupart de mes travaux, repose sur un petit nombre de credos méthodologiques.

1.2.1 Du « vrai » au « comment »

Le premier point important réside dans la place accordée au « comment », par opposition aux réponses par oui ou non.

1.2.1.1 Une tendance de fond

Cette tendance est significative dans plusieurs branches des mathématiques.

En logique d'abord, avec l'importance prise par la théorie de la démonstration et le rapport avec la programmation, à travers la correspondance de Curry-Howard. Ce point est particulièrement saillant dans les travaux de Girard, qui raffine la métaphore de la logique comme fondations de l'édifice des mathématiques en distinguant trois sous-sols : la validité booléenne, la notion de démonstration et la dynamique de l'élimination des coupures. Dans cette liste, chaque élément est vu comme plus profond que le précédent. Techniquement, il le raffine en ajoutant un niveau de « comment » : une démonstration décrit comment une formule est valide ; un processus d'élimination de coupures relie deux démonstrations.

La théorie des *catégories*, inventée dans les années 40 par Eilenberg et MacLane, a grandement favorisé cette tendance. Elle insiste en effet sur la notion de morphisme, et en particulier d'isomorphisme, entre objets d'une même catégorie. Elle met aussi en avant la notion d'*équivalence* entre catégories, plus générale et plus flexible que la notion d'isomorphisme, qui revient en gros à un isomorphisme à isomorphisme près. Elle encourage enfin la formulation d'énoncés de la forme « tel foncteur $\mathscr{C} \to \mathscr{D}$ est une équivalence », par opposition à « $\mathscr{C}$ et $\mathscr{D}$ sont équivalentes ». En fait, l'utilisation des catégories mène naturellement à l'itération de ce passage au « comment », poussant ainsi à l'extrême l'idée des sous-sols de Girard. Les catégoriciens appellent ce processus la *categorification* [14]. A la limite, on obtient les catégories dites (de dimension) supérieure [51].

1.2.1.2 Manifestation dans ce manuscrit

Ce parti pris se manifeste tout d'abord dans le choix des questions abordées. En partie 3, on raffine la sémantique catégorique des systèmes de récriture d'ordre supérieur, ce qui revient à distinguer différentes preuves d'égalité entre morphismes — autrement dit on se demande comment démontrer que deux morphismes sont égaux. Mais surtout, le but principal de cette partie réside dans la mathématisation du processus d'engendrement d'une relation binaire de récriture à partir de données basiques. En d'autres termes, on se demande comment procèdent les chercheurs en conception et sémantique des langages de programmation. En partie 4, la question centrale est là encore la construction de monades : plutôt que de simplement se demander si tel ou tel foncteur est une monade, on propose des procédés plus ou moins automatiques de construction de monades. Comme en partie 3, ceci se traduit par la mise en place d'une adjonction entre nos monades dites morphologiques et des données plus basiques. Enfin, sans fournir de réponse définitive, la partie 5 aborde la question de comment construire une sémantique de jeu pour un langage arbitraire éventuellement concurrent.

Au-delà des questions abordées, la place accordée au « comment » se traduit de manière peut-être plus technique par certains choix de conception ou d'énoncé. Typiquement, le passage des stratégies comme ensembles de parties « acceptées » aux stratégies comme préfaisceaux revient à remplacer un critère booléen (« telle partie est acceptée ou non ») par un ensemble. On peut comprendre ce dernier comme décrivant l'ensemble des manières d'accepter la partie en question, ou bien comme un ensemble d'états possibles après l'avoir acceptée.

1.2.2 Mathématiques et informatique

Le second point, peut-être moins prégnant, est la thèse que l'utilisation des mathématiques en informatique, et en particulier en sémantique des langages de programmation, en est à ses balbutiements. Outre le contenu de ce manuscrit, sur lequel je suis mal placé pour me

prononcer, ces dernières années ont été le théâtre de quelques connexions spectaculaires, qui laissent présager une activité extrêmement riche. On pense bien sûr au lien entre topologie algébrique, catégories supérieures et théorie des types découvert par Hofmann et Streicher [36] et récemment mis en avant par Voevodsky [42]. Mais on peut penser aussi aux applications de la topologie algébrique en imagerie ou plus près de nous en théorie de la concurrence [18], à l'approche catégorique de la théorie des réseaux [22, 77], à l'approche de la récriture à l'aide de catégories supérieures [30], ou à l'utilisation d'algèbres d'opérateurs en complexité, permise par la géométrie de l'interaction.

Il me parait important d'interagir avec les mathématiciens sur des questions aussi intimidantes que la certification de programmes (en particulier de compilateurs), la certification de démonstrations mathématiques, ou la séparation de classes de complexité algorithmique. Or, un des buts de ce manuscrit est de replacer certaines notions standard en informatique, telles que la récriture d'ordre supérieur, les réseaux d'interaction, les stratégies innocentes, dans un contexte plus significatif pour les mathématiciens. La construction d'un système de récriture d'ordre supérieur à partir de règles est vue comme un foncteur adjoint ; les réseaux d'interaction donnent lieu à un cas particulier de monade morphologique ; les stratégies innocentes sont vues comme des faisceaux sur un site combinatoire.

2 Préliminaires

Le présent manuscrit fait un usage constant de la théorie des catégories et suppose connues les notions de catégories, foncteurs, transformations naturelles, limites et colimites, foncteurs adjoints, faisceaux et préfaisceaux, bicatégories et pseudo catégories doubles. Toutes ces notions sauf la dernière sont traitées par les livres classiques de Mac Lane [55] et Mac Lane et Moerdijk [56]. Le lecteur débutant trouvera dans le récent livre de Leinster [52] une introduction moins aride d'une bonne partie de ces outils. On utilise aussi en de rares occasions des notions de catégories localement présentables [2].

Dans cette partie, après avoir fixé certaines notations de base (partie 2.1), on rappelle brièvement les notions de pseudo catégories doubles (partie 2.2) et de systèmes de transitions étiquetées (partie 2.3). Enfin, on donne en partie 2.4 une présentation plus détaillée des (co)fins et des extensions de Kan. Ces notions (et les liens entre elles) offrent en effet un outil de raisonnement combinatoire bien utile et, au fond, très intuitif. Cependant, bien que présentées en détail dans MacLane [55] et au début de MacLane et Moerdijk [56], mon expérience montre qu'elles sont généralement perçues comme difficiles par les informaticiens. Ceci peut raisonnablement s'expliquer par le fait que les démonstrations exposées dans ces deux livres prennent le parti d'une présentation élémentaire : elles introduisent seulement le minimum d'outils nécessaire. On suit ici la présentation de Riehl [66], qui, bien que reposant sur des concepts plus avancés tels que les limites et colimites pondérées (avec en sous-main la théorie des catégories enrichies [43]), me semble significativement plus claire. En particulier, on donne une démonstration efficace (bien connue) du lemme dit de « co-Yoneda », selon lequel tout préfaisceau est canoniquement colimite de représentables—présenté ici sous deux formes différentes, le lemme 2.4.21 et le corollaire 2.4.31.

2.1 Notations de base

On fixe ici certaines notations de base. Un entier naturel n est souvent vu comme l'ensemble fini $\{1, \ldots, n\}$. En de rares occasions, on préfère le voir comme l'ordinal correspondant, $\{0, \ldots, n-1\}$, auquel cas on le note en gras : $\mathbf{n}$.

Dans toute catégorie $\mathscr{C}$, on note $\mathscr{C}(A, B)$ l'ensemble des morphismes entre deux objets A et B, et id_A l'identité sur A. On note la composition traditionnellement comme la composition de fonctions : $g \circ f$. On note parfois $C \in \mathscr{C}$ pour $C \in \mathrm{ob}(\mathscr{C})$. Pour tout objet C et tout ensemble X, $X \cdot C$ désigne le coproduit itéré $|X|$ fois, c'est-à-dire $C + \cdots + C$.

Set désigne la catégorie habituelle des ensembles ; set désigne un squelette de la catégorie des ensembles finis, par exemple la catégorie des ordinaux finis et des fonctions arbitraires ; ford désigne la catégorie des ordinaux finis et des fonctions croissantes.

Pour toute catégorie $\mathscr{C}$, $\widehat{\mathscr{C}} = [\mathscr{C}^{\mathrm{op}}, \mathrm{Set}]$ désigne la catégorie des préfaisceaux sur $\mathscr{C}$ et des transformations naturelles. On note $\mathbf{y} : \mathscr{C} \to \widehat{\mathscr{C}}$ le plongement de Yoneda et on abrège souvent $\mathbf{y}_C$ en simplement C. Enfin, pour tout préfaisceau F sur $\mathscr{C}$, élément $x \in F(D)$ et morphisme $f : C \to D$, on note $x \cdot f$ pour $F(f)(x)$.

Remarque 2.1.1. Cette notation entre en conflit avec la notation $X \cdot C$ ci-dessus pour le coproduit itéré. En pratique, le contexte devrait suffire à lever l'ambiguïté, puisque dans $x \cdot f$ le facteur de droit f, un morphisme, agit sur le facteur de gauche x, un élément d'un préfaisceau, alors que dans $X \cdot C$, le membre de gauche, un ensemble, agit sur celui de droite, un objet.

On a plusieurs notions de finitude pour les préfaisceaux sur une catégorie $\mathscr{C}$. Une première est donnée par les préfaisceaux d'ensembles finis, c'est-à-dire les foncteurs $\mathscr{C}^{\mathrm{op}} \to \mathrm{set}$. Une deuxième est donnée par les préfaisceaux F dont l'ensemble des éléments $\sum_{C \in \mathrm{ob}(\mathscr{C})} F(C)$ est fini—attention, pour certains choix de $\mathscr{C}$, même les représentables peuvent ne pas être finis en ce sens. Une troisième consiste à considérer les préfaisceaux *finiment présentables* [2]. La définition standard de ceux-ci est qu'ils préservent les colimites filtrées, mais une caractérisation peut-être plus éclairante est qu'ils sont colimites finies de représentables. Dans tous les cas concrets que nous rencontrerons ici, les deux dernières notions coïncident.

Enfin, étant donné un cospan de foncteurs $\mathscr{A} \xrightarrow{F} \mathscr{C} \xleftarrow{G} \mathscr{B}$, on désigne par F/G la catégorie *comma*, dont les objets sont les triplets (A, B, φ) avec $A \in \mathscr{A}$, $B \in \mathscr{B}$ et $\varphi : F(A) \to G(B)$ et dont les morphismes $(A, B, \varphi) \to (A', B', \varphi')$ sont les paires d'un $u : A \to A'$ et d'un $v : B \to B'$ tels que

$$\begin{tikzcd} F(A) \arrow[r, "F(u)"] \arrow[d, "\varphi"'] & F(A') \arrow[d, "\varphi'"] \\ G(B) \arrow[r, "G(v)"'] & G(B') \end{tikzcd}$$

commute. Il est bien connu que cette catégorie admet une propriété universelle de dimension 2 analogue à celle d'un pullback [44].

2.2 Catégories doubles

Définition 2.2.1. Une *catégorie double* $\mathbb{D}$ est constituée de deux catégories $\mathbb{D}_h$ et $\mathbb{D}_v$ partageant le même ensemble d'objets, ainsi que, pour chaque carré

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "u"'] & B \arrow[d, "v"] \\ C \arrow[r, "g"'] & D, \end{tikzcd}$$

où $f \in \mathbb{D}_h(A, B)$, $g \in \mathbb{D}_h(C, D)$, $u \in \mathbb{D}_v(A, C)$, $v \in \mathbb{D}_v(B, D)$, d'un ensemble $\mathbb{D}(A, B, C, D, f, u, v, g)$ de *cellules*. On représente ces cellules en « remplissant » le carré en question, éventuellement avec une flèche double « $\Longrightarrow$ » comme pour les 2-cellules d'une 2-catégories (sauf que l'orientation est ici donnée par le périmètre).

Comme Paré [63], on note les identités et la composition de $\mathbb{D}_h$ de manière standard (id et $\circ$), alors que pour $\mathbb{D}_v$ on note $\mathrm{id}^\bullet$ et $\bullet$.

Ces données doivent de plus être munies

- d'une loi $\circ$ de composition *horizontale* associative de neutre $\mathrm{Id}_u \in \mathbb{D}(A, A, C, C, \mathrm{id}_A, u, u, \mathrm{id}_A)$, associant à chaque diagramme comme ci-dessous à gauche une cellule comme ci-dessous à droite :

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "u"'] \arrow[dr, phantom, "\overset{\alpha}{\Longrightarrow}"] & B \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "\overset{\beta}{\Longrightarrow}"] & E \arrow[d, "w"] \\ C \arrow[r, "g"'] & D \arrow[r, "k"'] & F \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} A \arrow[r, "h\circ f"] \arrow[d, "u"'] \arrow[dr, phantom, "\overset{\beta\circ\alpha}{\Longrightarrow}"] & B \arrow[d, "w"] \\ E \arrow[r, "k\circ g"'] & F, \end{tikzcd}$$

- d'une loi $\bullet$ de composition *verticale* associative et de neutre $\mathrm{Id}^\bullet_f \in \mathbb{D}(A, B, A, B, f, \mathrm{id}^\bullet_A, \mathrm{id}^\bullet_B, f)$, associant à chaque diagramme comme ci-dessous à gauche une cellule comme ci-dessous à droite :

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "u"'] \arrow[dr, phantom, "\overset{\alpha}{\Longrightarrow}"] & B \arrow[d, "v"] \\ C \arrow[r, "g"] \arrow[d, "u'"'] \arrow[dr, phantom, "\overset{\beta}{\Longrightarrow}"] & D \arrow[d, "v'"] \\ E \arrow[r, "h"'] & F \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} A \arrow[r, "f"] \arrow[d, "u'\bullet u"'] \arrow[dr, phantom, "\overset{\beta\bullet\alpha}{\Longrightarrow}"] & B \arrow[d, "v'\bullet v"] \\ E \arrow[r, "g"'] & F, \end{tikzcd}$$

vérifiant les axiomes suivants :

- $\mathrm{Id}_{v\bullet u} = \mathrm{Id}_v \bullet \mathrm{Id}_u$ pour tous $A \xrightarrow{u} B \xrightarrow{v} C$;
- $\mathrm{Id}^\bullet_{g\circ f} = \mathrm{Id}^\bullet_g \circ \mathrm{Id}^\bullet_f$ pour tous $A \xrightarrow{f} B \xrightarrow{g} C$;
- $\mathrm{Id}_{\mathrm{id}^\bullet_A} = \mathrm{Id}^\bullet_{\mathrm{id}_A}$ pour tout objet A ;
- $(\beta' \bullet \beta) \circ (\alpha' \bullet \alpha) = (\beta' \circ \alpha') \bullet (\beta \circ \alpha)$ pour tous

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "u"'] \arrow[dr, phantom, "\overset{\alpha}{\Longrightarrow}"] & B \arrow[r, "g"] \arrow[d, "v"'] \arrow[dr, phantom, "\overset{\beta}{\Longrightarrow}"] & C \arrow[d, "w"] \\ A' \arrow[r, "f'"] \arrow[d, "u'"'] \arrow[dr, phantom, "\overset{\alpha'}{\Longrightarrow}"] & B' \arrow[r, "g'"] \arrow[d, "v'"'] \arrow[dr, phantom, "\overset{\beta'}{\Longrightarrow}"] & C' \arrow[d, "w'"] \\ A'' \arrow[r, "f''"'] & B'' \arrow[r, "g''"'] & C''. \end{tikzcd}$$

Définition 2.2.2. Une *pseudo catégorie double*, en bref, est une catégorie double dans laquelle les $\mathrm{id}^{\bullet}_{A}$ et $\mathrm{Id}^{\bullet}_{f}$ ne sont neutres qu'à isomorphisme naturel cohérent près et les compositions $\bullet$ (sur les morphismes verticaux et les cellules) ne sont associatives qu'à isomorphismes naturels cohérents près. L'axiomatisation complète est donnée dans la thèse de Richard Garner [25].

Notation 2.2.1. Dans une (pseudo) categorie double $\mathbb{D}$, on notera
- $\mathbb{D}_h$ pour la catégorie horizontale,
- $\mathbb{D}_v$ pour la (bi)catégorie verticale
- et $\mathbb{D}_H$ pour la catégorie dont
  - les objets sont les morphismes verticaux,
  - les morphismes $u \to u'$ sont toutes les cellules de bords gauche et droit u et u', respectivement,
  - la composition est donnée par la composition horizontale de cellules.

Un morphisme dans $\mathbb{D}_H$ est dit *spécial* si ses bords haut et bas sont des identités. De la même manière, $\mathbb{D}_V$ désigne la bicatégorie dont les objets sont les morphismes horizontaux, les morphismes $h \to h'$ sont toutes les cellules de bord haut et bas h et h', respectivement, et la composition est donnée par la composition verticale de cellules.

Exemple 2.2.1. A partir de toute catégorie $\mathbb{C}$ munie de pushouts on peut construire la pseudo catégorie double $\mathbb{D} = \mathrm{Cospan}(\mathbb{C})$
- dont la catégorie horizontale $\mathbb{D}_h$ est $\mathbb{C}$,
- dont les morphismes verticaux $A \nrightarrow B$ sont les cospans $A \to C \leftarrow B$ et
- dont les cellules sont les diagrammes ci-dessous, le périmètre devant être clair :

$$\begin{tikzcd} A \arrow[r] \arrow[d, "s"'] & A' \arrow[d, "s'"] \\ B \arrow[r] & B' \\ C \arrow[r] \arrow[u, "t"] & C'. \arrow[u, "t'"'] \end{tikzcd}$$

La composition horizontale se fait comme dans $\mathbb{C}$ composante par composante. La composition verticale se fait par pushout dans $\mathbb{D}_v$ et par propriété universelle du pushout dans $\mathbb{D}_V$.

2.3 Systèmes de transitions et bisimulations

On utilise en partie 5 des notions standard en théorie de la concurrence [57]. On en adopte cependant une version légèrement plus générale que la version standard, qui est une sorte de version jouet de celle de Fiore [19] (voir *op cit.* pour d'autres travaux connexes). La différence principale entre notre version et celle de Fiore est que là où nous travaillons avec des graphes réflexifs, Fiore travaille avec des catégories. Par ailleurs, la seule différence

significative avec le cadre standard réside dans le fait qu'on accepte comme alphabets des graphes arbitraires—vus comme des alphabets « typés ». Une différence plus technique est qu'on considère des systèmes de transitions pouvant comporter plusieurs manières d'effectuer une transition entre deux états. Ce choix n'est dû qu'au fait que le développement s'en trouve facilité.

On travaillera—pour l'exemple—avec des graphes non-réflexifs en partie suivante, mais pour l'instant on se concentre sur les graphes réflexifs. On peut définir ces derniers de manière élémentaire, comme étant constitués d'un ensemble V de *sommets* et d'un ensemble E d'arêtes, chaque arête e possédant une *source* $s(e)$ et un *but* $t(e)$ dans V, ce qu'on note $e : s(e) \to t(e)$, munis pour tout $v \in V$ d'une arête spéciale $\mathrm{id}_v : v \to v$. Une définition équivalente, qui a le mérite de fournir directement une notion de morphisme, est la suivante :

Définition 2.3.3. Soit Γ_r la catégorie librement engendrée par le graphe a deux sommets $\star$ et $[1]$, deux arêtes $s, t : \star \to [1]$ et une arête $e : [1] \to \star$, quotientée par les équations

$$e \circ s = \mathrm{id}_\star \qquad \text{et} \qquad e \circ t = \mathrm{id}_\star.$$

Soit $\mathrm{RGph} = \widehat{\Gamma_r}$ la catégorie des préfaisceaux sur Γ_r. On appelle ses objets des *graphes réflexifs*.

On omet parfois l'adjectif « réflexif » lorsque le contexte est suffisamment clair. C'est tout ! On définit ainsi la catégorie des *systèmes de transition* sur un graphe A simplement comme la catégorie RGph/A des graphes réflexifs au-dessus de A. A est alors appelé l'*alphabet* du système de transition.

Notation 2.3.2. Plus une convention qu'une notation, en fait : on considère une arête $e : v \to v'$ dans G comme une transition de v' vers v. Ceci est motivé par le sens analogue choisi pour les morphismes verticaux dans les terrains de jeux, qui à son tour présente l'avantage de mettre en avant l'analogie entre l'axiome de fibration et les pullbacks.

Pour tout graphe réflexif G, soit $G^\star$ la catégorie engendré par le graphe sous-jacent à G, c'est-à-dire en oubliant sa structure réflexive. En d'autre termes, $G^\star$ a les mêmes sommets que G et ses morphismes sont les chemins arbitraires de G. Les identités et la composition sont données par les chemins vides et la concaténation. On a donc :

Proposition 2.3.1. L'assignation $G \mapsto G^\star$ s'étend en un foncteur $\mathrm{RGph} \to \mathrm{Cat}$.

Remarque 2.3.2. Attention, ce foncteur n'est pas l'adjoint à gauche du foncteur d'oubli $\mathscr{U} : \mathrm{Cat} \to \mathrm{RGph}$. Rapidement, considérons le graphe réflexif terminal 1, qui possède exactement une flèche identité sur un unique sommet, disons x et soit $\Sigma\,\mathbb{N}$ le monoïde des entiers naturels vu comme une catégorie à un objet. Tout morphisme $1 \to \mathscr{U}(\Sigma\,\mathbb{N})$ envoie forcément id_x sur 0, l'identité de l'unique objet de $\Sigma\,\mathbb{N}$. On a donc $\mathrm{RGph}(1, \mathscr{U}(\Sigma\,\mathbb{N})) \cong 1$.

Or un foncteur $1^\star \to \Sigma\,\mathbb{N}$ doit certes envoyer le chemin vide—dans ce cadre l'identité de x—sur 0. Cependant, le chemin singleton (id_x) n'a aucun statut particulier dans $1^\star$, donc peut être envoyé sur n'importe quel entier. Ceci donne $\mathrm{Cat}\,(1^\star, \Sigma\,\mathbb{N}) \cong \mathbb{N}$, ce qui empêche $(-)^\star$ d'être adjoint à gauche de $\mathscr{U}$.

Un adjoint à gauche existe cependant :

Définition 2.3.4. Soit, pour tout $G \in \mathrm{RGph}$, $\mathrm{fc}\,(G)$ la catégorie obtenue en quotientant les morphismes de $G^\star$ par la plus petite congruence identifiant chaque chemin singleton (id_x) avec le chemin vide sur x, pour tout $x \in G$.

Autrement dit, les chemins de $G^\star$ sont considérés comme équivalents modulo effacement des arêtes identités. Tout chemin ρ de $A^\star$ a donc une forme normale $\lfloor \rho \rfloor$, obtenue en effaçant toutes ses arêtes identités. On appelle ces chemins *sans identités*.

Proposition 2.3.2. L'assignation $G \mapsto \mathrm{fc}\,(G)$ s'étend en un foncteur $\mathrm{RGph} \to \mathrm{Cat}$, qui est adjoint à gauche du foncteur d'oubli.

Notation 2.3.3. Etant donné un système de transitions $p : G \to A$ et une arête a dans A, on note $x \xleftarrow[A]{a} x'$ pour indiquer l'existence d'une arête $e \in G\,(x', x)$ telle que $p\,(e) = a$. On étend cette notation aux chemins, ce qui donne $x \xleftarrow[A]{r} x'$ pour tout $r \in A^\star(p\,(x')\,,\, p\,(x))$. On note de même $x \xleftarrow[A]{r} x'$ pour indiquer l'existence d'un chemin $\rho \in G^\star(x', x)$ dont l'image par $p^\star : G^\star \to A^\star$ est équivalente à r, c'est-à-dire $\lfloor p^\star(\rho) \rfloor = \lfloor r \rfloor$.

On conclut cette partie en revoyant les notions de bisimulation et de bisimilarité, fortes et faibles, qu'on reformule en se basant sur la notion suivante :

Définition 2.3.5. Une *fibration de graphes* est un morphisme $f : G \to G'$ dans RGph tel que pour tous $x \in G$, $y \in G'$ et $e' \in G'(y\,,\, f\,(x))$, il existe $x' \in G$ et $e \in G\,(x', x)$ tel que $f\,(e) = e'$.

Pour tous morphismes $p : G \to A$ et $p' : G' \to A$. Une *relation au-dessus de* A est un sous-graphe du pullback

$$\begin{tikzcd} G \times_A G' \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & G' \arrow[d, "p"] \\ G \arrow[r, "p'"'] & A\,. \end{tikzcd}$$

En particulier, les sources (resp. les buts) de deux arêtes $(e\,,\, e') \in R \subseteq G \times_A G'$ sont aussi reliées. On note de telles relations par une flèche barrée : $R : G \nrightarrow G'$.

On rencontrera ici principalement des relations *pleines*, c'est-à-dire telles que pour toutes arêtes $e : x \to y$ et $e' : x' \to y'$, $R(e, e') \iff R(x, x') \wedge R(y, y')$. Bien sûr, de telles relations sont entièrement déterminées par leur valeur sur les sommets.

Définition 2.3.6. Une *simulation* $G \nrightarrow G'$ est une relation R au-dessus de A telle que pour tout $e \in G(x', x)$, si $R(x, y)$, alors il existe y' et $e' \in G'(y', y)$ tels que $R(e, e')$. Une *bisimulation* est une simulation dont la réciproque (la relation $R' : G' \nrightarrow G$ telle que $R'(b, a) \iff R(a, b)$) est aussi une simulation.

Lorsque R est pleine, R est une simulation ssi pour tout $e \in G(x', x)$, si $R(x, y)$, alors il existe y' et $e' \in G'(y', y)$ tels que $R(x', y')$ et de plus e et e' ont la même image dans A.

Proposition 2.3.3. R est une simulation ssi sa première projection $R \hookrightarrow G \times_A G' \to G$ est une fibration de graphes. De même, R est une bisimulation ssi ses deux projections sont des fibrations de graphes.

Démonstration. Facile. □

Remarque 2.3.3. La caractérisation des simulations en termes de fibrations de graphes est due à Joyal et al. [41], qui ont observé qu'un morphisme $f : G \to G'$ dans RGph/A est une bisimulation fonctionnelle ssi pour tout carré commutatif de la forme

$$\begin{tikzcd} \mathbf{y}_\star \arrow[r] \arrow[d, "\mathbf{y}_t"'] & G \arrow[d, "f"] \\ \mathbf{y}_{[1]} \arrow[r] \arrow[ur, dashed] & G', \end{tikzcd}$$

il existe un morphisme en pointillés faisant commuter les deux triangles. Ici, $\mathbf{y}_t : \mathbf{y}_\star \to \mathbf{y}_{[1]}$ envoie le graphe réflexif à un sommet (et son arête identité) sur celui avec deux sommets et une arête non-identité e entre eux, en choisissant le but de e. Ceci revient précisément à dire que f est une fibration de graphes. Cette caractérisation peut sembler mystérieuse en ce qu'elle semble indépendante de A. En fait, R est une relation au-dessus de $G \times_A G'$ et f est un morphisme au-dessus de A.

Si on fixe G et G' au-dessus de A, on retrouve des résultats standard :

Proposition 2.3.4. Les bisimulations sont stables par union. L'union de toutes les bisimulations, la *bisimilarité*, est encore une bisimulation.

Si on se restreint aux *endo*relations $G \nrightarrow G$, on parle de bisimulations et de bisimilarité *dans* G.

Notation 2.3.4. La bisimilarité dans G au-dessus de A est notée $\sim_A$. Elle peut, par un léger abus de notation, se comprendre comme une relation d'équivalence sur tous les sommets de deux graphes arbitraires au-dessus de A. En effet, si G et G' sont des graphes au-dessus de A, on écrit $x \sim_A y$, si $x \in G$ et $y \in G'$, pour désigner la bisimilarité dans $G + G'$.

Avant d'aborder les bisimulations faibles, considérons un premier résultat de stabilité. Ce résultat est utile dans la démonstration du théorème 5.4.16, qui est omise ici : on le donne afin d'illustrer notre présentation des bisimulations.

Tout morphisme $f : A \to B$ induit par pullback un foncteur dit de *changement de base* $\Delta_f : \mathrm{RGph}/B \to \mathrm{RGph}/A$, qui admet un adjoint à gauche Σ_f donné par post-composition avec f.

Proposition 2.3.5. Pour tout morphisme de graphes réflexifs $f : A \to B$, les deux foncteurs $\Delta_f : \mathrm{RGph}/B \to \mathrm{RGph}/A$ et $\Sigma_f : \mathrm{RGph}/A \to \mathrm{RGph}/B$ préservent les bisimulations fonctionnelles.

Démonstration. Le cas de Σ_f est trivial. Pour Δ_f, on utilise la remarque *2.3.3*. Par le lemme des pullbacks, le carré ci-dessous à droite est un pullback. On vérifie que $\Delta_f(G) \to \Delta_f(G')$ est une bisimulation. En effet, considérons un carré comme ci-dessous à gauche :

$$\begin{tikzcd}
\mathbf{y}_\star \arrow[r] \arrow[d, "\mathbf{y}_t"'] & \Delta_f(G) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & G \arrow[d] \\
\mathbf{y}_{[1]} \arrow[r] \arrow[ur, dashed, "k"] \arrow[urr, dashed, "h"] & \Delta_f(G') \arrow[r] & G'.
\end{tikzcd}$$

Comme $G \to G'$ est une bisimulation, on obtient un unique morphisme h, dessiné en pointillés, faisant commuter les deux triangles correspondants. De plus, par propriété universelle du pullback, on obtient un unique morphisme k faisant commuter les deux triangles adéquats. Enfin, le triangle du haut commute après post-composition avec $\Delta_f(G) \to G$, ainsi qu'après post-composition avec $\Delta_f(G) \to \Delta_f(G')$. Il commute donc par unicité dans la propriété universelle du pullback. □

Remarque 2.3.4. Ce résultat est une instance du fait que la classe de droite de tout système de factorisation faible est stable par changement de base [37]. Le système de factorisation faible pertinent ici est à génération cofibrante, avec comme seule cofibration de base le morphisme $\mathbf{y}_t$.

Traitons à présent les bisimulations faibles. Pour cela, on transpose dans notre cadre la présentation des bisimulations faibles comme des bisimulations fortes sur un système de transitions « saturé » par les actions silencieuses. Techniquement, on utilise pour cela le foncteur fc vu ci-dessus.

Définition 2.3.7. Un morphisme $f : G \to G'$ in RGph/A est une *bisimulation fonctionnelle faible* ssi $\mathrm{fc}(f) : \mathrm{fc}(G) \to \mathrm{fc}(G')$ est une fibration de graphes.

Proposition 2.3.6. Ceci est équivalent au fait que pour toute arête $e : y' \to f(x)$ dans G', il existe et un sommet x' dans G et un chemin $r : x' \to^\star x$ tels que $\lfloor f^\star(r) \rfloor = \lfloor (e) \rfloor$.

Démonstration. Si e est une identité, alors on prend le chemin vide pour r. Il est évident que toute fibration de graphes satisfait la condition donnée. Réciproquement, la définition de fibration de graphes dans ce cas revient à une extension de la condition donnée au cas où e est un chemin arbitraire dans G'. On conclut par une simple induction sur la longueur de ce chemin. □

Remarque 2.3.5. La remarque *2.3.3* s'adapte au cas des bisimulations fonctionnelles faibles en passant de f à $\mathrm{fc}(f)$.

On passe maintenant au cas des bisimulations faibles éventuellement non fonctionnelles. Dans le cas fort, on a défini les relations entre deux graphes G et G' au-dessus de A comme les sous-objets du pullback $G \times_A G'$ et on a interprété les propriétés de simulation en demandant que les projections soient des fibrations de graphes.

On adapte cette approche ici, en remplaçant A par $\mathrm{fc}(A)$. Néanmoins, en général, $\mathrm{fc}(G) \times_{\mathrm{fc}(A)} \mathrm{fc}(G')$ et $\mathrm{fc}(G \times_A G')$ ne sont pas isomorphes, ce qui exige un choix de conception. On utilise ici le premier :

Définition 2.3.8. Une *simulation faible* $G \nrightarrow G'$ est une relation $R \subseteq \mathrm{fc}(G) \times_{\mathrm{fc}(A)} \mathrm{fc}(G')$ dont la première projection $R \hookrightarrow \mathrm{fc}(G) \times_{\mathrm{fc}(A)} \mathrm{fc}(G') \to \mathrm{fc}(G)$ est une fibration de graphes.

R est une *bisimulation faible* ssi ses deux projections sont des fibrations de graphes.

Concrètement, considérons $p : G \to A$ et $p' : G' \to A$, ainsi qu'une simulation faible R comme ci-dessus. Pour toute arête $r : x \leftarrow x'$ de $\mathrm{fc}(G)$, c'est-à-dire tout chemin sans identités $r : x \,{}^\star\!\!\leftarrow x'$ et $y \in G'$ tel que $R(x, y)$, la condition de fibration de graphes donne un chemin sans identités $r' : y \,{}^\star\!\!\leftarrow y'$ dans G', tel que $(r, r') \in R$. Si R est pleine, ceci est équivalent à l'existence, pour toute arête $e : x \leftarrow x'$ de G et $y \in G'$ tel que $R(x, y)$, d'un chemin sans identités $r' : y \,{}^\star\!\!\leftarrow y'$ tel que $R(x', y')$ et $\lfloor p(e) \rfloor = \lfloor (p')^\star(r') \rfloor$. On ne considère ici que des relations pleines et donc on utilise directement cette caractérisation.

Comme dans le cas fort, on a pour tous G et G' fixés au-dessus de A :

Proposition 2.3.7. Les bisimulations faibles sont fermées par union et l'union de toutes les bisimulations faibles, la *bisimilarité faible*, est la bisimulation faible maximum.

Notation 2.3.5. La bisimilarité faible au-dessus de A est notée $\approx_A$. Comme dans le cas fort, on la comprend parfois par abus de notation comme une relation d'équivalence sur tous les sommets de deux graphes arbitraires au-dessus de A.

2.4 Fins, cofins et extensions de Kan

Cette partie est une introduction aux fins et aux cofins (*end* et *coend* en anglais). Une présentation plus rigoureuse et détaillée est donnée dans Riehl [66] (en particulier dans le premier chapitre et la partie 7.1).

2.4.1 Motivation

Commençons par un exemple concret.

Définition 2.4.9. Soit Γ la catégorie librement engendrée par le graphe à deux sommets $\star$ et $[1]$ et deux arêtes $s, t : \star \to [1]$.

Il est bien connu que la catégorie des graphes (au sens des multi-graphes orientés) peut se définir comme la catégorie des préfaisceaux sur Γ :

Proposition 2.4.8. On a une équivalence de catégories $\mathrm{Gph} \simeq \widehat{\Gamma}$.

Tout graphe G est donc un foncteur $\Gamma^{\mathrm{op}} \to \mathrm{Set}$, dont on peut se demander quelles sont la limite et la colimite. La limite de G ne va pas nous occuper longtemps ici ; si G est le graphe $\sigma, \tau : E \rightrightarrows V$, on obtient l'égalisateur de σ et τ, c'est-à-dire l'ensemble des arêtes e telles que $\sigma(e) = \tau(e)$: les boucles.

La colimite est le quotient des sommets par la plus petite relation d'équivalence identifiant deux sommets v et v' dès qu'il existe $e \in E$ telle que $\sigma(e) = v$ et $\tau(e) = v'$. On obtient ainsi l'ensemble des composantes connexes de G.

Les colimites sont les exemples les plus familiers d'extensions de Kan à gauche, qui à leur tour se calculent à l'aide de cofins. Expliquons le lien en guise de motivation.

On verra plus bas que dire que C est une colimite de G avec λ comme cocône colimitant est équivalent à dire que le diagramme

$$\begin{tikzcd} \Gamma^{\mathrm{op}} \arrow[rr, "!"] \arrow[dr, "G"'] & \arrow[d, phantom, "\overset{\lambda}{\Rightarrow}"] & 1^{\mathrm{op}} \arrow[dl, "\ulcorner C \urcorner"] \\ & \mathrm{Set} & \end{tikzcd}$$

est une extension de Kan à gauche.

La propriété déterminante pour cela est que pour tout ensemble D et toute transformation naturelle

$$\begin{tikzcd}
\Gamma^{op} \arrow[rr, "!"] \arrow[dr, "G"'] & \arrow[d, phantom, "\overset{\gamma}{\Rightarrow}"] & 1^{op} \arrow[dl, "\ulcorner D \urcorner"] \\
& \mathrm{Set}, &
\end{tikzcd}$$

il existe une unique fonction $f : C \to D$ telle que γ se factorise en

$$\begin{tikzcd}
\Gamma^{op} \arrow[rr, "!"] \arrow[dr, "G"'] & \arrow[d, phantom, "\overset{\lambda}{\Rightarrow}"] & 1^{op} \arrow[dl, "\ulcorner C \urcorner"'] \arrow[dl, bend left, "\ulcorner f \urcorner"] \arrow[dl, bend right, "\ulcorner D \urcorner"'] \\
& \mathrm{Set}. &
\end{tikzcd}$$

Que nous dit cette propriété ? Une telle transformation naturelle γ n'est rien d'autre qu'un diagramme de la forme

$$\begin{tikzcd}
E \arrow[rr, shift left, "\sigma"] \arrow[rr, shift right, "\tau"'] \arrow[dr, "\gamma_{[1]}"'] & & V \arrow[dl, "\gamma_\star"] \\
& D &
\end{tikzcd} \tag{1}$$

tel que $\gamma_\star \circ G(s) = \gamma_{[1]}$ et $\gamma_\star \circ G(t) = \gamma_{[1]}$. Ceci revient bien sûr à dire que $\gamma_\star$ coégalise $G(s)$ et $G(t)$, donc l'extension à gauche recherchée est bien le coégalisateur de $G(s)$ et $G(t)$, autrement dit l'ensemble des composantes connexes de G.

Dans le langage des cofins, ceci s'exprime en utilisant le fait bien connu qu'une extension de Kan à gauche de la forme donnée se calcule comme la cofin

$$C = \int^{c \in \Gamma} G(c).$$

Les définitions qu'on verra plus bas entraînent que la propriété déterminante pour cela est littéralement la même que celle d'extension de Kan à gauche.

2.4.2 Définition et premiers exemples

Définition 2.4.10. Soient $\mathscr{C}$ et $\mathscr{E}$ des catégories et $F : \mathscr{C}^{op} \times \mathscr{C} \to \mathscr{E}$ un foncteur. Un *coin* pour F est constitué d'un objet $E \in \mathrm{ob}(\mathscr{E})$ muni d'une famille de morphismes $\lambda_C : E \to F(C, C)$ pour tout $C \in \mathrm{ob}(\mathscr{C})$ telle que pour tout $f : C \to C'$ le carré suivant commute :

$$\begin{tikzcd}
E \arrow[r, "\lambda_C"] \arrow[d, "\lambda_{C'}"'] & F(C, C) \arrow[d, "F(C,f)"] \\
F(C', C') \arrow[r, "F(f,C')"'] & F(C, C').
\end{tikzcd}$$

Un morphisme de coins $(E, \lambda) \to (E', \lambda')$ est un morphisme $m : E \to E'$ telle que pour tout $C \in \mathrm{ob}(\mathscr{C})$ on ait $\lambda'_C \circ m = \lambda_C$.

Proposition 2.4.9. Les coins pour F et leurs morphismes forment une catégorie $\mathrm{Coins}(F)$.

Définition 2.4.11. Une *fin* pour F, notée $\int_{C \in \mathscr{C}} F(C, C)$, est un objet terminal dans $\mathrm{Coins}(F)$.

Avant de présenter des exemples, donnons la définition duale des cofins.

Définition 2.4.12. Un *cocoin* pour F est constitué d'un objet $E \in \mathrm{ob}(\mathscr{E})$ et d'une famille de morphismes $\lambda : F(C, C) \to E$ pour tout $C \in \mathrm{ob}(\mathscr{C})$, telle que le carré suivant commute :

$$\begin{tikzcd}
F(C', C) \arrow[r, "F(C,f)"] \arrow[d, "F(f,C')"'] & F(C', C') \arrow[d, "\lambda_{C'}"] \\
F(C, C) \arrow[r, "\lambda_C"'] & E.
\end{tikzcd}$$

Un morphisme de cocoins $(E, \lambda) \to (E', \lambda')$ est un morphisme $m : E \to E'$ telle que pour tout $C \in \mathrm{ob}(\mathscr{C})$ on ait $m \circ \lambda_C = \lambda'_C$.

Proposition 2.4.10. Les cocoins pour F et leurs morphismes forment une catégorie $\mathrm{CoCoins}(F)$.

Définition 2.4.13. Une *cofin* pour F, notée $\int^{C \in \mathscr{C}} F(C, C)$, est un objet initial dans $\mathrm{CoCoins}(F)$.

Exemple 2.4.2. L'ensemble $\mathrm{Cat}(\mathscr{C}, \mathscr{D})(F, G)$ des transformations naturelles entre deux foncteurs $F, G : \mathscr{C} \to \mathscr{D}$, pour toutes catégories $\mathscr{C}$ et $\mathscr{D}$, peut s'exprimer comme la fin

$$\int_{C \in \mathscr{C}} \mathscr{D}(F(C), G(C)).$$

En effet, la condition de compatibilité sur les coins devient

$$\begin{tikzcd}
\int_C \mathscr{D}(F(C), G(C)) \arrow[r, "\lambda_C"] \arrow[d, "\lambda_{C'}"'] & \mathscr{D}(F(C), G(C)) \arrow[d, "\mathscr{D}(F(C),G(f))"] \\
\mathscr{D}(F(C'), G(C')) \arrow[r, "\mathscr{D}(F(f),G(C'))"'] & \mathscr{D}(F(C), G(C')).
\end{tikzcd}$$

Le cône λ donne ainsi pour chaque C et $\alpha \in \int_C \mathscr{D}(F(C), G(C))$ un morphisme $\lambda_C(\alpha) : F(C) \to G(C)$ tel que $G(f) \circ \lambda_C(\alpha) = \lambda_{C'}(\alpha) \circ F(f)$, c'est-à-dire précisément une transformation naturelle $F \to G$ dont la composante en C est donnée par $\lambda_C(\alpha)$.

Exemple 2.4.3. On a donc que $\int^{c \in \Gamma} G(c)$ est le cocoin initial pour G. Ici, G est vu comme le foncteur $G' : \Gamma^{op} \times \Gamma \to \mathrm{Set}$ défini par composition :

$$\Gamma^{op} \times \Gamma \xrightarrow{\pi} \Gamma^{op} \xrightarrow{G} \mathrm{Set}\,.$$

Puisque G' est trivial en son second argument, un cocoin est juste un ensemble E muni de deux fonctions $\lambda_\star : G(\star) \to E$ et $\lambda_{[1]} : G([1]) \to E$, telles que $\lambda_\star \circ G(f) = \lambda_{[1]}$, pour $f \in \{s, t\}$. On retombe bien sur le diagramme *(1)*.

Exemple 2.4.4. Dans les modèles de préfaisceaux [41, 11], on interprète les processus concurrents d'un langage donné par des préfaisceaux sur une certaine catégorie dite de « chemins ». Selon le degré de concurrence interne qu'on souhaite modéliser, plusieurs choix sont possibles, mais pour le présent exemple on se contente du plus simple : soit $\mathbb{T}$ la catégorie des *traces* sur un alphabet $\mathscr{A}$ donné, qui a pour objets l'ensemble $\mathscr{A}^\star = \sum_n \mathscr{A}^n$ des mots sur $\mathscr{A}$ et comme ensemble de morphismes $\mathscr{A}(T, T')$ le singleton 1 si T est un préfixe de T' (ce qu'on note $T \leqslant T'$) et l'ensemble vide sinon.

On peut interpréter la célèbre machine à café de Milner, donnée par le système de transitions suivant :

$$\begin{tikzcd}
 & \star \arrow[d, "a"'] & \\
 & x \arrow[dl, "b"'] \arrow[dr, "c"] & \\
y \arrow[d, "d"'] & & y' \arrow[d, "e"] \\
z & & z',
\end{tikzcd}$$

où on pense aux actions comme ceci :

- a = « recevoir 1€ »,
- b = « recevoir un signal sur le bouton café »,
- c = « recevoir un signal sur le bouton thé »,
- d = « faire un gobelet de café »,
- e = « faire un gobelet de thé ».

Comme il n'y a ici aucun branchement non-déterministe, on aimerait interpréter ce processus par un préfaisceau envoyant chaque trace acceptée sur le singleton. On peut le faire à la main, mais on va voir que les cofins offrent une formule synthétique. Un premier

essai pourrait utiliser les sommes : on définit la famille de traces $(T_i)_{i\in 2}$ par $T_1 = abd$ et $T_2 = ace$ et on pose

$$F_1(T) = \sum_{i\in 2} \mathbb{T}(T, T_i).$$

Ainsi, sur toute trace T préfixe de T_1 ou T_2, F_1 est non vide, alors qu'il est vide sur toutes les autres traces. Pourtant, on n'y est pas tout à fait, puisque lorsque $T = a$ on obtient $F_1(a) = 1+1 = 2$, qui n'est certes pas un singleton : on aimerait identifier les deux manières d'accepter la trace a. Il suffit pour cela de considérer le *span ambulant,* c'est-à-dire la catégorie Λ engendrée par le graphe $-1 \leftarrow 0 \rightarrow 1$, et le préfaisceau $G : \Lambda^{op} \rightarrow \mathrm{Set}$ constant égal à 1. On définit ensuite le plongement $P : \Lambda \rightarrow \mathbb{T}$ défini par le span

$$abd \leftarrow a \rightarrow ace,$$

puis

$$F(T) = \int^{c\in\Lambda} G(c)\times\mathbb{T}(T, P(c)) = \int^{c} \mathbb{T}(T, P(c)).$$

Ainsi, si T est préfixe d'un seul des T_i, $F(T)$ est un singleton comme précédemment, mais cette fois si T est préfixe à la fois de T_1 et T_2, alors T est aussi préfixe de a et les trois éléments

- $(\star, T \rightarrow abd) \in G(-1)\times\mathbb{T}(T, P(-1))$,
- $(\star, T \rightarrow ace) \in G(1)\times\mathbb{T}(T, P(1))$ et
- $(\star, T \rightarrow a) \in G(0)\times\mathbb{T}(T, P(0))$

sont identifiés parce que les deux diagrammes ci-dessous doivent commuter :

$$\begin{tikzcd}
 & G(-1)\times\mathbb{T}(T, a) \arrow[dl, "G(!)\times\mathbb{T}(T,P(!))"'] \arrow[dr, "G(!)\times id"] & \\
G(-1)\times\mathbb{T}(T, abd) \arrow[dr, bend right, "\lambda_{-1}"'] & & G(0)\times\mathbb{T}(T, a) \arrow[dl, bend left, "\lambda_0"] \\
 & \int^{c\in\Lambda} G(c)\times\mathbb{T}(T, P(c)) &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & G(1)\times\mathbb{T}(T, a) \arrow[dl, "G(!)\times\mathbb{T}(T,P(!))"'] \arrow[dr, "G(!)\times id"] & \\
G(1)\times\mathbb{T}(T, ace) \arrow[dr, bend right, "\lambda_{1}"'] & & G(0)\times\mathbb{T}(T, a) \arrow[dl, bend left, "\lambda_0"] \\
 & \int^{c\in\Lambda} G(c)\times\mathbb{T}(T, P(c)). &
\end{tikzcd}$$

2.4.3 Expression en termes de (co)limites ordinaires

Le résultat suivant devrait être clair :

Proposition 2.4.11. Pour tout $F : \mathscr{C}^{op}\times\mathscr{C} \rightarrow \mathscr{E}$ avec $\mathscr{E}$ une catégorie de préfaisceaux, la cofin $\int^{C} F(C, C)$ existe et est le coégalisateur du diagramme

$$\sum_{C, C', f : C \rightarrow C'} F(C', C) \underset{[\mathrm{inj}_C F(f, C)]_{C, C', f}}{\overset{[\mathrm{inj}_{C'} F(C', f)]_{C, C', f}}{\rightrightarrows}} \sum_C F(C, C).$$

Démonstration. Les colimites se calculent point-à-point dans les catégories de préfaisceaux et lorsque $\mathscr{E} = \mathrm{Set}$ la cofin en question est le quotient de $\sum_C F(C, C)$ par la relation d'équivalence identifiant (C, x) et (D, y) dès qu'il existe $z \in F(D, C)$ tel que $F(f, C)(z) = x$ et $F(D, f)(z) = y$. □

De même :

Proposition 2.4.12. La fin $\int_C F(C, C)$ est l'égalisateur du diagramme

$$\prod_C F(C, C) \underset{\langle F(C,f)\circ\pi_C\rangle_{C,C',f}}{\overset{\langle F(f,C')\circ\pi_{C'}\rangle_{C,C',f}}{\rightrightarrows}} \prod_{C,C',f:C\to C'} F(C, C').$$

Démonstration. Même argument, sauf qu'ici la fin en question est l'ensemble des familles $x \in \prod_C F(C, C)$ telles que pour tout $f : C \to C'$ on ait $F(C, f)(x_C) = F(f, C')(x_{C'})$. □

Les caractérisations précédentes présentent l'inconvénient de reposer sur les produits et coproduits. MacLane [55] montre qu'on peut s'en passer (dans le cas général). Pour cela, on définit pour chaque catégorie $\mathscr{C}$ la catégorie $\mathrm{Wed}(\mathscr{C})$ ayant pour ensemble d'objets $\mathrm{ob}(\mathscr{C}) + \mathrm{mor}(\mathscr{C})$ et deux morphismes $(\mathbf{l}\, C) \leftarrow (\mathbf{r}\, f) \to (\mathbf{l}\, C')$ pour chaque $f : C \to C'$ dans $\mathscr{C}$, où $\mathbf{l}$ et $\mathbf{r}$ désignent les injections de coproduit. On a de plus un foncteur de projection $\mathrm{pr}_{\mathscr{C}} : \mathrm{Wed}(\mathscr{C}) \to \mathscr{C}^{\mathrm{op}} \times \mathscr{C}$, envoyant chaque diagramme

$$(\mathbf{l}\, C) \leftarrow (\mathbf{r}\, f) \to (\mathbf{l}\, C')$$

sur

$$(C, C) \leftarrow (C', C) \to (C', C').$$

La proposition 2.4.11 entraîne clairement :

Proposition 2.4.13. Pour tout $F : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \to \mathscr{E}$ on a

$$\int^C F(C, C) \cong \mathrm{colim}(F \circ \mathrm{pr}_{\mathscr{C}}),$$

au sens où les deux existent en même temps et dans ce cas sont isomorphes dans $\mathrm{CoCoins}(F)$.

De manière duale,

Proposition 2.4.14. Pour tout $F : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \to \mathscr{E}$ on a

$$\int_C F(C, C) \cong \lim\left(F \circ \mathrm{pr}^{\mathrm{op}}_{\mathscr{C}^{\mathrm{op}}}\right),$$

au sens où les deux existent en même temps et dans ce cas sont isomorphes dans $\mathrm{Coins}(F)$.

Remarque 2.4.6. Le foncteur dont on prend la limite est ici le composé

$$\mathrm{Wed}\,(\mathscr{C}^{\mathrm{op}})^{\mathrm{op}} \xrightarrow{\mathrm{pr}^{\mathrm{op}}_{\mathscr{C}^{\mathrm{op}}}} \left((\mathscr{C}^{\mathrm{op}})^{\mathrm{op}} \times \mathscr{C}^{\mathrm{op}}\right)^{\mathrm{op}} = \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \xrightarrow{F} \mathscr{E},$$

qui envoie chaque diagramme

$$\mathbf{l}\,(C') \leftarrow \mathbf{r}\,(f) \rightarrow \mathbf{l}\,(C)$$

dans $\mathrm{Wed}\,(\mathscr{C}^{\mathrm{op}})$ sur

$$F\,(C', C') \rightarrow F\,(C, C') \leftarrow F\,(C, C).$$

Un corollaire immédiat de cette expression par limites et colimites des fins et cofins est :

Proposition 2.4.15. Tout foncteur continu préserve les fins ; tout foncteur cocontinu préserve les cofins.

Démonstration. On traite le cas des cofins. Pour tout foncteur cocontinu $G : \mathscr{E} \rightarrow \mathscr{K}$, on a

$$G\left(\int^{C} F\,(C, C)\right) \cong G\,(\mathrm{colim}\,(F \circ \mathrm{pr}_{\mathscr{C}})) \cong \mathrm{colim}\,(G \circ F \circ \mathrm{pr}_{\mathscr{C}}) \cong \int^{C} G\,(F\,(C, C)). \qquad \square$$

Un exemple de foncteur continu est donné par le lemme bien connu suivant :

Lemme 2.4.16. Tout préfaisceau représentable sur $\mathscr{C}$ préserve les limites, ou autrement dit envoie les colimites de $\mathscr{C}$ sur des limites de Set.

Démonstration. On a pour tout $C \in \mathrm{ob}\,(\mathscr{C})$ et tout foncteur $F : \mathscr{D} \rightarrow \mathscr{C}$:

$$\mathscr{C}\,(\mathrm{colim}\,F, C) \cong \lim \mathscr{C}\,(F\,(-), C)$$

par propriété universelle de $\mathrm{colim}\,F$. $\square$

Corollaire 2.4.17. Pour tout $F : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \rightarrow \mathscr{E}$ et $D \in \mathscr{C}$, on a

$$\mathscr{E}\left(\int^{C} F\,(C, C), D\right) \cong \int_{C} \mathscr{E}\,(F\,(C, C), D).$$

Démonstration. On a

$$\begin{aligned} \mathscr{E}\left(\textstyle\int^{C} F\,(C, C), D\right) &\cong \mathbf{y}_D\,(\mathrm{colim}\,(F \circ \mathrm{pr}_{\mathscr{C}})) && \text{par la proposition 2.4.13} \\ &\cong \lim\left(\mathbf{y}_D \circ F^{\mathrm{op}} \circ \mathrm{pr}^{\mathrm{op}}_{\mathscr{C}}\right) && \text{par le lemme.} \end{aligned}$$

Mais on a évidemment $\mathrm{pr}_{\mathscr{C}} = \mathrm{pr}_{(\mathscr{C}^{\mathrm{op}})^{\mathrm{op}}}$, donc la dernière limite est en fait également la fin de

$$\left((\mathscr{C}^{\mathrm{op}})^{\mathrm{op}} \times \mathscr{C}^{\mathrm{op}}\right) \xrightarrow{F^{\mathrm{op}}} \mathscr{E}^{\mathrm{op}} \xrightarrow{\mathbf{y}_D} \mathrm{Set}\,,$$

qu'on écrit aussi $\int_C \mathscr{E}(F(C, C), D)$ comme souhaité. □

On a encore plus directement :

Corollaire 2.4.18. Pour tout $F : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \to \mathscr{E}$ et $D \in \mathscr{C}$, on a

$$\mathscr{E}\left(D, \int_C F(C, C)\right) \cong \int_C \mathscr{E}(D, F(C, C)).$$

On a enfin, dernier résultat avant le fameux lemme de co-Yoneda, un analogue du théorème de Fubini pour les intégrales, dont on trouvera la preuve (dans le cas général) dans MacLane :

Lemme 2.4.19. Soit un foncteur $F : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \times \mathscr{D}^{\mathrm{op}} \times \mathscr{D} \to \mathscr{E}$. Si la cofin $\int^D F(C, C', D, D)$ existe pour toute paire $(C, C') \in \mathrm{ob}(\mathscr{C}) \times \mathrm{ob}(\mathscr{C})$, alors tout choix de telles cofins induit un foncteur $\int^D F(C_1, C_2, D, D) : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \to \mathrm{Set}$. La famille des donnée en (C, D) par

$$F(C, C, D, D) \xrightarrow{\lambda_D^{C,C}} \int^D F(C, C, D, D) \xrightarrow{\lambda_C} \int^C \int^D F(C, C, D, D)$$

fait de $\int^C \int^D F(C, C, D, D)$ une cofin $\int^{(C,D)} F(C, C, D, D)$.

Par le lemme précédent, on a facilement :

Corollaire 2.4.20. Pour tout F :

$$\int^C \int^D F(C, C, D, D) \cong \int^D \int^C F(C, C, D, D).$$

On peut enfin poser le fameux lemme de co-Yoneda, qui affirme que tout préfaisceau est canoniquement colimite de représentables :

Lemme 2.4.21. Pour tout $K : \mathscr{C}^{\mathrm{op}} \to \mathrm{Set}$ on a

$$K \cong \int^C K(C) \times \mathbf{y}_C\,,$$

où on rappelle que $X \cdot E$ désigne le coproduit itéré de E, X fois, c'est-à-dire $\sum_{x \in X} E$ ou encore $X \times E$.

Démonstration. On montre que les préfaisceaux covariants représentables donnés par les deux termes sont isomorphes, ce qui entraîne le résultat par le lemme de Yoneda (Yoneda covariant entraîne donc co-Yoneda contravariant!). Pour cela, considérons un préfaisceau arbitraire $X : \mathscr{C}^{\mathrm{op}} \to \mathrm{Set}$. On a :

$$\begin{aligned}
& \widehat{\mathscr{C}}\left(\int^{C} K(C) \times \mathbf{y}_C, X\right) & \\
\cong\ & \int_{C'} \mathrm{Set}\left(\int^{C} K(C) \times \mathscr{C}(C', C), X(C')\right) & \text{par l'exemple 2.4.2} \\
\cong\ & \int_{C'} \int_{C} \mathrm{Set}\left(K(C) \times \mathscr{C}(C', C), X(C')\right) & \text{par le corollaire 2.4.17} \\
\cong\ & \int_{C'} \int_{C} \mathrm{Set}\left(\mathscr{C}(C', C), \mathrm{Set}(K(C), X(C'))\right) & \text{par curryfication} \\
\cong\ & \int_{C'} \mathrm{CAT}(\mathscr{C}, \mathrm{Set})\left(\mathscr{C}(C', -), \mathrm{Set}(K(-), X(C'))\right) & \text{par l'exemple 2.4.2} \\
\cong\ & \int_{C'} \mathrm{Set}(K(C'), X(C')) & \text{par le lemme de Yoneda covariant} \\
\cong\ & \widehat{\mathscr{C}}(K, X) & \text{par l'exemple 2.4.2.}
\end{aligned}$$

□

2.4.4 Lien avec les extensions de Kan

Pour conclure cette brève introduction, on revient sur le lien avec les extensions de Kan, évoqué en partie 2.4.1.

Définition 2.4.14. Pour tous foncteurs $\mathscr{E} \xleftarrow{F} \mathscr{C} \xrightarrow{G} \mathscr{D}$, une *extension à gauche de* F *le long de* G est une transformation naturelle

$$\begin{tikzcd}
\mathscr{C} \arrow[rr, "G"] \arrow[dr, "F"'] & \arrow[d, phantom, "\overset{\eta}{\Rightarrow}"] & \mathscr{D} \arrow[dl, "L"] \\
& \mathscr{E} &
\end{tikzcd}$$

telle que toute transformation naturelle α comme à gauche ci-dessous se factorise de manière unique à travers η comme à droite ci-dessous :

$$\begin{tikzcd}
\mathscr{C} \arrow[rr, "G"] \arrow[dr, "F"'] & \arrow[d, phantom, "\overset{\alpha}{\Rightarrow}"] & \mathscr{D} \arrow[dl, "H"] \\
& \mathscr{E} &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathscr{C} \arrow[rr, "G"] \arrow[dr, "F"'] & \arrow[d, phantom, "\overset{\eta}{\Rightarrow}"] & \mathscr{D} \arrow[dl, "L"] \arrow[dl, bend right, "H"'] \arrow[d, phantom, "\overset{\bar{\alpha}}{\Rightarrow}"] \\
& \mathscr{E}. &
\end{tikzcd}$$

Clairement, on a :

Proposition 2.4.22. Si des extensions de Kan à gauche (resp. droite) le long de G existent pour tous les $F : \mathscr{C} \to \mathscr{E}$, alors tout choix de telles extensions induit un foncteur $\mathrm{CAT}(\mathscr{C}, \mathscr{E}) \to \mathrm{CAT}(\mathscr{D}, \mathscr{E})$, adjoint à gauche (resp. droite) du foncteur $\mathrm{CAT}(G, \mathscr{E}) : \mathrm{CAT}(\mathscr{D}, \mathscr{E}) \to \mathrm{CAT}(\mathscr{C}, \mathscr{E})$ de précomposition par G.

Réciproquement, si de tels adjoints existent, alors ils donnent des extensions de Kan à gauche (resp. droite).

On peut montrer :

Proposition 2.4.23. Si $\mathscr{E}$ est cocomplète, L existe et on a

$$L(D) \cong \int^{C \in \mathrm{ob}(\mathscr{C})} \mathscr{D}(G(C), D) \cdot F(C).$$

Démonstration. Soit L' le foncteur défini par le membre de droite. L'assignation $F \mapsto L'$ induit bien un foncteur $\mathrm{CAT}(\mathscr{C}, \mathscr{E}) \to \mathrm{CAT}(\mathscr{D}, \mathscr{E})$, dont on vérifie qu'il est bien adjoint à gauche de la précomposition par G. Pour cela, soit $X : \mathscr{D} \to \mathscr{E}$ un foncteur arbitraire. On a :

$$\begin{aligned}
& \mathrm{CAT}(\mathscr{D}, \mathscr{E})(L', X) \\
\cong\ & \textstyle\int_D \mathscr{E}(L'(D), X(D)) && \text{par l'exemple 2.4.2} \\
\cong\ & \int_D \mathscr{E}\Big(\textstyle\int^{C \in \mathrm{ob}(\mathscr{C})} \mathscr{D}(G(C), D) \cdot F(C), X(D)\Big) && \text{par définition de } L' \\
\cong\ & \int_D \textstyle\int_C \mathscr{E}(\mathscr{D}(G(C), D) \cdot F(C), X(D)) && \text{par le corollaire 2.4.17} \\
\cong\ & \int_C \textstyle\int_D \mathscr{E}(\mathscr{D}(G(C), D) \cdot F(C), X(D)) && \text{par l'analogue du théorème de Fubini} \\
\cong\ & \int_C \textstyle\int_D \mathrm{Set}(\mathscr{D}(G(C), D), \mathscr{E}(F(C), X(D))) && \text{par propriété des coproduits} \\
\cong\ & \textstyle\int_C \mathscr{E}(F(C), X(G(C))) && \text{par le lemme de Yoneda covariant} \\
\cong\ & \mathrm{CAT}(\mathscr{C}, \mathscr{E})(F, X \circ G) && \text{par l'exemple 2.4.2.}
\end{aligned}$$

□

On note au passage la version (automatique) où on étend un foncteur contravariant, qui sera utile en partie 5.

Proposition 2.4.24. Etant donnés des foncteurs $\mathscr{E} \xleftarrow{F} \mathscr{C}^{op}$ et $\mathscr{C} \xrightarrow{G} \mathscr{D}$, si $\mathscr{E}$ est cocomplète, l'extension de Kan à gauche de F le long de G^{op}, $L : \mathscr{D}^{op} \to \mathscr{E}$, existe et on a

$$L(D) \cong \int^{C \in ob(\mathscr{C})} \mathscr{D}(D, G(C)) \cdot F(C).$$

Remarque 2.4.7. Le calcul effectué à la main dans l'exemple 2.4.4 découle automatiquement de ce résultat.

On a un résultat analogue pour les extensions de Kan à droite :

Proposition 2.4.25. Etant donnés des foncteurs $\mathscr{E} \xleftarrow{F} \mathscr{C} \xrightarrow{G} \mathscr{D}$ avec $\mathscr{E}$ complète, l'adjoint à droite R de la précomposition par G existe et on a

$$R(D) \cong \int_{C \in ob(\mathscr{C})} F(C)^{\mathscr{D}(D, G(C))},$$

où E^X désigne par définition le produit itéré de E, X fois, c'est-à-dire $\prod_{x \in X} E$.

Démonstration. Soit R' le foncteur défini par le membre de droite. L'assignation $F \mapsto R'$ induit bien un foncteur $CAT(\mathscr{C}, \mathscr{E}) \to CAT(\mathscr{D}, \mathscr{E})$, dont on vérifie qu'il est bien adjoint à droite de la précomposition par G. Pour cela, soit $X : \mathscr{D} \to \mathscr{E}$ un foncteur arbitraire. On a :

$$\begin{aligned}
& CAT(\mathscr{D}, \mathscr{E})(X, R') \\
\cong\ & \textstyle\int_D Set(X(D), R'(D)) && \text{par l'exemple 2.4.2} \\
\cong\ & \int_D Set\left(X(D), \int_{C \in ob(\mathscr{C})} F(C)^{\mathscr{D}(D, G(C))}\right) && \text{par définition de } R' \\
\cong\ & \int_D \int_C \mathscr{E}\left(X(D), F(C)^{\mathscr{D}(D, G(C))}\right) && \text{par le corollaire 2.4.18} \\
\cong\ & \int_C \int_D \mathscr{E}\left(X(D), F(C)^{\mathscr{D}(D, G(C))}\right) && \text{par l'analogue du théorème de Fubini} \\
\cong\ & \int_C \textstyle\int_D Set(\mathscr{D}(D, G(C)), \mathscr{E}(X(D), F(C))) && \text{par propriété des produits} \\
\cong\ & \textstyle\int_C \mathscr{E}(X(G(C)), F(C)) && \text{par le lemme de Yoneda} \\
\cong\ & CAT(\mathscr{C}, \mathscr{E})(X \circ G, F) && \text{par l'exemple 2.4.2.}
\end{aligned}$$

□

De même que ci-dessus, on note la version où on étend un foncteur contravariant, qui sera elle aussi utile en partie 5.

Proposition 2.4.26. Etant donnés des foncteurs $\mathscr{C} \xleftarrow{F} \mathscr{C}^{op}$ et $\mathscr{C} \xrightarrow{G} \mathscr{D}$, si $\mathscr{E}$ est complète, l'extension de Kan à droite de F le long de G^{op}, $L : \mathscr{D}^{op} \to \mathscr{E}$, existe et on a

$$R(D) \cong \int_{C \in ob(\mathscr{C})} F(C)^{\mathscr{D}(G(C),D)}.$$

2.4.5 Une application classique : réalisation géométrique et foncteur singulier

On présente ici un exemple d'application des résultats des parties précédentes en topologie algébrique. Il s'agit d'une situation classique où on considère une représentation combinatoire d'espaces topologiques, qu'on souhaite associer à la notion standard. On peut considérer par exemple les ensembles simpliciaux qui sont des préfaisceaux sur une certaine catégorie Δ, dont il nous suffira ici de savoir que les objets sont les entiers naturels. On sait de plus définir un foncteur de *réalisation géométrique* $|-| : \Delta \to \mathrm{Top}$ vers la catégorie des espaces topologiques et des fonctions continues, qui associe à chaque entier n le *simplexe* Δ_n de dimension n, dont la définition importe peu ici. Par extension de Kan à gauche, ce foncteur induit un foncteur cocontinu dit de *réalisation géométrique* $\widehat{\Delta} \to \mathrm{Top}$, lui aussi noté $|-|$, qui par la proposition 2.4.23 est défini par

$$|X| \cong \int^{n} \widehat{\Delta}(\mathbf{y}_n, X) \cdot \Delta_n \cong \int^{n} X(n) \cdot \Delta_n.$$

Ce dernier foncteur admet un adjoint à droite, le foncteur dit *singulier*, qui associe à chaque espace topologique X le préfaisceau $\mathrm{Sing}(X)$ défini par

$$\mathrm{Sing}(X)(d) = \mathrm{Top}(|d|, X).$$

Tout cela donne un joli diagramme, qu'on reproduit pour la énième fois :

$$\begin{tikzcd}
\Delta \arrow[rr, "\mathbf{y}"] \arrow[dr, "|-|"'] & & \widehat{\Delta} \arrow[dl, bend left, "|-|"] \arrow[dl, phantom, "\top"] \\
& \mathrm{Top}. \arrow[ur, bend left, "\mathrm{Sing}"] &
\end{tikzcd}$$

Dans le cas général, on considère $\mathscr{C}$ petite et $\mathscr{E}$ cocomplète et on obtient un diagramme :

$$\begin{tikzcd}
\mathscr{C} \arrow[rr, "\mathbf{y}"] \arrow[dr, "G"'] & & \widehat{\mathscr{C}} \arrow[dl, bend left, "\mathrm{lan}_{\mathbf{y}} G"] \arrow[dl, phantom, "\top"] \\
& \mathscr{E}. \arrow[ul, bend left, "\mathrm{Sing}"] &
\end{tikzcd}$$

Le foncteur de réalisation géométrique envoie alors tout préfaisceaux $X \in \widehat{\mathscr{C}}$ sur $\int^C \widehat{\mathscr{C}}(\mathbf{y}_C, X) \cdot G(C)$, c'est-à-dire par Yoneda sur $\int^C X(C) \cdot G(C)$. Le foncteur singulier est quant à lui défini par

$$\mathrm{Sing}(E)(C) = \mathscr{E}(G(C), E).$$

La propriété d'adjonction se vérifie directement de manière calculatoire à l'aide des outils rappelés en partie 2 :

$$\begin{aligned}
\mathscr{E}\left(\int^C X(C) \cdot G(C), E\right) &\cong \int_C \mathscr{E}(X(C) \cdot G(C), E) \\
&\cong \int_C \mathrm{Set}(X(C), \mathscr{E}(G(C), E)) \\
&\cong \int_C \mathrm{Set}(X(C), \mathrm{Sing}(E)(C)) \\
&\cong \widehat{\mathscr{C}}(X, \mathrm{Sing}(E)).
\end{aligned}$$

2.4.6 Lien avec les (co)limites pondérées

En partie précédente on a montré que les extensions de Kan à droite et à gauche peuvent respectivement s'exprimer comme des fins et des cofins, moyennant certaines hypothèses de (co)complétude. On conclut à présent cette partie préliminaire en montrant que ces fins et cofins peuvent aussi être vues comme des limites et colimites *pondérées*, au sens des catégories enrichies [43]. Sauf qu'ici on travaille dans le cadre non enrichi, ce qui donne une version simplifiée de la théorie, pouvant éventuellement servir de tremplin vers l'original. Une référence intermédiaire très utile dans le cas enrichi sur Cat est l'article de Kelly [44]. La présentation adoptée ici suit de près le chapitre 7 de Riehl [66].

Pour motiver la construction, commençons par observer qu'un cône de $m \in \mathscr{M}$ vers $F : \mathscr{C} \to \mathscr{M}$ n'est rien d'autre qu'une transformation naturelle

$$\begin{tikzcd}
\mathscr{C} \arrow[r, bend left, "\Delta 1"] \arrow[r, bend right, "\mathscr{M}(m, F-)"'] \arrow[r, phantom, "\Downarrow"] & \mathrm{Set}.
\end{tikzcd}$$

Ainsi, une limite de F est entièrement déterminée par un isomorphisme naturel en m :

$$\mathscr{M}(m, \lim F) \cong [\mathscr{C}, \mathrm{Set}](\Delta 1, \mathscr{M}(m, F-)).$$

De manière symétrique, une colimite pour F est entièrement déterminée par un isomorphisme naturel en m :

$$\mathscr{M}(\operatorname{colim} F, m) \cong \widehat{\mathscr{C}}(\Delta 1, \mathscr{M}(F-, m)).$$

Attention : comme le foncteur $\mathscr{M}(F-, m)$ est contravariant, on doit passer à la catégorie $\widehat{\mathscr{C}}$.

L'idée de base des limites et colimites pondérées est de généraliser ceci en introduisant un *poids* arbitraire W pour remplacer $\Delta 1$, qui sera un préfaisceau covariant dans le cas des limites et contravariant dans le cas des colimites.

Définition 2.4.15. Une *limite* de $F : \mathscr{C} \to \mathscr{M}$ pondérée par $W : \mathscr{C} \to \mathrm{Set}$ est un objet $\lim^W F$ muni d'un isomorphisme naturel en m :

$$\mathscr{M}\left(m, \lim^W F\right) \cong [\mathscr{C}, \mathrm{Set}](W, \mathscr{M}(m, F-)).$$

Définition 2.4.16. Une *colimite* de $F : \mathscr{C} \to \mathscr{M}$ pondérée par $W : \mathscr{C}^{\mathrm{op}} \to \mathrm{Set}$ est un objet $\operatorname{colim}^W F$ muni d'un isomorphisme naturel en m :

$$\mathscr{M}\left(\operatorname{colim}^W F, m\right) \cong \widehat{\mathscr{C}}(W, \mathscr{M}(F-, m)).$$

Lorsque $\mathscr{M}$ est complète (resp. cocomplète), les limites (resp. colimites) pondérées se réduisent à des fins (resp. cofins).

Lemme 2.4.27. Si $\mathscr{M}$ est cocomplète, alors $\operatorname{colim}^W F \cong \int^c W(c) \cdot F(c)$.

Démonstration. On a

$$\begin{aligned}
\mathscr{M}\left(\textstyle\int^c W(c) \cdot F(c), m\right) &\cong \textstyle\int_c \mathscr{M}(W(c) \cdot F(c), m) \\
&\cong \textstyle\int_c \mathrm{Set}(W(c), \mathscr{M}(F(c), m)) \\
&\cong \widehat{\mathscr{C}}(W, \mathscr{M}(F(-), m)).
\end{aligned}$$

□

On note au passage que ceci induit une expression différente de la proposition 2.4.23 :

Proposition 2.4.28. Dans le cadre de la proposition 2.4.23, on a

$$L(D) \cong \operatorname{colim}^{\mathrm{Sing}_G(D)} F,$$

où $\mathrm{Sing}_G : \mathscr{D} \to \widehat{\mathscr{C}}$ désigne le foncteur singulier induit par G.

Lemme 2.4.29. Si $\mathscr{M}$ est complète, alors $\lim^W F \cong \int_c F(c)^{W(c)}$.

Démonstration. On a

$$\begin{aligned}\mathscr{M}\left(m, \int_c F(c)^{W(c)}\right) &\cong \textstyle\int_c \mathrm{Set}\left(W(c), \mathscr{M}(m, F(c))\right)\\ &\cong [\mathscr{C}, \mathrm{Set}](W, \mathscr{M}(m, F-)).\end{aligned}$$

□

Comme vu plus haut, on peut bien sûr exprimer ces fins et cofins comme des limites et colimites standard. Néanmoins, la forme rencontrée ici admet une présentation alternative, particulièrement simple, grâce à la construction de Grothendieck. On traite le cas des colimites.

Etant donné un préfaisceau $W : \mathscr{C}^{\mathrm{op}} \to \mathrm{Set}$, soit $p_W : \mathrm{el}\, W \to \mathscr{C}$ la fibration discrète correspondante.

Lemme 2.4.30. On a $\mathrm{colim}^W F \cong \mathrm{colim}\left(\mathrm{el}\, W \xrightarrow{p_W} \mathscr{C} \xrightarrow{F} \mathscr{M}\right)$.

Démonstration. Par la proposition 2.4.11 et le lemme 2.4.27, on a

$$\begin{aligned}\mathrm{colim}^W F &\cong \mathrm{coeq}\left(\textstyle\sum_{C \xrightarrow{f} C'} W(C') \cdot F(C) \rightrightarrows \sum_C W(C) \cdot F(C)\right)\\ &\cong \mathrm{coeq}\left(\textstyle\sum_{C \xrightarrow{f} C'} \sum_{w' \in W(C')} F(C) \rightrightarrows \sum_C \sum_{w \in W(C)} F(C)\right)\\ &\cong \mathrm{coeq}\left(\textstyle\sum_{C \xrightarrow{f} C', w' \in W(C')} F(C) \rightrightarrows \sum_{(C,w) \in \mathrm{el}(W)} F(C)\right).\end{aligned}$$

Par ailleurs, par la caractérisation bien connue des colimites en termes de coproduits et de coégalisateurs, on a :

$$\mathrm{colim}\,(F \circ p_W) \cong \mathrm{coeq}\left(\sum_{((C,w) \xrightarrow{f} (C',w')) \in \mathrm{el}(W)} F(C) \rightrightarrows \sum_{(C,w) \in \mathrm{el}(W)} F(C)\right),$$

où les deux morphismes donnés sont définis en tout f par

- $F(C) \xrightarrow{\mathrm{inj}_{(C,w)}} \sum_{(C,w) \in \mathrm{el}(W)} F(C)$ et
- $F(C) \xrightarrow{F(f)} F(C') \xrightarrow{\mathrm{inj}_{(C',w')}} \sum_{(C,w) \in \mathrm{el}(W)} F(C)$.

Les deux coégalisateurs coïncident donc puisque la fonction entre indices

$$\begin{array}{rcl} \sum_{C,C',f\in\mathscr{C}(C,C')} W(C') & \to & \sum_{(C,w),(C',w')} \mathrm{el}(W)((C,w),(C',w')) \\ (C,C',f,w') & \mapsto & (C,(w'\cdot f),C',w',f) \end{array}$$

est bijective par définition de $\mathrm{el}(W)$. □

Ce résultat permet enfin de retrouver l'expression standard [56] d'un préfaisceau comme colimite indexée par sa catégorie d'éléments :

Corollaire 2.4.31. Tout préfaisceau $K : \mathscr{C}^{op} \to \mathrm{Set}$ est colimite de

$$\mathrm{el}\,K \xrightarrow{p_K} \mathscr{C} \xrightarrow{\mathbf{y}} \widehat{\mathscr{C}}\,.$$

Démonstration. On a par les résultats précédents :

$$K \cong \int^c K(c)\cdot \mathbf{y}_c \cong \mathrm{colim}^K \mathbf{y} \cong \mathrm{colim}\left(\mathrm{el}\,K \to \mathscr{C} \to \widehat{\mathscr{C}}\right).$$ □

3 De la récriture d'ordre supérieur à la sémantique 2-catégorique

Spostati un po', che devo fare una sostituzione.

Anonyme

3.1 Un lieur pour les gouverner tous

On s'intéresse dans cette partie à l'article LMCS [32], qui propose une approche abstraite de la méthode traditionnelle de définition de langages de programmation par une syntaxe munie d'une relation de réduction modélisant l'exécution.

Pour expliquer les origines de cette approche, revenons un peu en arrière : dans sa thèse [49], Lawvere propose une approche abstraite de la notion de théorie algébrique, fondée sur les catégories à produits finis. Par exemple, une théorie $\mathscr{T}$ constituée d'un seul opérateur binaire est modélisée par une catégorie $\mathscr{C}(\mathscr{T})$ engendrée par un objet, disons $\mathtt{t}$, et un morphisme de type $\mathtt{t}\times\mathtt{t} \to \mathtt{t}$ représentant l'opération en question. Le processus d'engendrement de $\mathscr{C}(\mathscr{T})$ est simple et le cadre fournit directement une notion de modèle : les foncteurs (préservant les produits finis) de $\mathscr{C}(\mathscr{T})$ vers une autre catégorie, typiquement les ensembles.

En gros, le travail présenté ici constitue une double extension de celui de Lawvere, pour prendre en compte d'une part la liaison de variable et d'autre part la dynamique. Le premier point vient du fait que pratiquement tous les langages de programmation utilisent des variables muettes, à commencer par le plus simple d'entre eux, le λ-calcul. Le second point fait simplement référence à la relation de réduction. En fait, ces deux extensions étaient (presque) présentes séparément dans la littérature.

Pour gérer la liaison de variables, on reprend une idée de base de la *récriture d'ordre supérieur* [62] : représenter toutes les opérations liantes à l'aide d'une seule, le λ du λ-calcul simplement typé. Au-delà de l'évident attrait Tolkienesque de l'approche, on peut aussi y voir un avantage pratique de factorisation : dans n'importe quel assistant de preuve,

il suffit d'avoir une bibliothèque efficace de λ-calcul simplement typé pour aborder la récriture d'ordre supérieur. En combinant cette représentation des variables liées et le lien bien connu entre ce λ-calcul et les catégories cartésiennes fermées, il semble naturel de passer des catégories à produits finis aux catégories cartésiennes fermées. Ceci rend en effet parfaitement compte de la syntaxe en présence de variables liées et le λ-calcul simplement typé fournit une construction automatique et explicite des programmes à partir d'une signature.

Quant à l'aspect dynamique, l'exécution, Rydeheard et Stell [67], puis Power [65], le traitent pour les systèmes de récriture (du premier ordre), en se fondant sur les 2-catégories à produits finis : les morphismes sont des programmes, les étapes de réduction étant des « morphismes entre les morphismes », appelés 2-cellules.

Ma contribution [32], présentée dans ce chapitre, a donc consisté à concilier ces deux extensions : on étudie les 2-catégories cartésiennes fermées comme cadre sémantique pour la récriture avec lieurs, c'est-à-dire la récriture d'ordre supérieur. On propose aussi une notion de signature et une construction explicite de la 2-catégorie cartésienne fermée engendrée par une signature. Les résultats principaux démontrent d'une part l'adéquation de cette construction avec la construction standard de la relation de réduction associée à un système de récriture d'ordre supérieur, et d'autre part le fait qu'elle décrit en effet une adjonction entre nos signatures et les 2-catégories cartésiennes fermées. Un bémol à apporter à cette description réside dans la notion de morphisme. Chez Lawvere, un morphisme entre deux modèles M_1 et M_2 d'une théorie $\mathscr{T}$ dans une catégorie à produits finis $\mathscr{E}$ est une transformation naturelle $\mathscr{E}(\mathscr{T}) \underset{M_2}{\overset{M_1}{\Downarrow \alpha}} \mathscr{E}$. Le point important est ici que grâce à la propriété universelle des produits, une telle transformation est automatiquement monoïdale, c'est-à-dire que $\alpha_{A\times B} = \alpha_A \times \alpha_B$ (à isomorphisme canonique près). Le problème est que la notion naïve de transformation naturelle fermée est très restrictive, à cause de la contravariance de la flèche, puisqu'elle se limite à certains isomorphismes. La question de la pertinence de cette notion de morphisme du point de vue dénotationnel reste à explorer.

Travaux connexes : la sémantique bialgébrique

De nombreux autres travaux ont proposé de mathématiser le processus de définition des langages de programmation et de leurs modèles dénotationnels, de manières très différentes. A ma connaissance, la seule approche rendant compte à la fois de la structure syntaxique des programmes (en particulier la substitution), la relation de réduction, ainsi que la notion de modèle associée est la sémantique dite *bialgébrique* initiée par Turi [74, 20, 21]. En théorie elle répond donc entièrement à la question initiale. Sauf qu'elle présente certains inconvénients. Le premier d'entre eux est que des doutes demeurent sur sa portée.

En effet, si elle rend parfaitement compte de langages simples tels que CCS [57], le π-calcul [58], un langage à peine plus compliqué, lui a posé des difficultés considérables, qui sont encore examinées aujourd'hui. La solution la plus crédible à ce jour semble être celle de Staton [70], mais elle est loin d'être élémentaire et on a du mal à imaginer que la communauté l'adopte comme outil standard. Un autre point faible de la théorie est sa capacité limitée à proposer une construction systématique pour engendrer l'ensemble des programmes, la réduction et la notion de modèle. Plus précisément, ces trois éléments découlent de la donnée d'une monade T, d'une comonade C, ainsi que d'une loi de distributivité de C sur T. Or — et la version initiale de la théorie en rend bien compte dans les cas simples — pour le sémanticien, T devrait en fait être automatiquement engendrée à partir d'une *signature*, c'est-à-dire d'une famille d'opérations de base. A ma connaissance, ceci est pour l'instant limité à des opérations sans lieurs de variables, ce qui exclut par exemple des langages aussi standards que CCS (avec création de canaux), le π-calcul, ou le λ-calcul pur.

Pour comparer mon travail avec cette approche, rappelons les deux approches catégoriques de la notion de théorie algébrique, au sens standard. L'une est celle initiée par Lawvere dans sa thèse [49], rappelée plus haut. L'autre approche modélise chaque théorie algébrique par une monade (finitaire) sur les ensembles. Dans l'exemple $\mathscr{T}$ ci-dessus, cette monade associerait à chaque ensemble X l'ensemble des termes engendrés par l'opération donnée, à variables dans X. Linton [54] a montré que ces deux approches étaient essentiellement équivalentes. La sémantique bialgébrique peut être vue comme une extension de l'approche monadique à un cadre dynamique, dans le sens où les programmes sont munis d'une relation de réduction. Mon approche aborde la question du point de vue des théories de Lawvere, en ajoutant les exponentielles aux produits finis dans les catégories considérées.

Exemple 3.1.1. Pour éviter d'exhiber d'emblée l'exemple auto-référent et donc légèrement perturbant qui vient juste après, considérons les formules en logique du premier ordre. On peut les représenter par le λ-calcul simplement typé sur deux types de base, disons t pour « terme » et f pour « formule », avec pour opérations $\mathrm{eq} : t \to t \to f$, $\mathrm{and} : f \to f \to f$, $\mathrm{or} : f \to f \to f$, $\mathrm{not} : f \to f$, $\mathrm{all} : f^t \to f$, and $\mathrm{ex} : f^t \to f$.

Il est facile de définir par induction un encodage des formules à n variables libres en des λ-termes de type f à n variables libres de type t. Par exemple, la formule $\exists\, x, x = x$ sera codée par $\mathrm{ex}\,(\lambda x\,.\,\mathrm{eq}\, x\, x)$.

On considère les λ-termes comme équivalents modulo $\beta\eta$. Les résultats habituels de décomposition des formes normales donnent le principe d'induction souhaité sur les formules. Considérons ainsi un terme $x_1 : t, \ldots, x_n : t \vdash M : f$ arbitraire. Par le lemme de la branche principale (voir section 10.3.1 de [29]), il prend exactement l'une des formes suivantes :

- $\mathrm{eq}\, x_i\, x_j$ pour $i, j \in n$,

- $\mathrm{and}\, M_1 M_2$ avec $x_1 : t, \ldots, x_n : t \vdash M_i : f$ pour $i = 1, 2$,
- $\mathrm{or}\, M_1 M_2$ avec $x_1 : t, \ldots, x_n : t \vdash M_i : f$ pour $i = 1, 2$,
- $\mathrm{not}\, M'$ avec $x_1 : t, \ldots, x_n : t \vdash M' : f$,
- $\mathrm{all}\,(\lambda x : t.\, M')$ avec $x_1 : t, \ldots, x_n : t, x : t \vdash M' : f$, ou
- $\mathrm{ex}\,(\lambda x : t.\, M')$ avec $x_1 : t, \ldots, x_n : t, x : t \vdash M' : f$.

Un dernier point : on s'autorisera des constantes d'arités plus grandes que 1. Par exemple, on aura tendance à prendre pour and et or deux constantes d'arité $f, f \vdash f$; c'est-à-dire qu'on écrira non plus $\mathrm{and}\, M_1 M_2$ mais $\mathrm{and}\,(\!| M_1, M_2 |\!)$. Non seulement cette présentation se révèle plus aisée techniquement, mais en plus elle semble indispensable à l'organisation catégorique de l'approche, que nous aborderons plus loin.

Exemple 3.1.2. Voici l'exemple auto-référent promis : il s'agit du λ-calcul pur. Pour le représenter à l'aide de constantes simplement typées, on considère un seul type de base, disons t pour « terme » et deux constantes $l : t^t \vdash t$ et $a : t, t \vdash t$. On représentera par exemple le terme $\Omega = (\lambda x.\, x\, x)(\lambda x.\, x\, x)$ par $a\,(\!|\mathrm{delta}, \mathrm{delta}|\!)$, où $\mathrm{delta} = l\,(\!|\lambda x : t.\, a\,(\!| x, x |\!)|\!)$. La fonction générale de traduction $[\![-]\!]$ est définie inductivement par :

$$\begin{aligned} [\![x]\!] &= x \\ [\![M\,N]\!] &= a\,(\!|[\![M]\!], [\![N]\!]|\!) \\ [\![\lambda x.\, M]\!] &= l\,(\!|\lambda x : t.\, [\![M]\!]|\!). \end{aligned}$$

Là encore, le lemme de la branche principale nous donne qu'un terme $x_1 : t, \ldots, x_n : t \vdash M : t$ en forme β-normale η-longue est d'exactement l'une des formes suivantes
- x_i pour $i \in n$,
- $a\,(\!| M_1, M_2 |\!)$ avec $x_1 : t, \ldots, x_n : t \vdash M_i : t$ pour $i = 1, 2$,
- $l\,(\!|\lambda x : t.\, M'|\!)$ avec $x_1 : t, \ldots, x_n : t, x : t \vdash M' : t$.

3.2 Signatures

L'idée de la récriture d'ordre supérieur est d'utiliser l'encodage précédent pour spécifier des systèmes de récriture avec lieurs. Une règle dans un tel système est donc en première approximation une paire de λ-termes simplement typés de même *typage*, c'est-à-dire de même type dans le même contexte.

Exemple 3.2.3. La β-réduction du λ-calcul pur de l'exemple *3.1.2* se représente par la paire du redex $x : t, y : t^t \vdash a\,(\!| l\,(\!| y |\!), x |\!) : t$ et du réduit $x : t, y : t^t \vdash y\, x : t$. Par exemple, si on substitue à y le codage de l'identité, $\mathrm{id} = l\,(\!|\lambda x : t.\, x|\!)$, on obtient que $a\,(\!|\mathrm{id}, x|\!)$ se réduit en $(\lambda x : t.\, x)\, x$, qui est $\beta\,\eta$-équivalent à x. On obtient donc la réduction $a\,(\!|\mathrm{id}, x|\!) \longrightarrow x$. Insistons ici sur la différence subtile mais capitale entre l'application du λ-calcul pur codé, $a\,(\!| M_1, M_2 |\!)$, et l'application du λ-calcul simplement typé hôte. La règle β codée ici dit donc

que la première, lorsque M_1 est de la forme $l\,M_1'$, se réduit en la seconde, c'est-à-dire ici en $M_1'\,M_2$.

En récriture d'ordre supérieur, une *signature* est un ensemble de telles règles, soumis à certaines contraintes ayant pour but d'assurer la décidabilité de certains problèmes, par exemple la recherche de redex ou de paires critiques. L'élaboration de notre cadre sémantique nous impose d'abandonner ces contraintes au profit d'un lien plus étroit entre les signatures et nos modèles sémantiques, les *2-catégories cartésiennes fermées*.

L'idée est donc de définir nos signatures par :

- un ensemble X_0 de *types de base*,
- pour chaque typage $\Gamma \vdash A$, où A est un type simple sur X_0 et Γ est une liste de types simples, un ensemble $X_1(\Gamma \vdash A)$ de *constantes* d'arité $\Gamma \vdash A$
- et enfin pour chaque typage $\Gamma \vdash A$ et chaque couple M, N de λ-termes simplement typés dans $\Gamma \vdash A$, un ensemble $X_2(\Gamma \vdash M \longrightarrow N : A)$ de *règles* reliant M et N.

Formalisons un peu plus :

Définition 3.2.1. Soit $\mathscr{L}_0$ la monade sur Set associant à chaque ensemble X l'ensemble des types simples sur X, défini par la grammaire

$$\mathscr{L}_0(X) \ni A ::= x \mid 1 \mid A \times B \mid B^A,$$

où $x \in X$.

On note $\mathscr{S}_0(X) = \mathscr{L}_0(X)^* \times \mathscr{L}_0(X)$ l'ensemble des *typages* sur X.

Autrement dit, $\mathscr{L}_0$ est la monade librement engendrée par l'endofoncteur envoyant X sur $1 + X \times X + X \times X$.

Définition 3.2.2. Une *1-signature* est un triplet (X_0, X_1, ar) où X_0 et X_1 sont des ensembles et ar est une fonction $X_1 \to \mathscr{S}_0(X_0)$.

Ici, X_1 est bien sûr l'ensemble des constantes et la fonction donnée associe à chacune de ces constantes son typage, ou son *arité*.

Proposition 3.2.1. Les 1-signatures forment une catégorie Sig_1 dont les morphismes $(X_0, X_1, \mathrm{ar}) \to (X_0', X_1', \mathrm{ar}')$ sont les couples de fonctions (f_0, f_1) faisant commuter le diagramme

$$\begin{tikzcd}
X_1 \arrow[r, "f_1"] \arrow[d, "\mathrm{ar}"'] & X_1' \arrow[d, "\mathrm{ar}'"] \\
\mathscr{S}_0(X_0) \arrow[r, "\mathscr{S}_0(f_0)"'] & \mathscr{S}_0(X_0').
\end{tikzcd}$$

Définition 3.2.3. Soit $\mathscr{L}_1$ la monade sur Sig_1 associant à chaque 1-signature $\Sigma = (X_0, X_1, \mathrm{ar})$ la signature $\mathscr{L}_1(\Sigma)$ avec les mêmes types de base, mais avec $\mathscr{L}_1(\Sigma)(\Gamma \vdash A)$ contenant l'ensemble des λ-termes simplement typés avec constantes dans Σ, modulo $\beta\,\eta$.

Notation 3.2.1. On comprend ici $\Gamma = (A_1, \ldots, A_n)$ comme le contexte de typage $(1 : A_1, \ldots, n : A_n)$: un terme de $\mathscr{L}_1(\Sigma)(\Gamma \vdash A)$ a ses variables libres parmi $\{1, \ldots, n\}$. Comme dans les exemples ci-dessus, on note $c(\!|M_1, \ldots, M_n|\!)$ l'application d'une constante $c \in \Sigma(\Gamma \vdash A)$ à des termes $\Delta \vdash M_i : A_i$.

Il n'est pas complètement évident que $\mathscr{L}_1$ forme une monade : précisons un peu. L'idée est que $\mathscr{L}_1(\mathscr{L}_1(\Sigma))(\Gamma \vdash A)$ contient des termes dont les constantes sont elles-mêmes des termes dans $\mathscr{L}_1(\Sigma)$. Le cas crucial de la définition de la multiplication μ de la monade repose naturellement sur la substitution :

$$\mu_\Sigma(m(\!|M_1, \ldots, M_n|\!)) = m[\mu_\Sigma(M_1), \ldots, \mu_\Sigma(M_n)],$$

où

- on note m pour le terme de $\mathscr{L}_1(\Sigma)$ servant de constante et M_i pour les termes de $\mathscr{L}_1(\mathscr{L}_1(\Sigma))$ servant d'arguments;
- les variables libres de m étant les entiers de 1 à n (voir la notation *(3.2.1)*), on note $m[m_1, \ldots, m_n]$ la substitution parallèle de ces variables par $m_1, \ldots, m_n$.

De même, l'unité η de la monade envoie une constante $c \in \Sigma(\Gamma \vdash A)$ sur le terme $c(\!|1, \ldots, n|\!)$, où $\Gamma = (A_1, \ldots, A_n)$. On est maintenant presque en mesure de définir les signatures.

Définition 3.2.4. Pour toute 1-signature Σ, soit $\Sigma_{\|}$ l'ensemble $\sum_{\Gamma, A}(\Sigma(\Gamma \vdash A))^2$.
L'assignation $\Sigma \mapsto \Sigma_{\|}$ s'étend évidemment en un endofoncteur $\mathrm{Sig}_1 \to \mathrm{Set}$.

Notation 3.2.2. On note $\Sigma(\Gamma \vdash A)$ pour $\mathrm{ar}^{-1}(\Gamma \vdash A)$. De plus, pour $M, N \in \Sigma(\Gamma \vdash A)$, on note $\Gamma \vdash M \longrightarrow N : A$ le quadruplet (Γ, A, M, N), vu comme un élément de $X_{\|}$.

L'ensemble $\Sigma_{\|}$ contient donc les couples de même typage dans Σ.

Définition 3.2.5. Une *signature* est une 1-signature $\Sigma = (X_0, X_1, \mathrm{ar})$, munie d'un ensemble X_2 de *règles* et d'une fonction $\mathrm{red} : X_2 \to \mathscr{L}_1(\Sigma)_{\|}$.

On appelle l'image par red d'une règle son *bord*.

Proposition 3.2.2. Les signatures forment une catégorie Sig dont les morphismes $(\Sigma, X_2, \mathrm{red}) \to (\Sigma', X_2', \mathrm{red}')$ sont les couples (f_2, f) où f_2 est une fonction $X_2 \to X_2'$ et $f : \Sigma \to \Sigma'$ est un morphisme de 1-signatures faisant commuter le diagramme

$$\begin{tikzcd}
X_2 \arrow[r, "f_2"] \arrow[d, "red"'] & X_2' \arrow[d, "red'"] \\
\mathscr{L}_1(\Sigma)_{\|} \arrow[r, "\mathscr{L}_1(f)_{\|}"'] & \mathscr{L}_1(\Sigma')_{\|}.
\end{tikzcd}$$

Exemple 3.2.4. On peut à présent comprendre l'exemple *3.2.3* comme une signature. La 1-signature sous-jacente correspond à l'exemple *3.1.2* : elle a un seul type de base, t, et deux constantes l et a de typages respectifs $t^t \vdash t$ et $t, t \vdash t$. On a enfin exactement une règle, qu'on appellera beta, envoyée par red sur la paire

$$x : t, y : t^t \vdash a(\!|l(\!|y|\!), x|\!) \longrightarrow y\,x : t.$$

En particulier, on verra plus bas que, tout terme clos de type t^t s'écrivant sous la forme $\lambda x : t\,.\,M$ avec $x : t \vdash M : t$, toute instance de la règle beta où x et y sont respectivement remplacés par des termes clos N et $\lambda x : t\,.\,M$ a pour réduit $\vdash (\lambda x : t\,.\,M)N : t$, qui est égal modulo β à $\vdash M[x \mapsto N] : t$.

3.3 Termes de réduction

Le but est maintenant d'engendrer à partir d'une signature arbitraire une structure représentant la relation engendrée par le système de récriture d'ordre supérieur correspondant. On procède, comme souvent autour de la correspondance de Curry-Howard, en considérant les arbres de dérivation comme des termes. Plus précisément, à partir d'une signature $S = (\Sigma, X_2, \mathrm{red})$ arbitraire on va définir une relation de réduction $\longrightarrow_S$ sur les éléments de $\mathscr{L}_1(\Sigma)$, dont on considère les arbres de preuve comme des éléments de la signature $\mathscr{L}(S)$ engendrée par S. Pour intégrer au maximum la notion de permutation de réductions indépendantes, cette relation $\longrightarrow_S$ est « parallèle », au sens de la réduction parallèle en λ-calcul. Enfin, elle est transitive, parce qu'on choisit de représenter les séquences de réductions. On donne le langage de termes associé directement avec la relation de réduction en figure 3.1 ci-dessous. Il y a un seul jugement de typage, de la forme $\Gamma \vdash P : M \longrightarrow N : A$, où Γ est une liste de variables deux à deux disjointes et toutes munies d'un type. On utilise la virgule pour la concaténation partielle de telles listes, définie seulement si les ensembles de variables utilisés dans les deux arguments sont disjoints. On construit les règles en s'assurant que si $\Gamma \vdash P : M \longrightarrow N : A$, alors $\Gamma \vdash M : A$ et $\Gamma \vdash N : A$. Pour guider l'intuition du lecteur, mentionnons que P donnera lieu à une cellule

$$\begin{tikzcd}
\Gamma \arrow[r, bend left, "M"] \arrow[r, bend right, "N"'] \arrow[r, phantom, "\Downarrow P"] & A
\end{tikzcd}$$

dans une 2-catégorie.

La première règle décrit l'instanciation d'une règle de X_2. Considérons pour commencer le cas où les P_i sont des identités, c'est-à-dire que $M_i = N_i$ pour tout $i \in n$. La règle consiste alors à remarquer que notre terme de départ, $M[M_1, \ldots, M_n]$, est obtenu à partir du membre gauche de la règle $r \in X_2$ en instanciant ses variables $1, \ldots, n$ respectivement par $M_1, \ldots, M_n$. On peut donc appliquer la règle r et le réduit obtenu est le membre droit N auquel on applique la même substitution. La possibilité donnée par la règle d'appliquer r à des réductions arbitraires $P_1, \ldots, P_n$ contribue à faire que le système représente les réductions parallèles.

$$\frac{\Gamma \vdash P_1 : M_1 \longrightarrow N_1 : \Delta_1 \quad \ldots \quad \Gamma \vdash P_n : M_n \longrightarrow N_n : \Delta_n}{\Gamma \vdash r\langle\!\langle P_1, \ldots, P_n \rangle\!\rangle : M[M_1, \ldots, M_n] \longrightarrow N[N_1, \ldots, N_n] : A}\,(r \in X_2(\Delta \vdash M \longrightarrow N : A))$$

$$\frac{\Gamma \vdash P : M_1 \longrightarrow M_2 : A \quad \Gamma \vdash Q : M_2 \longrightarrow M_3 : A}{\Gamma \vdash P\,;\,Q : M_1 \longrightarrow M_3 : A}$$

$$\frac{}{\Gamma, x : A, \Delta \vdash x : x \longrightarrow x : A} \qquad \frac{}{\Gamma \vdash () : () \longrightarrow () : 1}$$

$$\frac{\Gamma \vdash P_1 : M_1 \longrightarrow N_1 : \Delta_1 \quad \ldots \quad \Gamma \vdash P_n : M_n \longrightarrow N_n : \Delta_n}{\Gamma \vdash c(\!|P_1, \ldots, P_n|\!) : c(\!|M_1, \ldots, M_n|\!) \longrightarrow c(\!|N_1, \ldots, N_n|\!) : A}\,(c \in \Sigma(\Delta \vdash A))$$

$$\frac{\Gamma, x : A \vdash P : M \longrightarrow N : B}{\Gamma \vdash \lambda x : A\,.\,P : \lambda x : A\,.\,M \longrightarrow \lambda x : A\,.\,N : B^A}$$

$$\frac{\Gamma \vdash P : M \longrightarrow M' : B^A \quad \Gamma \vdash Q : N \longrightarrow N' : A}{\Gamma \vdash PQ : MN \longrightarrow M'N' : B}$$

$$\frac{\Gamma \vdash P : M \longrightarrow M' : A \quad \Gamma \vdash Q : N \longrightarrow N' : B}{\Gamma \vdash (P, Q) : (M, N) \longrightarrow (M', N') : A \times B}$$

$$\frac{\Gamma \vdash P : M \longrightarrow N : A \times B}{\Gamma \vdash \pi P : \pi M \longrightarrow \pi N : A}$$

$$\frac{\Gamma \vdash P : M \longrightarrow N : A \times B}{\Gamma \vdash \pi' P : \pi' M \longrightarrow \pi' N : B}$$

Figure 3.1 – Termes de réduction engendrés par S

La deuxième règle est simplement une règle de composition séquentielle. Elle correspond à la clause de clôture transitive dans la version propositionnelle de la réduction : si $M_1 \longrightarrow M_2$ et $M_2 \longrightarrow M_3$, alors $M_1 \longrightarrow M_3$. La composition des deux réductions P et Q se note $P\,;\,Q$.

Remarque 3.3.1. Dans l'article original, on annote l'opérateur de composition séquentielle ; par le terme M_2 et les opérateurs de projection π et π' par les types A et B, le tout afin de savoir retrouver facilement toute la dérivation de typage à partir du terme de réduction.

En fait on peut montrer que l'annotation de la composition séquentielle est superflue. Autrement dit, si on prend comme définition des réductions la syntaxe

$$\begin{aligned} P, Q \ ::= \ & x \mid () \mid r\langle\!\langle P_1, \ldots, P_n \rangle\!\rangle \mid (P\,;\,Q) \\ \mid \ & c(\!| P_1, \ldots, P_n |\!) \mid \lambda x : A\,.\,P \mid (P\,Q) \mid (P, Q) \mid \pi\,P \mid \pi'\,P \end{aligned}$$

et qu'on considère les règles de typage de la figure 3.1 comme un prédicat, on a :

Lemme 3.3.3. Si une réduction non typée P est typable dans un contexte Γ, alors son bord est unique ainsi que la dérivation de typage correspondante, au choix des variables introduites près.

Démonstration. Une induction facile sur P. □

En gros, on a fini ! Les règles suivantes sont des règles de congruence, qui pourraient se résumer aux deux règles

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash M : M \longrightarrow M : A} \qquad \frac{\Gamma \vdash P : M \longrightarrow N : A}{\Gamma \vdash C[P] : C[M] \longrightarrow C[N] : A}\,.$$

Dans la seconde règle, C représente un « contexte », notion pénible à définir formellement, ce que les règles de congruence de la figure 3.1 nous évitent de faire.

Remarque 3.3.2. La similitude entre les règles de formation de $r\langle\!\langle P_1, \ldots, P_n \rangle\!\rangle$ et $c(\!| P_1, \ldots, P_n |\!)$ peut entraîner une certaine confusion. Pourtant, ces deux règles ont une signification très différente. La première représente l'application d'une règle de récriture r, instanciée avec $P_1, \ldots, P_n$. La seconde représente la mise en contexte des n réductions parallèles $P_1, \ldots, P_n$, sous l'opération de base c.

Exemple 3.3.5. Considérons le λ-terme pur $(\lambda x\,.\,((\lambda y\,.\,y)\,x))\,((\lambda z\,.\,z)\,z')$ et décomposons-le en $(\lambda x\,.\,M)\,N$, où $M = (\lambda y\,.\,y)\,x$ et $N = (\lambda z\,.\,z)\,z'$. Selon la représentation de l'exemple *3.1.2*, M devient $a(\!| l(\!|\lambda y : t\,.\,y|\!), x |\!)$ et N devient $a(\!| l(\!|\lambda z : t\,.\,z|\!), z' |\!)$. Par la

première règle et les règles de congruence (pour générer les identités), on a donc une réduction

$$z' : t, x : t \vdash \mathrm{beta}\langle\!\langle \lambda y : t.\, y, x \rangle\!\rangle : [\![M]\!] \longrightarrow (\lambda y : t.\, y)\, x =_{\beta\eta} x : t.$$

Par les règles de congruence, on obtient une réduction

$$z' : t \vdash l(\!|\lambda x : t.\, \mathrm{beta}\langle\!\langle \lambda y : t.\, y, x \rangle\!\rangle|\!) : [\![\lambda x.\, M]\!] \longrightarrow l(\!|\lambda x : t.\, x|\!) = [\![\lambda x.\, x]\!] : t.$$

Par ailleurs, on a une réduction

$$z' : t \vdash \mathrm{beta}\langle\!\langle \lambda z : t.\, z, z' \rangle\!\rangle : [\![N]\!] \longrightarrow z' = [\![z']\!] : t.$$

Appelons les deux dernières réductions P et Q, respectivement. Elles donnent par congruence une réduction

$$z' : t \vdash a(\!|P, Q|\!) : [\![(\lambda x.\, M)N]\!] \longrightarrow [\![(\lambda x.\, x)z']\!] : t.$$

Ce dernier réduit rend possible une nouvelle réduction

$$z' : t \vdash \mathrm{beta}\langle\!\langle \lambda x : t.\, x, z' \rangle\!\rangle : [\![(\lambda x.\, x)z']\!] \longrightarrow z' : t.$$

En rassemblant tout ça, on obtient une réduction

$$z' : t \vdash a(\!|P, Q|\!)\,;\, \mathrm{beta}\langle\!\langle \lambda x : t.\, x, z' \rangle\!\rangle : [\![(\lambda x.\, M)N]\!] \longrightarrow z' : t.$$

3.4 Equations entre réductions

Comme on l'a vu dans l'exemple *3.3.5*, les réductions définies en section précédente permettent de représenter des réductions parallèles. Cependant, rien ne force deux réductions indépendantes à commuter.

Exemple 3.4.6. Reprenons les notations de l'exemple *3.3.5* et considérons la réduction $P' = a(\!|P, [\![N]\!]|\!) : [\![(\lambda x.\, M)N]\!] \longrightarrow [\![(\lambda x.\, x)N]\!]$ obtenue en plaçant P dans le contexte $a(\!|-, [\![N]\!]|\!)$. Considérons ensuite symmétriquement $Q' = a(\!|[\![\lambda x.\, x]\!], Q|\!) : [\![(\lambda x.\, x)N]\!] \longrightarrow [\![(\lambda x.\, x)z']\!]$. En composant séquentiellement ces deux réductions, on obtient une réduction

$$(P'; Q') : [\![(\lambda x.\, M)N]\!] \longrightarrow [\![(\lambda x.\, x)z']\!]$$

différente de $a(\!|P, Q|\!)$.

Comme on souhaite ici modéliser les réduction modulo permutation des étapes indépendantes, on ajoute des équations, dont la plus importante est peut-être l'équivalent de β pour les réductions :

$$(\lambda x : A \,.\, P)\, Q \equiv P\,[Q/x]\,. \tag{2}$$

Pour donner un sens à cette équation, il nous faut définir la substitution entre réductions. On procède en deux étapes :

- on définit d'abord la substitution d'une variable par une réduction *dans un terme,*
- puis la substitution d'une variable par un terme *dans une réduction.*

En fait, dans les deux cas, on définit directement la substitution simultanée de plusieurs variables.

Pour guider à nouveau l'intuition, ces deux opérations correspondent aux « moustaches » gauche et droite dans la 2-catégorie qu'on construira ci-dessous :

$$\begin{tikzcd} \Gamma \arrow[r, bend left, "N"] \arrow[r, bend right, "N'"'] \arrow[r, phantom, "\Downarrow Q"] & \Delta \arrow[r, "M"] & A \end{tikzcd} \qquad\qquad \begin{tikzcd} \Gamma \arrow[r, "N"] & \Delta \arrow[r, bend left, "M"] \arrow[r, bend right, "M'"'] \arrow[r, phantom, "\Downarrow P"] & A\,. \end{tikzcd}$$

Ici, $\Delta = (x_1 : \Delta_1, \ldots, x_n : \Delta_n)$, N et N' sont des n-uplets de termes $\Gamma \vdash N_i : \Delta_i$ et $\Gamma \vdash N'_i : \Delta_i$ et Q est un n-uplet de réductions $\Gamma \vdash Q_i : N_i \longrightarrow N'_i : \Delta_i$. Dans le premier cas, on remplace chaque variable x_i par N_i dans M, ce qu'on note $M\,[Q]$. Dans le second, on remplace chaque x_i par Q_i dans P, ce qu'on note $P\,[N]$.

L'idée est de définir ensuite la substitution d'une variable par une réduction dans une réduction en composant verticalement les deux 2-cellules obtenues, c'est-à-dire en formant $M\,[Q]\,;\,P\,[N]$.

La première définition est bien sûr rendue possible par les règles de congruence et procède par induction sur M, chaque cas de l'induction correspondant à une règle de congruence :

$$\begin{aligned}
()\,[Q] &= () \\
x_i\,[Q] &= Q_i \\
c\,(\!|M_1, \ldots, M_p|\!)\,[Q] &= c\,(\!|M_1[Q], \ldots, M_p[Q]|\!) \\
(\lambda x : B \,.\, M)\,[Q] &= \lambda x : B \,.\, (M\,[Q, x]) \quad ()(\text{pour } x \notin \mathrm{dom}(\Delta)) \\
(M\,N)\,[Q] &= (M\,[Q]\,N\,[Q]) \\
(M, N)\,[Q] &= (M\,[Q], N\,[Q]) \\
(\pi\, M)\,[Q] &= \pi\,(M\,[Q]) \\
(\pi'\, M)\,[Q] &= \pi'(M\,[Q])\,.
\end{aligned}$$

La seconde définition procède elle par induction sur P :

$$
\begin{aligned}
(r\langle\!\langle P_1, \ldots, P_p\rangle\!\rangle)[N] &= r\langle\!\langle P_1[N], \ldots, P_p[N]\rangle\!\rangle \\
(P_1; P_2)[N] &= (P_1[N]; P_2[N]) \\
()[N] &= () \\
x_i[N] &= N_i \\
c(\!|P_1, \ldots, P_p|\!)[N] &= c(\!|P_1[N], \ldots, P_p[N]|\!) \\
(\lambda x : B.\, P')[N] &= \lambda x : B.\, (P'[N, x]) \quad ()(\text{pour } x \notin \mathrm{dom}(\Delta)) \\
(P_1 P_2)[N] &= (P_1[N] P_2[N]) \\
(P_1, P_2)[N] &= (P_1[N], P_2[N]) \\
(\pi P')[N] &= \pi(P'[N]) \\
(\pi' P')[N] &= \pi'(P'[N]).
\end{aligned}
$$

On peut enfin définir :

Définition 3.4.6. Etant donnés un n-uplet $\Gamma \vdash Q : N \longrightarrow N' : \Delta$ de réductions et une réduction $\Delta \vdash P : M \longrightarrow M' : A$, on définit la substitution $\Gamma \vdash P[Q] : M[N] \longrightarrow M[N'] : A$ par $P[N]; M'[Q]$.

Bien sûr, on aurait ici pu choisir l'autre définition, à savoir $M[Q]; P[N']$. On peut montrer que les deux sont équivalentes après coup, c'est-à-dire modulo l'ensemble des équations.

L'équation *(2)* fait maintenant sens. Les autres, bien que nombreuses, sont en fait naturelles et faciles. On peut les classer en quatre groupes.

- **Calcul :** En plus de *(2)*, on a d'abord une règle η pour les réductions : si P est de type B^A, alors $P \equiv \lambda x : A.\, (P\, x)$. De même, on a des règles β et η pour les projections :
$$\pi(P, Q) \equiv P \qquad\qquad \pi'(P, Q) \equiv Q$$
et si P est de type $A \times B$ alors $P \equiv (\pi P, \pi' P)$. Dans le même ordre d'idées, on a que si P est de type 1 alors $P \equiv ()$.
- **Catégorie :** On a ensuite des règles assurant que la composition séquentielle munit les réductions $\Gamma \vdash M \longrightarrow N : A$ d'une structure de catégorie, pour tous Γ, M, N, A. Par exemple, grâce aux règles de congruence on a une injection des termes dans les réductions et l'image d'un terme $\Gamma \vdash M : A$ par cette injection est l'identité sur M. Autrement dit, on impose l'équation $M; P \equiv P \equiv P; M$. De même, on impose $P; (Q; R) \equiv (P; Q); R$.
- **Congruence :** On a de plus des règles de congruence, qui permettent aux équations de s'appliquer dans tout contexte. Par exemple, si $P \equiv P'$ et $Q \equiv Q'$, alors $P; Q \equiv P'; Q'$.
- **Relèvement :** Enfin, on a un groupe de règles faisant remonter la composition séquentielle vers le haut de la réduction, l'idée étant que toute réduction doit pouvoir s'écrire

comme une composition séquentielle de réductions atomiques. Typiquement, si $r \in X_2(\Delta \vdash M_1 \longrightarrow M_2 : A)$, $\Gamma \vdash P : N_1 \longrightarrow N_2 : \Delta$ et $\Gamma \vdash Q : N_2 \longrightarrow N_3 : \Delta$, alors

$$(r\langle\!\langle P \rangle\!\rangle ; M_2[Q]) \equiv r\langle\!\langle P ; Q \rangle\!\rangle \equiv (M_1[P] ; r\langle\!\langle Q \rangle\!\rangle).$$

3.5 Une monade sur les signatures

Dans les sections précédentes, on a défini les réductions engendrées par une signature $S = (X_2, \Sigma, \mathrm{red})$. En fait, ces réductions donnent lieu à une nouvelle signature :

Définition 3.5.7. Pour toute signature $S = (X_2, \Sigma, \mathrm{red})$, soit $\mathscr{L}(S)$ la signature donnée par Σ, avec comme règles $\Gamma \vdash M \longrightarrow N : A$ l'ensemble des réductions $\Gamma \vdash P : M \longrightarrow N : A$, modulo les équations survolées en section précédente, avec la fonction de bord $\mathrm{red}_{\mathscr{L}(S)}$ évidente.

Remarque 3.5.3. Attention, ici Γ n'est à strictement parler qu'une liste de formules, pas un contexte de typage. On rappelle que d'après la notation *(3.2.1)*, on considère implicitement la liste de formules $A_1, \ldots, A_n$ comme le contexte $1 : A_1, \ldots, n : A_n$.

$\mathscr{L}$ s'étend de manière évidente en un endofoncteur sur Sig et on a :

Proposition 3.5.4. $\mathscr{L}$ forme une monade sur Sig.

Démonstration. En effet, qu'est-ce qu'une réduction P dans $\mathscr{L}(\mathscr{L}(S))$? C'est une réduction dont les règles $M \longrightarrow N$ sont elles-mêmes des règles de $\mathscr{L}(S)$. Une telle réduction P s'envoie naturellement dans $\mathscr{L}(S)$ en remplaçant toutes les occurrences $Q\langle\!\langle P_1, \ldots, P_n \rangle\!\rangle$ de règles par $Q[\mu(P_1), \ldots, \mu(P_n)]$ en utilisant la substitution définie en section précédente. Ici, $Q \in \mathscr{L}(S)$ est une réduction simple, mais les P_i sont dans $\mathscr{L}(\mathscr{L}(S))$; on les envoie donc récursivement dans $\mathscr{L}(S)$. Ceci définit la multiplication de notre monade, son neutre envoyant simplement $r \in S(\Gamma \vdash M \longrightarrow N : A)$ sur $r\langle\!\langle 1, \ldots, n \rangle\!\rangle$ (où on rappelle que $\Gamma = (A_1, \ldots, A_n)$ est vu comme le contexte $1 : A_1, \ldots, n : A_n$). Le fait que ces opérations satisfont les axiomes de monade est une simple vérification. □

3.6 Algèbres et 2-catégories

La monade $\mathscr{L}$ définie en section précédente fournit directement une première notion de modèle pour chaque signature S :

Définition 3.6.8. La catégorie des *modèles naïfs* d'une signature S est la catégorie tranche $\mathscr{L}(S)/\mathscr{L}-\mathrm{Alg}$ des $\mathscr{L}$-algèbres sous $\mathscr{L}(S)$.

Ces $\mathscr{L}$-algèbres forment des 2-catégories dans le sens suivant. Pour une $\mathscr{L}$-algèbre $S = (\Sigma, X_2, \mathrm{red})$ de morphisme structurel $a_S : \mathscr{L}(S) \longrightarrow S$, on prend comme catégorie sous-jacente la catégorie syntaxique associée à $\Sigma = (X_0, X_1, \mathrm{ar})$: elle a pour objets les formules de $\mathscr{L}_0(X_0)$ et pour morphismes $A \longrightarrow B$ les éléments de $\mathscr{L}_1(S)(A \vdash B)$. La composition est donnée par substitution : étant donnés $M \in \mathscr{L}_1(S)(A \vdash B)$ and $N \in \mathscr{L}_1(S)(B \vdash C)$, on définit $N \circ M = N[M]$. La catégorie obtenue est cartésienne fermée exactement comme dans le cas standard. Enfin, étant donnés deux termes M et N dans $\mathscr{L}_1(S)(A \vdash B)$, on prend comme 2-cellules $M \longrightarrow N$ les éléments de $S(A \vdash M \longrightarrow N : B)$.

La composition verticale est donnée par l'interprétation de l'opération « ; » : étant donnés $\alpha \in S(A \vdash M_1 \longrightarrow M_2 : B)$ et $\beta \in S(A \vdash M_2 \longrightarrow M_3 : B)$, on forme la réduction $x : A \vdash (\alpha\langle\!\langle x \rangle\!\rangle ; \beta\langle\!\langle x \rangle\!\rangle) : M_1 \longrightarrow M_3 : B$ dans $\mathscr{L}(S)(A \vdash M_1 \longrightarrow M_3 : B)$ (en identifiant $A \vdash B$ avec $x : A \vdash B$ pour plus de lisibilité). On définit ensuite la composition verticale $\beta \bullet \alpha$ comme étant l'interprétation de cette réduction, i.e. :

$$\beta \bullet \alpha = a_S(\alpha\langle\!\langle x \rangle\!\rangle ; \beta\langle\!\langle x \rangle\!\rangle),$$

qui est bien dans $S(A \vdash M_1 \longrightarrow M_3 : B)$

De manière similaire, la composition horizontale est donnée par substitution : étant donnés $\alpha \in S(A \vdash M \longrightarrow M' : B)$ et $\beta \in S(B \vdash N \longrightarrow N' : C)$, on forme la réduction

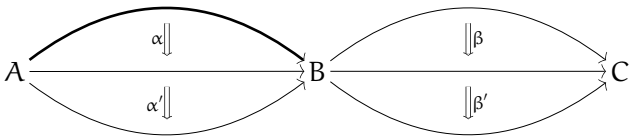
$$A \vdash \beta\langle\!\langle y \rangle\!\rangle[\alpha\langle\!\langle x \rangle\!\rangle] : N[M] \longrightarrow N'[M'] : C,$$

qui n'est autre que

$$A \vdash \beta\langle\!\langle \alpha\langle\!\langle x \rangle\!\rangle \rangle\!\rangle : N[M] \longrightarrow N'[M'] : C.$$

où on identifie $B \vdash C$ avec $y : B \vdash C$ et $A \vdash B$ avec $x : A \vdash B$, toujours pour une meilleure lisibilité.

La loi d'échange, qui stipule que pour tous



on a $(\beta' \bullet \beta) \circ (\alpha' \bullet \alpha) = (\beta' \circ \alpha') \bullet (\beta \circ \alpha)$, est une simple conséquence de la définition de la substitution et des équations de relèvement et de catégorie.

On arrive à :

Lemme 3.6.5. Les données ci-dessus définissent une 2-catégorie $\mathscr{F}(S, a_S)$ pour toute $\mathscr{L}$-algèbre (S, a_S). Cette assignation s'étend en un foncteur $\mathscr{F} : \mathscr{L}\text{–Alg} \longrightarrow 2\,\mathrm{Cat}$.

Les 2-catégories obtenues sont de plus cartésiennes fermées, dans le sens suivant.

Définition 3.6.9. Un *produit* de deux objets A et B dans une 2-catégorie $\mathscr{C}$ est un diagramme $A \xleftarrow{\pi} A \times B \xrightarrow{\pi'} B$ tel que pour tout objet C, le foncteur

$$\mathscr{C}(C, A \times B) \xrightarrow{\langle \mathscr{C}(C,\pi), \mathscr{C}(C,\pi') \rangle} \mathscr{C}(C, A) \times \mathscr{C}(C, B)$$

soit un isomorphisme.

Remarque 3.6.4. Comme cette famille de foncteurs est 2-naturelle en C, tout inverse en est aussi 2-naturelle, de sorte qu'on demande en fait un isomorphisme 2-naturel entre deux 2-foncteurs $\mathscr{C}^{op} \longrightarrow \mathrm{Cat}$.

De même :

Définition 3.6.10. Un *objet terminal* dans une 2-catégorie $\mathscr{C}$ est un objet 1 tel que l'unique foncteur $\mathscr{C}(C, 1) \longrightarrow 1$ est un isomorphisme de catégories, pour tout C.

L'argument précédent sur la 2-naturalité s'applique.

Définition 3.6.11. Une 2-catégorie *a les produits finis* ssi elle admet un objet terminal et si tout couple d'objet admet un produit dans le sens précédent.

Définition 3.6.12. Dans une 2-catégorie $\mathscr{C}$ avec produits finis, une *exponentielle* d'un objet B par un objet A est un objet B^A muni d'un morphisme $\mathrm{ev}_{A,B} : B^A \times A \longrightarrow B$ tel que le foncteur

$$\mathscr{C}\left(C, B^A\right) \longrightarrow \mathscr{C}(C \times A, B)$$

envoyant tout morphisme $f : C \longrightarrow B^A$ sur le composé

$$C \times A \xrightarrow{f \times A} B^A \times A \xrightarrow{\mathrm{ev}_{A,B}} B$$

soit un isomorphisme pour tout objet C.

Encore une fois, comme la famille de foncteurs considérée est 2-naturelle en C, on demande en fait un isomorphisme 2-naturel entre $\mathscr{C}(-, B^A)$ et $\mathscr{C}(- \times A, B)$.

Définition 3.6.13. Une 2-catégorie *cartésienne fermée* $\mathscr{C}$ est une 2-catégorie munie d'un choix d'objet terminal, ainsi que de produits et d'exponentielles pour tous couples d'objets. Un 2-foncteur entre deux 2-catégories cartésiennes fermées est dit *2-cartésien fermé* ssi il préserve exactement ces données. Les 2-catégories cartésiennes fermées forment une sous-catégorie de $2\,\mathrm{Cat}$, qu'on notera $2\,\mathrm{CCCat}$.

Remarque 3.6.5. On impose en particulier que tout foncteur 2-cartésien fermé préserve les projections et les morphismes d'évaluation $\mathrm{ev}_{A,B}$.

Lemme 3.6.6. Le foncteur $\mathscr{F}$ du lemme précédent peut se voir comme un foncteur $\mathscr{L}-\mathrm{Alg} \longrightarrow 2\,\mathrm{CCCat}$.

Ce foncteur nous donne un premier lien entre $\mathscr{L}$-algèbres et 2-catégories cartésiennes fermées. Complétons à présent ce lien en construisant un adjoint à droite à $\mathscr{F}$. Comment associer une signature à une 2-catégorie cartésienne fermée $\mathscr{C}$? Naturellement, on prend l'ensemble des objets de $\mathscr{C}$ comme types de base. Pour tout séquent $\Gamma \vdash A$ sur $\mathrm{ob}(\mathscr{C})$, on prend comme ensemble de constantes l'ensemble des morphismes $[\![\Gamma]\!]_{\mathscr{C}} \longrightarrow [\![A]\!]_{\mathscr{C}}$ dans $\mathscr{C}$, où $[\![-]\!]_{\mathscr{C}}$ est défini par induction comme suit :

$$\begin{aligned}
[\![C]\!]_{\mathscr{C}} &= C \\
[\![1]\!]_{\mathscr{C}} &= 1_{\mathscr{C}} \\
[\![A\times B]\!]_{\mathscr{C}} &= [\![A]\!]_{\mathscr{C}} \times_{\mathscr{C}} [\![B]\!]_{\mathscr{C}} \\
[\![B^A]\!]_{\mathscr{C}} &= [\![B]\!]_{\mathscr{C}}^{[\![A]\!]_{\mathscr{C}}} \\
[\![A_1, \ldots, A_n]\!]_{\mathscr{C}} &= [\![A_1\times\ldots\times A_n]\!]_{\mathscr{C}}.
\end{aligned}$$

Appelons $\mathscr{V}_1(\mathscr{C})$ la 1-signature obtenue. Par une induction similaire, cette fois sur les termes plutôt que sur les types, on obtient une structure de $\mathscr{L}_1$-algèbre sur $\mathscr{V}_1(\mathscr{C})$, qui revient au fait que le langage interne des catégories cartésiennes fermées est le λ-calcul simplement typé. Par exemple, le terme $\Gamma \vdash (M, N) : A\times B$ est interprété comme le couple

$\left\langle [\![M]\!]_{\mathscr{C}}, [\![N]\!]_{\mathscr{C}} \right\rangle : [\![\Gamma]\!]_{\mathscr{C}} \longrightarrow [\![A\times B]\!]_{\mathscr{C}}$. On utilise enfin cette structure de $\mathscr{L}_1$-algèbre pour définir l'ensemble des règles de la signature associée à $\mathscr{C}$: entre deux termes M et N dans $\mathscr{L}_1(\mathscr{V}_1(\mathscr{C}))(\Gamma \vdash A)$, on prend comme règles de réduction l'ensemble $\mathscr{C}\left([\![\Gamma]\!]_{\mathscr{C}}, [\![A]\!]_{\mathscr{C}}\right)\left([\![M]\!]_{\mathscr{C}}, [\![N]\!]_{\mathscr{C}}\right)$ des 2-cellules entre les interprétations de M et N.

Lemme 3.6.7. Ces données définissent une signature $\mathscr{V}(\mathscr{C})$, pour tout $\mathscr{C}$, assignation qui s'étend en un foncteur $\mathscr{V} : 2\,\mathrm{CCCat} \longrightarrow \mathscr{L}-\mathrm{Alg}$, adjoint à droite de $\mathscr{F}$.

La structure de $\mathscr{L}$-algèbre sur $\mathscr{V}(\mathscr{C})$ revient à interpréter les termes de réduction comme des constructions dans la 2-catégorie cartésienne fermée $\mathscr{C}$. Par exemple, $\lambda x : A\,.\,P$ devient la curryfication des 2-cellules $\mathscr{C}(C\times A, B) \to \mathscr{C}(C, B^A)$.

En combinant ce lemme avec la proposition 3.5.4, on obtient deux adjonctions composables :

$$\begin{tikzcd}
\mathrm{Sig} \arrow[r, bend left, "\mathscr{A}"] \arrow[r, phantom, "\perp"] & \mathscr{L}-\mathrm{Alg} \arrow[l, bend left, "\mathscr{U}"] \arrow[r, bend left, "\mathscr{F}"] \arrow[r, phantom, "\perp"] & 2\,\mathrm{CCCat} \arrow[l, bend left, "\mathscr{V}"]
\end{tikzcd}$$

d'où l'adjonction annoncée en début de chapitre, qu'on notera

$$\begin{tikzcd} \mathrm{Sig} \arrow[r, bend left, "\mathscr{H}"] \arrow[r, phantom, "\perp"] & 2\,\mathrm{CCCat}\,. \arrow[l, bend left, "\mathscr{W}"] \end{tikzcd}$$

3.7 Lien avec la récriture d'ordre supérieur

Dans les sections précédentes, nous avons construit une adjonction entre notre catégorie de signatures, Sig, et les 2-catégories cartésiennes fermées. Nous avions annoncé que cette adjonction fournirait un moyen de spécifier des systèmes de récriture d'ordre supérieur et d'engendrer leur relation de réduction. Nous consacrons la présente section à expliquer précisément en quel sens.

Le point de départ en est le lemme suivant :

Lemme 3.7.8. Les systèmes de récriture d'ordre supérieur de Nipkow [62] sont en bijection avec les signatures $S = (\Sigma\,, X_2\,, \mathrm{red})$ telles que pour toute règle $r \in X_2$, en posant $(\Gamma \vdash M\,, N : A) = \mathrm{red}\,(r) \in \mathscr{L}_1(\Sigma)_{\|}$, on ait

- M n'est pas une variable;
- A est un type de base;
- toute variable de Γ est libre dans M.

Ces restrictions ont pour but de rendre certains problèmes décidables. Le lemme nous permet de définir une catégorie HORS de *systèmes de récriture d'ordre supérieur,* comme la sous-catégorie pleine de Sig restreinte aux signatures satisfaisant les conditions ci-dessus.

Par ailleurs, tout système de récriture d'ordre supérieur $S = (\Sigma\,, X_2\,, \mathrm{red})$ donne lieu à une relation binaire de réduction entre termes, dont on ne considérera ici que la clôture réflexive et transitive. Intuitivement, cette relation applique les règles de X_2, éventuellement à des positions arbitraires du terme de départ. Une définition inductive possible en est donc une version « décatégorifiée » des règles de construction de $\mathscr{L}(S)$ en figure 3.1. Par exemple, la première règle devient :

$$\frac{\Gamma \vdash M_1 \longrightarrow N_1 : \Delta_1 \quad \dots \quad \Gamma \vdash M_n \longrightarrow N_n : \Delta_n}{\Gamma \vdash M[M_1, \dots, M_n] \longrightarrow N[N_1, \dots, N_n] : A}\,(r \in X_2(\Delta \vdash M \longrightarrow N : A)).$$

Définition 3.7.14. On note $\rightsquigarrow_S$ la relation obtenue.

On a de plus une certaine latitude quant aux variables libres que peuvent avoir les termes considérés. Comme notre λ-calcul simplement typé est muni de couples et de projections, on peut se restreindre à une seule variable libre. On peut donc considérer cette rela-

tion comme une relation entre morphismes parallèles de la catégorie sous-jacente à $\mathscr{H}(S)$, ce qui nous permet de construire :

Définition 3.7.15. Pour tout système de récriture d'ordre supérieur $S = (\Sigma, X_2, \mathrm{red})$, avec $\Sigma = (X_0, X_1, \mathrm{ar})$, soit $\mathscr{R}(S)$ la 2-catégorie ayant
- pour objets les types de $\mathscr{L}_0(X_0)$,
- pour morphismes $A \longrightarrow B$ les termes $A \vdash B$,
- comme ensemble de morphismes $M \longrightarrow N$, avec $A \vdash M, N : B$, le singleton si $M \rightsquigarrow_S N$ et l'ensemble vide sinon.

Par les propriétés de stabilité de $\rightsquigarrow_S$ par substitution, ainsi que par sa construction, on obtient :

Proposition 3.7.9. $\mathscr{R}(S)$ est une 2-catégorie cartésienne fermée.

De plus, par construction, pour chaque réduction $P : M \longrightarrow N$ dans $\mathscr{H}(S)$ on obtient une (unique) 2-cellule $M \rightsquigarrow_S N$ dans $\mathscr{R}(S)$. Inversement, si $M \rightsquigarrow_S N$, alors on en choisit un arbre de preuve qui fournit une réduction $M \longrightarrow N$. Il s'ensuit facilement :

Théorème 3.7.10. Pour tout système de récriture d'ordre supérieur S, il existe un 2-foncteur cartésien fermé $\mathscr{H}(S) \longrightarrow \mathscr{R}(S)$, localement plein, qui est l'identité sur les objets et les morphismes.

Comme cette assignation est évidemment naturelle, on obtient en fait :

Corollaire 3.7.11. Il existe une transformation naturelle

$$\begin{tikzcd}
\mathrm{HORS} \arrow[rr, hook] \arrow[dr, "\mathscr{R}"'] & & \mathrm{Sig} \arrow[dl, "\mathscr{H}"] \\
& 2\,\mathrm{CCCat} \arrow[u, phantom, "\Uparrow"] &
\end{tikzcd}$$

dont les composantes sont des 2-foncteurs localement pleins, identité sur les objets et les morphismes.

3.8 Une application et des perspectives

Nous avons jusqu'ici construit cadre général dont la section précédente montre qu'il fournit automatiquement, pour chaque système de récriture d'ordre supérieur S, une notion de modèle dénotationnel pour S, donnée par la tranche $\mathscr{H}(S)/2\,\mathrm{CCCat}$, dans le style de la sémantique fonctorielle de Lawvere [49]. Nous avons néanmoins annoncé que la portée

de cette construction se limitait aux langages avec lieurs dont la relation de réduction est une congruence.

Pour comprendre si on pouvait contourner cette limitation, au cours du stage de L3 d'Aurore Alcolei en 2013, nous avons effectué une étude de cas sur le λ-calcul pur en appel par valeur, Λ_v.

L'idée est de partir de deux types de base au lieu d'un : on garde le type t pour les termes, mais on considère en plus un type v pour les valeurs, en ajoutant une opération de conversion des valeurs en termes. On considère ainsi la 1-signature Σ_v donnée par $\{t, v\}$, avec comme opérations l de typage $t^v \vdash v$, a de typage $t, t \vdash t$ et r de typage $v \vdash t$. On prend ensuite une unique règle beta de type

$$x : t^v,\ y : v \vdash a\,(\!|\, r\,(\!|\, l\,(\!|\, x \,|\!)\,|\!)\,,\ r\,(\!|\, y \,|\!)\,|\!) \longrightarrow x\,y\,.$$

Du point de vue syntaxique, on obtient une correspondance précise dans le sens où les λ-termes purs à variables libres dans $\{1, \ldots, n\}$, notés $\Lambda_v(n)$, sont en bijection avec les λ-termes de typage $1 : v, \ldots, n : v \vdash t$ en forme β-normale η-longue, notés $\mathscr{L}_1(\Sigma_v)(v^n \vdash t)$. On note ϕ_n cette bijection. On a de même une bijection ϕ^V_n entre l'ensemble $\Lambda^V_v(n)$ des valeurs à variables libres dans $\{1, \ldots, n\}$ et $\mathscr{L}_1(\Sigma_v)(v^n \vdash v)$.

La situation est plus délicate pour la substitution et la réduction. On peut arguer que la notion la plus importante de substitution pour l'appel par valeur intervient lorsqu'on souhaite remplacer une variable par une valeur. En effet, c'est la seule notion de substitution utilisée par la seule règle non contextuelle de Λ_v :

$$(\lambda x.\, M)\,V \rightsquigarrow_v M[x \mapsto V]\,.$$

Pour ce cas, Aurore a montré que pour tous n et m, le diagramme suivant commute :

$$\begin{tikzcd}
\Lambda_v(n) \times \Lambda^V_v(m)^n \arrow[r, "\phi_n \times (\phi^V_m)^n"] \arrow[d, "\text{subst}"'] & \mathscr{L}_1(\Sigma_v)(v^n \vdash t) \times \mathscr{L}_1(\Sigma_v)(v^m \vdash v)^n \arrow[d, "\text{subst}"] \\
\Lambda_v(m) \arrow[r, "\phi_m"'] & \mathscr{L}_1(\Sigma_v)(v^m \vdash t)\,.
\end{tikzcd}$$

Enfin, faute de temps, le stage d'Aurore n'a permis de montrer que la correction vis-à-vis de la réduction :

Proposition 3.8.12. Pour tous n et $M, N \in \Lambda_v(n)$, si $M \rightsquigarrow_v N$ alors $\mathscr{L}(\Sigma_v)(v^n \vdash \phi_n(M) \longrightarrow \phi_n(N) : t)$ est non vide.

La complétude reste pour l'instant à prouver, mais l'exercice laisse à penser que la récriture d'ordre supérieur peut permettre, modulo de petits encodages, de représenter des stratégies de réduction évoluées. L'idéal, cependant, serait d'avoir une correspondance plus directe. Une piste pour cela consisterait à adapter l'approche décrite ici au *modèle des tuiles* de l'école Montanari [24]. Il faudrait d'abord enrichir le modèle des tuiles avec de la liaison de variable. La question semble difficile : la tentative de Bruni et Montanari [7] se solde par une absence de résultat significatif. Il semble cependant clair que la structure algébrique adaptée sera une variante des catégories doubles d'Ehresmann [17]. L'étape suivante serait de construire une adjonction entre une catégorie adéquate de signatures et ces catégories doubles.

4 De la syntaxe graphique à la sémantique catégorique

Les bâtards, c'est l'avenir de l'humanité.

Baya Ben Mahmoud
dans *Le nom des gens* de Michel Leclerc (2010).

En partie précédente, on a décrit une version abstraite de la pratique traditionnelle en sémantique qui consiste à définir une syntaxe et une relation de réduction, puis à en étudier les modèles dénotationnels.

On présente ici une approche similaire dans l'esprit, mais dédiée aux calculs graphiques et aux notions algébriques associées, tels que décrits dans l'introduction. L'idée initiale en est qu'on peut donner une présentation rigoureuse et de haut niveau, en termes de préfaisceaux [56], des dessins utilisés pour expliquer le calcul graphique. Cette présentation permet de plus de donner une définition équivalente de la notion algébrique associée, qui est en quelque sorte une version rigoureuse de la définition informelle en termes de dessins. En effet, la notion considérée est équivalente à la notion d'algèbre pour une monade *ad hoc*, construite automatiquement à partir des dessins. Cette dernière construction découle d'une généralisation commune des foncteurs *analytiques* de Joyal [40] et des foncteurs *familiaux* (ou *familialement représentables*) de Carboni et Johnstone [8, 9].

Sur le plan technique, cette partie consiste donc largement en une introduction à la théorie des foncteurs familiaux, ainsi qu'en une esquisse de l'extension que Richard Garner et moi en faisons aux foncteurs analytiques dans un article en cours de publication [26]. Côté applications, on se contente ici d'esquisser le cas très simple des *multicatégories* [50, 48].

4.1 Motivation

Avant de plonger dans la technique, commençons par motiver le sujet par des questions logiques et informatiques, mais aussi mathématiques, une fois n'est pas coutume.

Comme évoqué en introduction, le but à long terme de ce travail est d'apporter une réponse systématique aux questions de la forme : « qu'est-ce qu'un modèle pour tel calcul », en particulier lorsque le calcul considéré est intuitivement « graphique ».

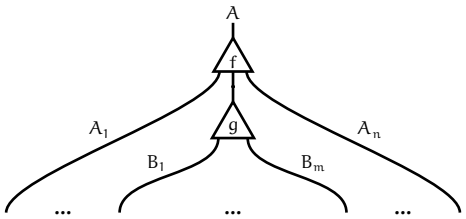
Prenons ici comme exemple principal le cas des multicatégories. Ces structures, introduites en logique par Lambek, fournissent une notion naturelle de modèle dénotationnel pour une sorte de squelette du calcul des séquents intuitionniste de Gentzen, qu'on aurait dépouillé des divers connecteurs logiques en ne gardant que la règle de coupure.

Sans entrer dans le détail, la définition formelle des multicatégories est terriblement bureaucratique — et rarement donnée dans son intégralité, encore plus dans sa version symétrique. Pourtant, elle s'explique très bien au moyen de quelques dessins [51].

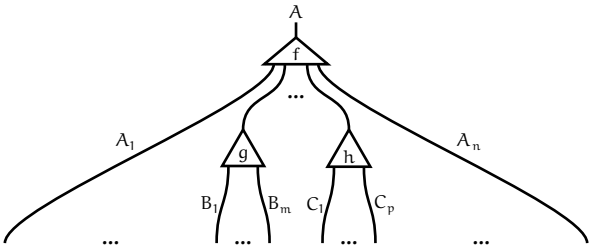
Notre point de départ consiste à donner un sens rigoureux à ces dessins, en gros des réseaux de preuve pour le squelette du calcul des séquents évoqué ci-dessus, qui servira ensuite à donner une définition alternative, à mon sens plus intuitive, des multicatégories. Bien sûr, on perd quelque chose en cours de route, ici une certaine économie dans la définition traditionnelle : établir une structure de multicatégorie sur un modèle donné sera plus difficile avec la nouvelle définition. Notre approche n'enlève donc rien au mérite de la définition traditionnelle, qu'on peut à présent voir comme un moyen efficace de construire des multicatégories.

Une multicatégorie $\mathscr{M}$ est un ensemble $\mathrm{ob}(\mathscr{M})$ d'*objets*, muni pour chaque *séquent* $(A_1, \ldots, A_n \vdash A) \in \mathrm{ob}(\mathscr{M})^\star \times \mathrm{ob}(\mathscr{M})$ d'un ensemble $\mathscr{M}(A_1, \ldots, A_n \vdash A)$ d'*opérations* de type $(A_1, \ldots, A_n \vdash A)$, d'identités $\mathrm{id}_A \in \mathscr{M}(A \vdash A)$, ainsi que de lois de composition décrites informellement en dessinant chaque opération $f \in \mathscr{M}(A_1, \ldots, A_n \vdash A)$ comme un triangle

et en voyant les lois de composition comme associant à chaque diagramme



un élément noté $f \circ_i g$ de $\mathscr{M}(A_1, \ldots, A_{i-1}, B_1, \ldots, B_m, A_{i+1}, \ldots, A_n \vdash A)$ (où i est l'indice du port dans lequel on branche g). On demande enfin que les lois de composition et les identités satisfassent des lois graphiquement évidentes. Par exemple, on demande que les deux manières de parser le dessin suivant coïncident :



Avec la notation ci-dessus, on écrit

$$(f \circ_i g)_{i-1+k+m} h = (f \circ_{i+k} h) \circ_i g$$

où $i + k$ est l'indice auquel h est branché dans f.

On se propose ici de court-circuiter cette définition intuitivement évidente mais formellement compliquée. Le point de départ est l'idée de définir les multicatégories comme algèbres d'une certaine monade sur la catégorie des préfaisceaux sur $\mathbb{M}$, où $\mathbb{M}$ est la catégorie librement engendrée par le graphe

- dont l'ensemble de sommets est $1 + \mathbb{N}$ — on note $\star$ l'élément du terme 1 et $[n]$ pour l'élément du terme de droite correspondant à chaque $n \in \mathbb{N}$,
- muni seulement d'arêtes $s_1^n, \ldots, s_n^n, t^n : \star \to [n]$ pour tout $n \in \mathbb{N}$.

On omet dans la suite les exposants lorsque le contexte lève toute ambiguïté. On note ainsi par exemple $s_i : \star \to [n]$.

Le choix de cette catégorie s'explique par le fait que $\widehat{\mathbb{M}}$ est une bonne définition pour la catégorie des *multigraphes* : chaque préfaisceau X sur $\mathbb{M}$ est doté d'un ensemble de sommets $X(\star)$ et pour chaque séquent $(A_1, \ldots, A_n \vdash A) \in X(\star)^{n+1}$, d'un ensemble de multi-arêtes donné par le pullback

$$\begin{tikzcd}
X(A_1, \ldots, A_n \vdash A) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X[n] \arrow[d, "\langle\langle X(s_i)\rangle_{i \in n}, X(t)\rangle"] \\
1 \arrow[r, "\langle \ulcorner A_1 \urcorner, \ldots, \ulcorner A_n \urcorner, \ulcorner A \urcorner \rangle"'] & X(\star)^{n+1}.
\end{tikzcd}$$

Autrement dit, $X[n]$ est l'ensemble des multi-arêtes à n entrées et c'est l'action de chaque $s_i : \star \to [n]$ et de t qui donne A_i. Les dessins utilisés ci-dessus pour esquisser la définition de multicatégorie peuvent s'interpréter comme des multigraphes. Ainsi, comme le graphe à une arête n-aire est le représentable sur $[n]$, le dessin de f seul ci-dessus est simplement le représentable sur $[n]$. De même, le dessin pour la composition $f \circ_i g$ est le pushout

$$\begin{tikzcd}
\star \arrow[r, "s_i"] \arrow[d, "t"'] & {[n]} \arrow[d, "\iota_{n,i,m}"] \\
{[m]} \arrow[r, "r_{n,i,m}"'] & {[n] \circ_i [m]}, \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

où on omet le plongement de Yoneda pour plus de lisibilité. Enfin, le diagramme pour $(f \circ_{i+k} h) \circ_i g$ est la colimite de

$$[m] \xleftarrow{t} \star \xrightarrow{s_i} [n] \xleftarrow{s_{i+k}} \star \xrightarrow{t} [p].$$

L'observation cruciale est enfin qu'on peut définir inductivement la classe tree $\subseteq \widehat{\mathbb{M}}^2$ des *arbres finis*, et ses sous-classes tree_n des arbres à n feuilles, pour chaque $n \in \mathbb{N}$. Plus précisément, pour chaque n, tree_n est constitué de morphismes $\left[[\sigma_i]_{i \in n}, \tau\right] : (n+1)\cdot\star \to T$ dans $\widehat{\mathbb{M}}$: T est l'arbre en question, chaque σ_i identifie une feuille et τ donne la racine.

On donne la définition complète pour plus de rigueur et pour montrer qu'elle reste élémentaire, mais elle est sans surprise et le lecteur pressé peut la passer sans dommage.

Définition 4.1.1. On définit la famille $\mathrm{tree}_n \subseteq \mathrm{ob}\left((n+1)\cdot\star/\widehat{\mathbb{M}}\right)$, indexée par les entiers comme étant la plus petite famille telle que

- le morphisme $\star + \star \xrightarrow{[\mathrm{id},\mathrm{id}]} \star$ est dans tree_1,
- chaque morphisme $(n+1)\cdot\star \xrightarrow{\left[[s_i^n]_{i\in n}, t^n\right]} [n]$ est dans tree_n,
- pour tous $\left[[\sigma_i]_{i\in n}, \tau\right] : (n+1)\cdot\star \to S$ dans tree_n, $\left[[\sigma'_j]_{j\in m}, \tau'\right] : (m+1)\cdot\star \to T$ dans tree_m et $i_0 \in n$, le pushout

  muni du morphisme
  $$\begin{tikzcd} \star \arrow[r, "\sigma_{i_0}"] \arrow[d, "\tau'"'] & S \arrow[d, "l_{n,i_0,m}"] \\ T \arrow[r, "r_{n,i_0,m}"'] & S \circ_{i_0} T, \arrow[ul, phantom, "\ulcorner"] \end{tikzcd}$$

  $$(n+m)\cdot\star \cong ((i_0-1)+m+(n-i_0)+1)\cdot\star \to S\circ_{i_0}T$$

  défini sur chaque composante par
  - $\left[l_{n,i_0,m}\circ\sigma_i\right]_{i\in(i_0-1)} : i_0-1 \to S\circ_{i_0}T$,
  - $\left[r_{n,i_0,m}\circ\sigma'_j\right]_{j\in m} : m \to S\circ_{i_0}T$,
  - $\left[l_{n,i_0,m}\circ\sigma_{i_0+i}\right]_{i\in(n-i_0)} : n-i_0 \to S\circ_{i_0}T$ et
  - $l_{n,i_0,m}\circ\tau : 1 \to S\circ_{i_0}T$

  est dans tree_{n+m-1}.

Ces classes induisent un endofoncteur M sur $\widehat{\mathbb{M}}$ défini en posant $M(X)(\star) = X(\star)$ et, pour tout $n \in \mathbb{N}$,

$$M(X)[n] = \sum_{t\in\mathrm{tree}_n} \widehat{\mathbb{M}}(t, X). \tag{3}$$

Pour être tout à fait précis, il ne faut prendre dans tree qu'un représentant de chaque classe d'isomorphisme. Les éléments de $M(X)[n]$ sont donc des arbres à n feuilles, étiquetés par X, c'est-à-dire que chaque sommet est étiqueté par un sommet de X et chaque multi-arête par une multi-arête de X de même arité, le tout en respectant les contraintes de « typage » imposées par X. On obtient donc sans surprise :

Théorème 4.1.1.
M est une monade et la catégorie des M-algèbres est équivalente à la catégorie des multicatégories.

On se concentrera ici sur la première partie du théorème, le fait que M soit une monade, sans contribuer à préciser la démonstration de la seconde partie (ni les quelques définitions qui manquent pour lui donner un sens), qu'on considère comme faisant partie du folklore du domaine.

Enchaînons plutôt avec la partie suivante en prenant un peu de recul. La définition inductive des arbres de la classe tree épouse parfaitement les règles de construction des réseaux de preuve squelettiques évoqués en début de partie. Si on comprend cette définition comme une version rigoureuse de ces règles, on constate que la monade M ci-dessus, et avec elle les multicatégories elles-mêmes, qui sont les modèles dénotationnels du calcul des séquents squelettique, sont engendrées directement par les réseaux.

La clef de voûte de cette construction est la forme particulière de M, qui se trouve avoir été étudiée depuis les années 90 par les catégoriciens, en tête desquels Carboni et Johnstone [8, 9].

Dans cette partie, on se demande à quel point la construction peut se généraliser et la réponse apportée est partielle. On propose d'abord une notion de foncteur analytique entre catégories de préfaisceaux, qui généralise à la fois les foncteurs familiaux entre catégories de préfaisceaux et les foncteurs analytiques sur les ensembles. On obtient enfin un résultat décrivant la construction d'une telle monade à partir d'un choix de règles de construction. Cependant, ce résultat n'est pas aussi général qu'on aurait pu l'espérer. Pour l'expliquer, commençons par rappeler superficiellement la théorie de Carboni et Johnstone.

4.2 Foncteurs familiaux sur les ensembles

Le point de départ de cette théorie est la constatation que de nombreux endofoncteurs importants sur les ensembles sont des coproduits de représentables (covariants).

Exemple 4.2.1. La monade « monoïde libre »

$$\mathscr{L}(X) = \sum_{n \in \mathbb{N}} X^n$$

peut s'écrire de manière équivalente comme

$$\mathscr{L}(X) \cong \sum_{n \in \mathbb{N}} \mathrm{Set}(n, X),$$

en voyant n comme l'ensemble $\{1, \ldots, n\}$.

En fait, cet exemple se généralise à toute signature algébrique, dans le sens suivant.

Définition 4.2.2. Une *signature algébrique* (monosortée) est une famille d'ensembles indexée par les entiers, ou, de manière équivalente, une fonction vers $\mathbb{N}$.

Intuitivement, chaque élément au-dessus de $n \in \mathbb{N}$ est vu comme une *opération* d'arité n. Chaque signature Σ engendre librement une monade T_Σ sur les ensembles, associant à chaque X l'ensemble des termes construits inductivement à partir des opérations, à variables dans X. Chaque terme dans $T_\Sigma(X)$ possède un ensemble fini d'occurrences de variables libres, qu'on peut de plus ordonner totalement, par exemple de gauche à droite dans l'arbre planaire représentant le terme en question. On obtient donc une fonction $\mathrm{vl} : T_\Sigma(X) \to \sum_n X^n$. De même, l'unique fonction $! : X \to 1$ induit une fonction $T_\Sigma(X) \xrightarrow{T_\Sigma(!)} T_\Sigma(1)$. Mieux, ces fonctions donnent un carré commutatif

$$\begin{tikzcd} T_\Sigma(X) \arrow[r] \arrow[d] & \sum_n X^n \arrow[d] \\ T_\Sigma(1) \arrow[r] & \sum_n 1^n \cong \mathbb{N}, \end{tikzcd}$$

qui se révèle en fait être un pullback. En effet, chaque terme t de $T_\Sigma(X)$ est entièrement déterminé par le terme $t' = T_\Sigma(!)(t)$ de $T_\Sigma(1)$ muni, pour chaque occurrence de variable, d'un élément de X. Si on désigne par $n = \mathrm{vl}(t')$ l'ensemble totalement ordonné de ces occurrences, donner un élément de X pour chacune revient à donner un élément de $\varphi \in X^n$. Au total, t est déterminé par la paire (t', φ). Réciproquement, étant donnée une telle paire telle que $\mathrm{vl}(t') = \left(\sum_n !^n\right)(\varphi) = |\varphi|$, on reconstruit de manière unique un terme $t \in T_\Sigma(X)$ dont les projections sont respectivement t' et φ.

On a en particulier démontré :

Proposition 4.2.2. Pour tout ensemble X, on a

$$T_\Sigma(X) \cong \sum_{t' \in T_\Sigma(1)} \mathrm{Set}(\mathrm{vl}(t'), X).$$

Ce résultat montre que T_Σ s'exprime ainsi comme un coproduit de représentables (les $\mathrm{vl}(t')$).

Définition 4.2.3. Un endofoncteur sur les ensembles est *familial* ssi il est naturellement isomorphe à un coproduit de représentables sur des ensembles finis.

Observons que pour tout T familial, disons tel que $T(X) \cong \sum_{o \in O} \mathrm{Set}(E(o), X)$ pour une fonction donnée $E : O \to \mathbb{N}$, la projection $\pi_1 : T(1) \xrightarrow{\sim} O$ est un isomorphisme. On a donc

$$T(X) \cong \sum_{x \in T(1)} \mathrm{Set}(E(\pi_1(x)), X).$$

Tout foncteur familial T est donc naturellement isomorphe à $X \mapsto \sum_{x \in T(1)} \mathrm{Set}(E'(x), X)$ pour une certaine fonction $E' : T(1) \to \mathbb{N}$.

Ces foncteurs familiaux ont plusieurs propriétés utiles. D'abord, ils sont stables par composition. En les munissant d'une notion adéquate de transformation naturelle, ils forment même une sous catégorie monoïdale de $\mathrm{CAT}(\mathrm{Set}, \mathrm{Set})$, où cette dernière est munie du tenseur donné par la composition de foncteurs. La notion en question est :

Définition 4.2.4. Soient S et T deux foncteurs familiaux sur les ensembles, avec $S(X) \cong \sum_{x \in S(1)} \mathrm{Set}(E_S(x), X)$ et $T(X) \cong \sum_{x \in T(1)} \mathrm{Set}(E_T(x), X)$. Une transformation naturelle $\alpha : S \to T$ est *familiale* si et seulement s'il existe un isomorphisme naturel

$$\begin{tikzcd} S(1) \arrow[rr, "\alpha_1"] \arrow[dr, "E_S"'] & \arrow[d, phantom, "\overset{\varphi}{\Leftarrow}"] & T(1) \arrow[dl, "E_T"] \\ & \mathrm{set}, & \end{tikzcd}$$

tel que pour tout $X \in \mathrm{Set}$,

$$\alpha_X(x, \psi : E_S(x) \to \mathrm{set}) = \left(\alpha_1(x), E_T(\alpha_1(x)) \xrightarrow{\varphi_x} E_S(x) \xrightarrow{\psi} \mathrm{set}\right),$$

où on rappelle que set est une petite catégorie équivalente à la catégorie des ensembles finis, par exemple les ordinaux finis avec comme morphismes toutes les fonctions.

Remarque 4.2.1. On aurait pu s'attendre pour cette définition à une utilisation de la catégorie discrète $\mathbb{N}$ en lieu et place de set. Cela contraindrait néanmoins la notion de familialité plus que nécessaire. Par exemple, on a une transformation naturelle $\tau : \mathscr{L} \to \mathscr{L}$ de la monade « monoïde libre » vers elle-même qui renverse son argument. Plus précisément, $\mathscr{L}(1)$ est ici $\mathbb{N}$ et $E_{\mathscr{L}}$ envoie chaque n sur lui-même vu comme un ensemble fini. On définit $\tau_1 : \mathbb{N} \to \mathbb{N}$ comme étant l'identité, mais pour chaque $n \in \mathbb{N}$, on prend $\varphi_n : n \to n$ dans set comme étant le renversement $(i \in n) \mapsto (n - i + 1)$. Ainsi, chaque liste $(x_1, \ldots, x_n) \in \mathscr{L}(X) = \sum_n X^n$, vue comme une fonction $\psi : n \to X$ est envoyée par τ_X sur la fonction $n \xrightarrow{\varphi_n} n \xrightarrow{\psi} X$, c'est-à-dire la liste $(x_n, \ldots, x_1)$.

On a facilement :

Proposition 4.2.3. Les foncteurs familiaux sur Set, muni des transformations familiales forment une sous-catégorie de $\mathrm{CAT}(\mathrm{Set}, \mathrm{Set})$, notée $\mathrm{FAM}(\mathrm{Set}, \mathrm{Set})$.

Un point crucial de la théorie des foncteurs familiaux est leur stabilité par composition, qui peut se voir comme le fait que $\mathrm{FAM}(\mathrm{Set}, \mathrm{Set})$ est une sous-catégorie *monoïdale* de $\mathrm{CAT}(\mathrm{Set}, \mathrm{Set})$. La démonstration la plus efficace passe par leur caractérisation en termes d'éléments *strictement génériques* [76].

Définition 4.2.5. Etant donné un foncteur $F : \mathscr{C} \to \mathscr{D}$ entre deux catégories, $C \in \mathscr{C}$ et $D \in \mathscr{D}$, un morphisme $s : D \to F(C)$ est *strictement générique* ssi pour tout carré comme ci-dessous il existe un unique morphisme diagonal h tel que $F(h) \circ s = t$ et $g \circ h = f$:

$$\begin{tikzcd} D \arrow[r, "t"] \arrow[d, "s"'] & F(X) \arrow[d, "F(g)"] \\ F(C) \arrow[r, "F(f)"'] \arrow[ur, dashed, "F(h)"] & F(Y). \end{tikzcd}$$

L'objet C est l'*arité* de s.

Un morphisme $f : D \to F(X)$ est *couvert* par le générique strict s ssi il existe $\varphi : C \to X$ tel que f se décompose en $D \xrightarrow{s} F(C) \xrightarrow{F(\varphi)} F(X)$.

On dit enfin que F admet des factorisations strictement génériques relativement à D si pour tout $X \in \mathscr{C}$, tout morphisme $D \to F(X)$ est couvert par un générique strict. F admet des factorisations strictement génériques (tout court) si F en admet relativement à tout objet D.

Proposition 4.2.4. Etant donné un foncteur $F : \mathscr{C} \to \mathscr{D}$, les conditions suivantes sont équivalentes :

1. F admet des factorisations strictement génériques ;
2. le foncteur $F/X : \mathscr{C}/X \to \mathscr{D}/F(X)$ induit par F admet un adjoint à gauche.

Démonstration. Etant donné $x : D \to F(X)$, son image par un adjoint à gauche L_X à F/X et l'unité reviennent précisément à une décomposition de x en $D \xrightarrow{\eta_x} F(C) \xrightarrow{F(L_X(x))} F(X)$. La propriété universelle de η_x revient précisément à sa généricité stricte. □

Définition 4.2.6. On appelle *adjoints à droite locaux* les foncteurs vérifiant les conditions équivalentes de la proposition précédente.

On a le résultat utile suivant :

Lemme 4.2.5. Soit $F : \mathscr{A} \to \widehat{\mathscr{C}}$, où $\mathscr{A}$ admet les coproduits. Si F admet des factorisations strictement génériques relativement aux représentables alors F en admet relativement aux coproduits de représentables.

Démonstration. Soit $D = \sum_{i \in I} \mathbf{y}_{c_i}$, pour un ensemble I et une famille $(c_i)_{i \in I}$ d'objets de $\mathscr{C}$ et soit $f : D \to F(X)$ un morphisme. On a par hypothèse $f = [f_i]_{i \in I}$, pour $f_i = f \circ \mathrm{inj}_i$, où $\mathrm{inj}_i : c_i \to D$ est la $i^{\text{ème}}$ injection du coproduit. F admettant des factorisations strictement génériques relativement aux représentables, chaque f_i se factorise en $c_i \xrightarrow{s_i} F(C_i) \xrightarrow{F(\varphi_i)} F(X)$ avec s_i strictement générique. On vérifie facilement que le

composé $\sum_i c_i \xrightarrow{\sum_i f_i} \sum_i F(C_i) \xrightarrow{[F(\mathrm{inj}_i)]_i} F\left(\sum_i C_i\right)$ est strictement générique ; or il couvre le morphisme de départ, via $F\left([\varphi_i]_i\right)$. □

On arrive enfin à la caractérisation attendue des foncteurs familiaux en termes de génériques stricts. Celle-ci repose sur un critère additionnel de finitarité, dans le sens suivant.

Définition 4.2.7. Un adjoint à droite local est *finitaire* ssi tous ses génériques stricts $c \to F(A)$ relativement à des représentables ont une arité (ici A, voir la définition 4.2.5) finiment présentable.

Remarque 4.2.2. Dans le cadre des adjoints à droite locaux, cette notion coïncide avec la notion standard de finitarité, qui demande que le foncteur en question préserve les colimites filtrées [2].

Proposition 4.2.6. Un foncteur F sur les ensembles est familial ssi c'est un adjoint à droite local finitaire.

Notons que cette proposition, contrairement à la précédente, est spécialisée au cas des ensembles.

Démonstration. Les ensembles forment une catégorie de préfaisceaux — sur la catégorie terminale. Il n'y a essentiellement qu'un représentable, mais tous les objets en sont des coproduits puisque $X \cong \sum_{x \in X} 1$ pour tout X. Par le lemme, F est donc un adjoint à droite local ssi il admet des factorisations strictement génériques depuis l'unique représentable. Il suffit donc de montrer que F est familial ssi il admet des factorisations strictement génériques d'arités finies relativement à 1.

Pour F admettant de telles factorisations strictement génériques, en fixant un choix de génériques stricts par classe d'isomorphisme, disons $\mathrm{Gen}(F)$, on a facilement que la fonction associant à chaque générique strict $1 \to F(C)$ de $\mathrm{Gen}(F)$ son composé avec

$F(C) \xrightarrow{F(!)} F(1)$ induit une bijection $\mathrm{Gen}(F) \xrightarrow{\sim} F(1)$. Toujours pour de tels foncteurs, la fonction $\sum_{(s:1\to F(C))\in \mathrm{Gen}(F)} \mathrm{Set}(C, X) \to F(X)$ envoyant (s, φ) sur $1 \xrightarrow{s} F(C) \xrightarrow{F(\varphi)} F(X)$ s'avère donc aussi bijective. On a montré

$$F(X) \cong \sum_{(s:1\to F(C))\in \mathrm{Gen}(F)} \mathrm{Set}(C, X) \cong \sum_{x\in F(1)} \mathrm{Set}(E(x), X),$$

où $E : F(1) \to \mathrm{set}$ désigne la fonction associant à chaque x le C de sa factorisation strictement générique $1 \xrightarrow{s_x} F(C) \to F(1)$. On obtient donc qu'un adjoint à droite local finitaire est forcément familial.

Réciproquement, si F est familial, on peut par construction factoriser tout $(x, \varphi) : 1 \to F(X)$, avec $x \in F(1)$ et $\varphi : E_F(x) \to X$, en $1 \xrightarrow{(x,\mathrm{id})} F(E_F(x)) \xrightarrow{F(\varphi)} F(X)$. Pour montrer que F est un adjoint à droite local finitaire, comme $E_F(x)$ est fini, il suffit de montrer que (x, id) est générique. Pour cela, considérons un carré commutatif

$$\begin{tikzcd}
1 \arrow[r, "{(y,\psi)}"] \arrow[d, "{(x,\mathrm{id})}"'] & F(Y) \arrow[d, "F(f)"] \\
F(E_F(x)) \arrow[r, "F(\varphi)"'] & F(Z).
\end{tikzcd}$$

La commutativité de ce carré signifie par définition que $(x, f\circ\psi) = (y, \varphi)$, avec $\psi : E_F(y) \to Y$. En particulier, on a $x = y$, donc $E_F(x) = E_F(y)$, et donc $f\circ\psi = \varphi$ fait bien sens. Mais du coup, il est évident que ψ lui-même fournit le relèvement attendu, qui est bien unique, ce qui conclut la démonstration. □

Revenons à présent à :

Lemme 4.2.7. Les adjoints à droite locaux sont stables par composition.

Démonstration. Considérons $F : \mathscr{C} \to \mathscr{D}$ et $G : \mathscr{D} \to \mathscr{E}$ deux adjoints à droite locaux. On montre que GF est un adjoint à droite local. Pour cela, considérons un morphisme arbitraire $f : E \to GF(X)$. Comme G est un adjoint à droite local, on peut factoriser f en $E \xrightarrow{s} G(D) \xrightarrow{G(\varphi)} G(F(X))$ avec s générique pour G. Comme F est un adjoint à droite local, on peut factoriser φ en $D \xrightarrow{t} F(C) \xrightarrow{F(\psi)} F(X)$ avec t générique pour F. Il suffit donc de montrer que les composés de la forme $E \xrightarrow{s} G(D) \xrightarrow{G(t)} G(F(C))$ sont génériques pour GF. Considérons donc un carré comme l'extérieur de

$$\begin{tikzcd}
E \arrow[r, "f"] \arrow[d, "s"'] & GF(X) \arrow[dd, "GF(\psi)"] \\
G(D) \arrow[ur, dashed, "G(g)"] \arrow[d, "G(t)"'] & \\
G(F(C)) \arrow[r, "GF(\varphi)"'] \arrow[uur, dashed, "GF(h)"'] & GF(X).
\end{tikzcd}$$

Par généricité de s pour G, on trouve un g comme indiqué, tel que $G(g) \circ s = f$ et $F(\psi) \circ g = F(\varphi) \circ t$. Par généricité de t pour F, on trouve alors h comme indiqué tel que $F(h) \circ t = g$ et $\psi \circ h = \varphi$. Ce h satisfait donc les conditions de relèvement souhaitées ; reste à montrer son unicité. Mais pour tout tel relèvement, disons h', tel que $GF(h') \circ G(t) \circ s = f$ et $\psi \circ h' = \varphi$, le composé $F(h') \circ t$ satisfait les conditions pour être un relèvement vis-à-vis de s, donc par unicité on a $F(h') \circ t = g$. Mais du coup, h' lui-même est un relèvement vis-à-vis de t et donc est égal à h. □

On a prouvé :

Proposition 4.2.8. $\mathrm{FAM}(\mathrm{Set}, \mathrm{Set})$ forme une sous-catégorie monoïdale de $\mathrm{CAT}(\mathrm{Set}, \mathrm{Set})$.

4.3 Foncteurs familiaux entre catégories de préfaisceaux

Bien sûr, les foncteurs familiaux sur les ensembles introduits en partie précédente ne suffisent pas à rendre compte d'exemples utiles tels que les multicatégories. On s'intéresse donc à les généraliser aux catégories de préfaisceaux. Le point délicat saute aux yeux : notre démonstration de la stabilité par composition passe pour les ensembles parce tout ensemble est coproduit de représentables, ce qui est faux en général dans les catégories de préfaisceaux (voir par exemple le cas des graphes).

Cependant, jusqu'ici tout va bien :

Lemme 4.3.9. Si $\mathscr{A}$ est cocomplète et le foncteur $F : \mathscr{A} \to \widehat{\mathscr{C}}$ admet des factorisations strictement génériques relativement aux représentables, alors F en admet relativement à tout préfaisceau.

Définition 4.3.8. Soit $\mathrm{Fam}(F)$ la classe des objets relativement auxquels F admet des factorisations strictement génériques.

Le lemme ci-dessus revient donc à dire que $\mathrm{Fam}(F)$ est stable par colimites arbitraires. On prouve en fait le résultat plus général :

Définition 4.3.9. Soit $\mathrm{Gen}(F)$ la sous-catégorie pleine de $\widehat{\mathscr{C}}/F$ dont les objets sont les génériques stricts.

Lemme 4.3.10. La sous-catégorie $\mathrm{Gen}(F)$ est stable par colimites dans $\widehat{\mathscr{C}}/F$.

Démonstration. La démonstration du lemme 4.2.5 s'adapte pour montrer que $\mathrm{Gen}(F)$ est stable par coproduits. Comme tout préfaisceau est colimite de représentables et toute colimite s'exprime comme coégalisateur de coproduits, il suffit de montrer que cette classe est stable par coégalisateurs. Considérons donc un coégalisateur

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, "q"] & Q \end{tikzcd}$$

avec A et B dans $\mathrm{Fam}(F)$. Soit $h : Q \to F(X)$. On choisit une factorisation strictement générique $A \xrightarrow{s} F(C) \xrightarrow{F(\varphi)} F(X)$ de $h \circ q \circ f = h \circ q \circ g$, ainsi qu'une factorisation strictement générique $B \xrightarrow{t} F(D) \xrightarrow{F(\psi)} F(X)$ de $h \circ q$ on obtient un diagramme comme l'extérieur de

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] \arrow[d, "s"'] & B \arrow[r, "t"] & F(D) \arrow[d, "F(\psi)"] \\ F(C) \arrow[rr, "F(\varphi)"'] \arrow[urr, dashed, shift left, "F(k)"] \arrow[urr, dashed, shift right, "F(l)"'] & & F(X) \end{tikzcd}$$

donc par généricité stricte de s deux morphismes k et l comme indiqué, faisant commuter le diagramme en série. Considérons maintenant un coégalisateur $\gamma : D \to E$ de k et l. Par sa propriété universelle on obtient un unique $\chi : E \to X$ tel que $\chi \circ \gamma = \psi$. De plus, $F(\gamma) \circ t$ coégalise f et g par construction, donc par propriété universelle de q on obtient un unique $\xi : Q \to F(E)$ tel que $\xi \circ q = F(\gamma) \circ t$. Enfin, on a que $F(\chi) \circ \xi \circ q = F(\psi) \circ t$ donc par unicité dans la propriété universelle de q on obtient $F(\chi) \circ \xi = h$. Au total, on obtient un diagramme commutatif

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] \arrow[d, "s"'] & B \arrow[r, "q"] \arrow[d, "t"'] & Q \arrow[d, "\xi"'] \arrow[dr, "h"] & \\ F(C) \arrow[r, shift left, "F(k)"] \arrow[r, shift right, "F(l)"'] & F(D) \arrow[r, "F(\gamma)"'] & F(E) \arrow[r, "F(\chi)"'] & F(X). \end{tikzcd}$$

Il suffit alors de montrer que pour tout diagramme

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] \arrow[d, "s"'] & B \arrow[r, "q"] \arrow[d, "t"'] & Q \arrow[d, "\xi"'] \\ F(C) \arrow[r, shift left, "F(k)"] \arrow[r, shift right, "F(l)"'] & F(D) \arrow[r, "F(\gamma)"'] & F(E), \end{tikzcd}$$

où s et t sont strictement génériques et q et γ sont respectivement des coégalisateurs dans $\widehat{\mathscr{C}}$ et $\mathscr{A}$, ξ est lui aussi strictement générique. Or, ceci découle facilement des propriétés universelles de q et γ, ainsi que de la généricité stricte de s et t. □

On peut donc développer la théorie des foncteurs familiaux entre catégories de préfaisceaux en suivant le modèle des ensembles.

Définition 4.3.10. Un foncteur $F : \mathscr{A} \to \widehat{\mathscr{D}}$ est *familial* ssi les foncteurs induits

$$F(-)(d) : \mathscr{A} \to \mathrm{Set}$$

sont des coproduits de représentables sur des préfaisceaux finiment présentables, naturellement en d.

Autrement dit, on a pour tout X et d,

$$F(X)(d) \cong \sum_{i \in I_d} \mathscr{A}(A_i, X).$$

Comme dans le cas ensembliste, on a $F(1)(d) \cong I_d$, donc on peut en fait écrire

$$F(X)(d) \cong \sum_{i \in F(1)(d)} \mathscr{A}(A_i, X).$$

Là où dans le cas ensembliste on avait une fonction $F(1) \to \mathrm{set}$, la donnée des A_i revient ici en fait à un foncteur $E_F : \mathbf{y}/F(1) \to \mathscr{A}_f$, à partir duquel on retrouve F comme

$$F(X)(d) \cong \sum_{x \in F(1)(d)} \mathscr{A}(E_F(d, x), X).$$

Exemple 4.3.2. Un exemple de foncteur familial en ce sens est bien sûr la monade M du théorème 4.1.1, par définition (voir *(3)*). Un exemple similaire est donné par la monade sur $\mathrm{Gph} \simeq \widehat{\Gamma}$ (voir la définition 2.3.3) envoyant chaque graphe sur le graphe de ses chemins (la monade « catégorie libre »). Le rôle de la famille tree y est joué par la famille des graphes *linéaires*, ou *filiformes* (les suites d'arêtes composables).

Grâce au lemme 4.3.10, la démonstration de la proposition 4.2.6 s'adapte (en remplaçant quelques occurrences de 1 par d) et on a :

Corollaire 4.3.11. Un foncteur entre catégories de préfaisceaux est familial ssi c'est un adjoint à droite local finitaire.

Par le lemme 4.2.7, on a donc :

Corollaire 4.3.12. Les foncteurs familiaux entre catégories de préfaisceaux forment une sous 2-catégorie de CAT.

4.4 Monades familiales libres et monades morphologiques

Revenons à présent à la question de départ. Nous savons que notre exemple, la monade M des multicatégories, est un exemple d'enfofoncteur familial sur une catégorie de préfaisceaux. Ses arités sont données par les arbres, c'est-à-dire par la famille tree. Cependant, notre but n'est pas encore atteint, dans le sens où notre théorie des foncteurs familiaux ne permet pas encore ni de démontrer facilement que M est une monade, ni, plus subtil, de comprendre en quoi elle est engendrée non pas par les arbres, mais par la seule famille correspondant aux dessins de la partie 4.1. Plus précisément :

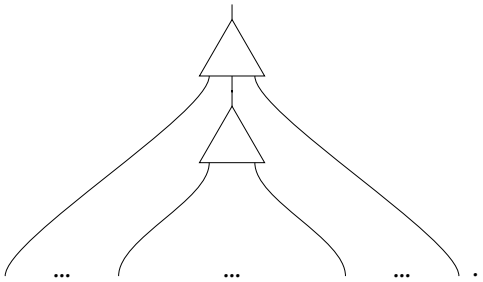
Définition 4.4.11.
On définit la famille $\mathrm{gentree}_n \subseteq \mathrm{ob}\left((n+1)\cdot\star/\widehat{\mathbb{M}}\right)$, indexée par les entiers comme étant constituée
- du morphisme $\star+\star \xrightarrow{[\mathrm{id},\mathrm{id}]} \star$ dans $\mathrm{gentree}_1$,
- pour tous $n, m \in \mathbb{N}$ et $i_0 \in n$, du pushout

muni du morphisme
$$\begin{tikzcd} \star \arrow[r, "s^n_{i_0}"] \arrow[d, "t^m"'] & {[n]} \arrow[d, "l_{n,i_0,m}"] \\ {[m]} \arrow[r, "r_{n,i_0,m}"'] & {[n]\circ_{i_0}[m]}, \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

$$\begin{tikzcd} (n+m)\cdot\star \arrow[d, "{\left[\left[l_{n,i_0,m}\circ s^n_i\right]_{i\in(i_0-1)}, \left[r_{n,i_0,m}\circ s^m_j\right]_{j\in m}, \left[l_{n,i_0,m}\circ s^n_{i_0+i}\right]_{i\in(n-i_0)}, l_{n,i_0,m}\circ t^n\right]}"] \\ {[n]\circ_{i_0}[m]} \end{tikzcd}$$
$$\cong ((i_0-1)+m+(n-i_0))\cdot\star$$

dans $\mathrm{gentree}_{n+m-1}$.

En images, le second point inclut les préfaisceaux de la forme suivante, munis de leur *bord*, c'est-à-dire de l'inclusion de leurs sommets connectés à une seule multi-arête :



On aimerait voir M comme engendrée par la seule famille gentree. Pour apprivoiser ces deux problèmes, commençons par rendre plus explicite les génériques stricts associés à un composé de foncteurs familiaux $\widehat{\mathscr{C}} \xrightarrow{F} \widehat{\mathscr{D}} \xrightarrow{G} \widehat{\mathscr{E}}$. On rappelle que les génériques stricts pour GF sont de la forme $E \xrightarrow{s} G(D) \xrightarrow{G(t)} GF(C)$, avec s générique pour G et t générique pour F. On peut comprendre l'arité C en fonction des opérations atomiques, c'est-à-dire de $F(1)$ et $G(1)$. On s'intéresse plus spécialement au cas des opérations atomiques pour GF, c'est-à-dire au cas où E est représentable. On a vu pour les coproduits puis pour les coégalisateurs, donc pour les colimites arbitraires, que les génériques stricts sont stables par colimites dans $\widehat{\mathscr{D}}/F$. On peut en fait montrer que tous les génériques stricts sont de cette forme. En effet, pour tout générique strict $t : D \to F(C)$, on peut associer à chaque élément $x \in D(d)$ une factorisation $d \xrightarrow{t_x} F(E_t(x)) \xrightarrow{F(\varphi_x)} F(C)$ de $t \circ \mathbf{y}_x$ avec t_x strictement générique. Par définition de la généricité stricte, cette assignation s'étend en un foncteur $\underline{t} : (\mathbf{y}/D) \to (\widehat{\mathscr{D}}/F)$ (qui se factorise par $\mathbf{y}/F$) et on a :

Lemme 4.4.13. Le générique strict t est colimite de $\underline{t}$ dans $\widehat{\mathscr{D}}/F$.

Démonstration. On sait que le foncteur d'oubli $(\widehat{\mathscr{D}}/F) \to \widehat{\mathscr{D}} \times \widehat{\mathscr{C}}$ crée les colimites et on sait aussi que la colimite de $\underline{t}$ dans $\widehat{\mathscr{D}}/F$ est à nouveau générique. En choisissant une colimite, disons C', de la projection sur $\widehat{\mathscr{C}}$, on obtient donc un générique strict $t' : D \to F(C')$, qui par sa propriété universelle de colimite admet un morphisme vers t. Les deux génériques stricts t et t' sont donc connectés dans D/F : ils sont isomorphes. □

Par conséquent, on a

$$GF(1)(e) \cong \sum_{x \in G(1)(e)} \widehat{\mathscr{D}}(E_G(x), F(1))$$

et $E_{GF}(x, (\varphi : E_G(x) \to F(1)))$ est la colimite du diagramme

$$(\mathbf{y}/E_G(x)) \xrightarrow{\mathbf{y}/\varphi} (\mathbf{y}/F(1)) \xrightarrow{E_F} \widehat{\mathscr{C}}.$$

Ceci nous fournit un moyen de calculer les génériques stricts d'un composé de foncteurs familiaux. Appliquons ce résultat en itérant l'endofoncteur gt défini par la famille gentree :

$$\mathrm{gt}(X)(\star) = X(\star) \qquad \text{et} \qquad \mathrm{gt}(X)[n] = \sum_{t \in \mathrm{gentree}_n} \widehat{\mathscr{C}}(t, X).$$

On voit apparaître des arbres de profondeur arbitraire.

Exemple 4.4.3. Considérons $t = ([2] \circ_2 [2]) \in \mathrm{gt}(1)[3] = \mathrm{gentree}_3$. Comme $\mathrm{gt}(X)(\star) = X(\star)$, un diagramme $\varphi : t \to \mathrm{gt}(1)$ est déterminé par une paire $(t_1, t_2) \in \mathrm{gentree}_2^2$. Prenons par exemple $t_1 = ([1] \circ_1 [2])$ et $t_2 = ([2] \circ_1 [1])$. Un calcul facile donne que $\mathbf{y}/t$ est l'ordre partiel engendré par

$$\begin{tikzcd}
 & (\star, a) \arrow[d, "t"] & \\
 & ([2], x_1) & \\
(\star, b) \arrow[ur, "s_1"] & (\star, c) \arrow[u, "s_2"'] \arrow[d, "t"] & \\
 & ([2], x_2) & \\
(\star, d) \arrow[ur, "s_1"] & & (\star, e). \arrow[ul, "s_2"']
\end{tikzcd}$$

On prend donc la colimite dans $\widehat{\mathscr{C}}$ du diagramme

$$\begin{tikzcd}
 & \star \arrow[d, "t"] & \\
 & t_1 & \\
\star \arrow[ur, "s_1"] & \star \arrow[u, "s_2"'] \arrow[d, "t"] & \\
 & t_2 & \\
\star \arrow[ur, "s_1"] & & \star, \arrow[ul, "s_2"']
\end{tikzcd}$$

qui est évidemment le pushout de $t_1 \xleftarrow{s_2} \star \xrightarrow{t} t_2$, c'est-à-dire l'arbre t' représenté en figure 4.2.

On pourrait donc être tenté de considérer la monade Gt librement engendrée par gt. Cependant, dans $\mathrm{gt}^2(1)$, l'élément (t, φ) diffère de celui obtenu en partant par exemple de $u = ([3] \circ_3 [1])$ et en prenant comme diagramme celui déterminé par $u_1 = t'$ et $u_2 = \star$. Autrement dit, on aura bien dans Gt toutes les opérations de multicatégorie, mais elles ne satisferont pas les conditions de cohérence souhaitées.

Néanmoins, on peut formaliser brutalement ces conditions de cohérence, en fait pour tout endofoncteur familial T sur $\widehat{\mathbb{M}}$, en demandant, en gros, que chaque opération soit déterminée par sa forme, son arité. Tentons de donner une définition précise. Naïvement, on pourrait décréter qu'un endofoncteur familial T sur $\widehat{\mathbb{M}}$ satisfait aux conditions de cohérence graphique ssi $E_T : (\mathbf{y}/T(1)) \to \widehat{\mathbb{M}}$ est *essentiellement injectif* sur les objets,

c'est-à-dire que si $E_T(c, x) \cong E_T(c, y)$, pour tous $x, y \in T(1)(c)$, alors $x = y$. Cependant, il semble préférable de prendre en compte dans la notion de forme les « résidus » d'une opération $x \in T(1)(c)$, c'est-à-dire ses images par les actions des morphismes de $\mathbb{M}$. On considère donc que la *forme* de $x \in T(1)(c)$ est le foncteur composé

$$(\mathbb{M}/c) \cong (\mathbf{y}/c) \xrightarrow{\mathbf{y}/x} (\mathbf{y}/T(1)) \xrightarrow{E_T} \widehat{\mathbb{M}}$$

et on pose :

Définition 4.4.12. Un endofoncteur familial T sur $\widehat{\mathbb{M}}$ est *morphologique* ssi pour tous $c \in \mathbb{M}$ et $x, y \in T(1)(c)$, si x et y ont des formes isomorphes alors $x = y$. Formellement, si $E_T \circ (\mathbf{y}/x) \cong E_T \circ (\mathbf{y}/y)$ alors $x = y$.

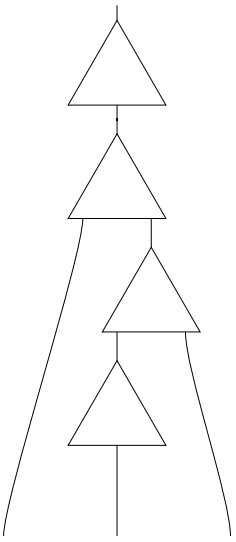


Figure 4.2 – Termes de réduction engendrés par S

On observe alors qu'il existe un endofoncteur morphologique finitaire universel, comprenant précisément une opération pour chaque forme.

Définition 4.4.13. Soit U l'endofoncteur familial défini en prenant pour chaque $U(1)(c)$ un choix global de représentants de classes d'isomorphisme des foncteurs $\mathbf{y}/c \to \widehat{\mathbb{M}}_f$, avec $E_U(x) = x(\mathrm{id}_c)$.

Pour chaque morphisme $f : d \to c$ dans $\mathbb{M}$, on pose de plus

$$U(f)[x] = \downarrow\left(\mathbf{y}/d \xrightarrow{\mathbf{y}/f} \mathbf{y}/c \xrightarrow{x} \widehat{\mathbb{M}}_f\right),$$

où $\downarrow x$ désigne le représentant choisi de la classe de x.

L'universalité de U s'exprime par le fait que c'est un objet faiblement terminal :

Proposition 4.4.14. Pour tout endofoncteur familial finitaire F, les fonctions

$$\begin{aligned} F(X)(c) \cong \sum_{x \in F(1)(c)} \widehat{\mathbb{M}}(E_F(x), X) &\to U(X)(c) \\ (x, \varphi) &\mapsto (\downarrow[\mathbf{y}/c \xrightarrow{\mathbf{y}/x} \mathbf{y}/F(1) \xrightarrow{E_F} \widehat{\mathbb{M}}_f], \varphi) \end{aligned}$$

définissent une transformation naturelle familiale $! : F \to U$.

Démonstration. Facile. □

A ce stade, on est tout près de savoir conclure. En effet, si on sait munir U d'une structure de monade, alors en raisonnant dans le treillis des sous-objets du préfaisceau $U(1)$, on peut pour tout $\Sigma \subseteq U(1)$ considérer la plus petite sous-monade T_Σ de U contenant Σ. Pour tous sous-objets $X, Y \subseteq U(1)$, on peut définir leur *composition* $Y \otimes X$ comme l'image de la transformation naturelle

$$F_Y \circ F_X \hookrightarrow U \circ U \xrightarrow{\mu_U} U. \tag{4}$$

On construit alors T_Σ comme

$$T_\Sigma = \bigcup_{n \in \mathbb{N}} (I \cup \Sigma)^{\otimes n},$$

où par convention $\Sigma^{\otimes 0}$ désigne le sous-préfaisceau I de $U(1)$ correspondant au foncteur identité, c'est-à-dire

$$I(c) = \left\{\left[\mathbf{y}/c \xrightarrow{\mathrm{dom}_c} \mathbb{M} \xrightarrow{\mathbf{y}} \widehat{M}_f\right]\right\}.$$

Ceci sert bien sûr à s'assurer que T_Σ contient l'identité, ce qui fournit l'unité de la monade. Pour voir que I induit le foncteur identité, on calcule directement :

$$\begin{aligned} F_I(X)(c) &\cong \sum_{x \in I(c)} \widehat{\mathbb{M}}(E_I(x), X) \\ &\cong \widehat{\mathbb{M}}(\mathbf{y} \circ \mathrm{dom}_c(\mathrm{id}_c), X) \\ &\cong \widehat{\mathbb{M}}(\mathbf{y}_c, X) \\ &\cong X(c). \end{aligned}$$

Le fait qu'on prenne l'image du morphisme *(4)* signifie qu'on ne garde que la forme des opérations composites.

Exemple 4.4.4. On peut voir la famille gentree de la définition 4.4.11 comme un sous-préfaisceau de U : chaque élément de $\mathrm{gentree}_n$ est un préfaisceau X muni d'une transformation naturelle $(n+1)\cdot\star \to X$. Or, $\mathbb{M}/[n]$ est la catégorie engendrée par le graphe

$$\begin{tikzcd}
 & (\star, t) \arrow[d, "t"] & \\
 & ([n], \mathrm{id}_{[n]}) & \\
(\star, s_1) \arrow[ur, "s_1"] & \cdots & (\star, s_n). \arrow[ul, "s_n"']
\end{tikzcd}$$

Une forme $\mathbf{y}/[n] \to \widehat{\mathbb{M}}_f$ valant $\star$ sur chaque objet de la forme $(\star, f)$ revient donc à un diagramme de la forme

$$\begin{tikzcd}
 & \star \arrow[d, "\tau"] & \\
 & X & \\
\star \arrow[ur, "\sigma_1"] & \cdots & \star, \arrow[ul, "\sigma_n"']
\end{tikzcd}$$

c'est-à-dire à un morphisme $(n+1)\cdot\star \to X$. Comme le composé

$$\mathbf{y}/\star \xrightarrow{f} \mathbf{y}/[n] \xrightarrow{(X, [\tau, [\sigma_i]_i])} \widehat{\mathbb{M}}_f$$

est toujours égal à

$$\mathbf{y}/\star \cong 1 \xrightarrow{\ulcorner\star\urcorner} \widehat{\mathbb{M}}_f$$

on fixe $\mathrm{gentree}(\star) = \{\ulcorner\star\urcorner\}$ et on obtient bien un sous-préfaisceau de U. Pour concrétiser, dans l'exemple *4.4.3*, on a considéré deux opérations de même forme dans Gt, $(t, [t_1, t_2])$ et $(u, [u_1, u_2])$. Dans T_{gt}, elles sont identifiées.

Le seul point en suspens reste la structure de monade de U. A ce sujet s'ouvre une voie tellement tentante que notre travail avec Richard Garner s'est focalisé dessus. Si U était proprement terminal, pas seulement faiblement, on pourrait conclure directement. En effet, on a montré au lemme 4.2.7 que les foncteurs familiaux (finitaires) sont stables par composition ; les endofoncteurs familiaux sur $\widehat{\mathbb{M}}$ forment donc une sous-catégorie monoïdale de $\mathrm{CAT}(\widehat{\mathbb{M}}, \widehat{\mathbb{M}})$. Or, tout objet terminal dans une catégorie monoïdale est canoniquement un monoïde. La multiplication $1 \otimes 1 \to 1$ est l'unique morphisme du bon type, l'unité est l'identité et les lois de monoïde sont automatiquement satisfaites grâce à l'unicité dans la propriété universelle de 1.

Cependant, en général, U n'est pas proprement terminal.

Exemple 4.4.5. Considérons le cas $\mathbb{M} = 1$, celui des ensembles. La source du problème est que U admet des automorphismes non-triviaux. Dans ce cas, en effet, les éléments de $\mathsf{U}(\star)$

(où $\star$ est l'unique objet de la catégorie terminale 1) sont les foncteurs $1/\star \cong 1 \to \mathrm{Set}_f \simeq$ set, c'est-à-dire les ensembles finis. On définit alors pour tout $n \in$ set l'automorphisme τ_n comme étant le renversement $i \mapsto n-i+1$, ce qui nous fournit un automorphisme non-trivial de U, qui ne peut exister pour un objet terminal :

$$\begin{aligned} U(X)(c) \cong \textstyle\sum_{x \in U(1)(c)} \mathrm{Set}(x, X) &\to U(X)(c) \\ (x : 1 \to \mathrm{set}, \varphi : x \to X) &\mapsto (x, \varphi \circ \tau_x). \end{aligned}$$

4.5 Foncteurs analytiques

Dans cette partie, nous montrons comment étendre la classe de foncteurs considérés pour prendre en compte les automorphismes et ainsi obtenir une notion de foncteur morphologique qui admette un objet terminal. Cependant, la question n'est pas encore totalement résolue puisqu'on montrera aussi que ce faisant on perd la stabilité par composition. En partie suivante, on exhibera dans certains cas particuliers une classe moins générale de foncteurs, englobant quand-même les foncteurs familiaux, stable par composition et possédant un objet terminal.

L'idée de base pour étendre les foncteurs familiaux est, dans le cadre des ensembles, de passer aux foncteurs analytiques au sens de Joyal [40]. Ces derniers, par définition, sont les foncteurs $\mathrm{Set} \to \mathrm{Set}$ obtenus par extension de Kan à gauche le long du plongement $\mathbb{B} \hookrightarrow \mathrm{Set}$, où $\mathbb{B}$ désigne la catégorie des ordinaux finis munis des bijections entre eux. Autrement dit, $\mathbb{B} = \sum_n \mathfrak{S}_n$, où $\mathfrak{S}_n$ est le groupe d'automorphismes de n vu comme une catégorie. Par la formule standard pour les extensions à gauche, on a

Proposition 4.5.15. F est analytique ssi il existe $B \in \mathrm{CAT}(\mathbb{B}, \mathrm{Set})$ tel que

$$F(X) \cong \int^n B(n) \times \mathrm{Set}(n, X) \cong \sum_n \sum_{o \in \lfloor \pi_0(B(n)) \rfloor} \mathrm{Set}(n, X)/\mathfrak{S}_o,$$

où

- $\lfloor \pi_0(B(n)) \rfloor$ désigne un choix de représentants pour l'ensemble $\pi_0(B(n))$ des *orbites* de $B(n)$, c'est-à-dire le quotient de $B(n)$ par la relation identifiant chaque $x \in B(n)$ avec $x \cdot \sigma$ et
- $\mathfrak{S}_o$ désigne le sous-groupe de $\mathfrak{S}_n$ contenant les isomorphismes σ *fixant* o, c'est-à-dire tels que $\sigma \cdot o = o$, qui agit sur $\mathrm{Set}(n, X)$ par précomposition.

Démonstration. Le premier isomorphisme découle directement de la formule standard. Pour le second, par la caractérisation standard des cofins comme quotients, le premier isomorphisme entraîne que $F(X)$ est le quotient de $\sum_n B(n) \times \mathrm{Set}(n, X)$ par la relation identifiant $(n, x, \varphi : n \to X)$ et $(m, y, \psi : m \to X)$ dès que $n = m$ et qu'il existe un iso $\sigma : n \to n$ tel que $y = \sigma \cdot x$ et $\psi \circ \sigma = \varphi$. Visuellement :

$$\begin{tikzcd}
 & 1 \arrow[dl, "x"'] \arrow[dr, "y"] & \\
B(n) \arrow[rr, "B(\sigma)"] & & B(n) \\
n \arrow[rr, "\sigma"] \arrow[dr, "\varphi"'] & & n \arrow[dl, "\psi"] \\
 & X. &
\end{tikzcd}$$

Comme $\mathbb{B}$ est un groupoïde, cette relation est en effet directement une relation d'équivalence. Choisissons donc un représentant o par classe de connexité O dans $B(n)$. Pour tout $o' \in O$, il existe σ tel que $o' = \sigma \cdot o$, donc tout triplet (n, o', φ') est équivalent à un triplet de la forme (n, o, φ), ici $(n, o, \varphi' \circ \sigma)$. Pour caractériser le quotient, il reste donc à déterminer quand deux tels triplets sont équivalents. Par définition, $(n, o, \varphi) \sim (n, o, \varphi')$ lorsqu'il existe $\sigma \in \mathfrak{S}_n$ tel que $\sigma \cdot o = o$ et $\varphi' \circ \sigma = \varphi$, autrement dit lorsqu'il existe $\sigma \in \mathfrak{S}_o$ tel que $\varphi' \circ \sigma = \varphi$, d'où le résultat. □

Remarque 4.5.3. Cette démonstration, bien qu'élémentaire, est quelque peu subtile et j'encourage le lecteur à y passer le temps nécessaire. En particulier, les sous-groupes $\mathfrak{S}_o$ sont absolument cruciaux pour la suite.

Corollaire 4.5.16. Comme cas particulier, on a $F(1) = \sum_n \lfloor \pi_0(B(n)) \rfloor \cong \sum_n \pi_0(B(n))$.

Commençons par constater :

Proposition 4.5.17. Tout endofoncteur familial sur les ensembles est analytique.

Démonstration. Soit F familial, avec $F(X) \cong \sum_{x \in F(1)} \mathrm{Set}(E_F(x), X)$ pour tout X. On considère le foncteur $B_F : \mathbb{B} \to \mathrm{Set}$ défini par le coproduit

$$B_F = \sum_{x \in F(1)} \mathbf{y}_{E_F(x)}.$$

Autrement dit, pour chaque générique strict x, on crée une opération o_x dans B_F, d'arité $n = E_F(x)$. Un point important ici est que l'action de $\mathfrak{S}_n$ sur cette opération est libre : l'orbite de o_x est par définition en bijection avec $\mathfrak{S}_n$ et $\mathfrak{S}_{o_x}$ est le sous-groupe trivial $\{\mathrm{id}_n\}$. On peut donc sélectionner id_n comme représentant canonique de l'orbite et par la proposition 4.5.15 le foncteur associé à B_F envoie chaque X sur

$$\sum_n \sum_{\{x \in F(1) \mid E_F(x) = n\}} \mathrm{Set}(n, X) \cong \sum_{x \in F(1)} \mathrm{Set}(E_F(x), X) \cong F(X),$$

comme souhaité. □

Ce résultat met clairement en avant le fait que la richesse des foncteurs analytiques vient du quotient par $\mathfrak{S}_0$.

On peut construire pour la notion d'analyticité une correspondance analogue à celle entre la familialité et l'existence de génériques stricts, à laquelle il va précisément manquer cet aspect. Nous y reviendrons ensuite. Considérons pour chaque n l'ensemble $\mathrm{Gen}(F)(n)$ des éléments $1 \to F(n)$ de la forme $(o \in \pi_0(B(n)), \mathrm{id}_n)$. Ils satisfont une propriété à peine plus faible que dans le cas familial :

Définition 4.5.14. Pour un foncteur $F : \mathscr{C} \to \mathscr{D}$, $C \in \mathscr{C}$ et $D \in \mathscr{D}$, un morphisme $s : D \to F(C)$ est *générique* ssi pour tout carré comme ci-dessous il existe un morphisme diagonal h tel que $F(h) \circ s = t$ et $g \circ h = f$:

$$\begin{tikzcd} D \arrow[r, "t"] \arrow[d, "s"'] & F(X) \arrow[d, "F(g)"] \\ F(C) \arrow[r, "F(f)"'] \arrow[ur, dashed, "F(h)"] & F(Y). \end{tikzcd}$$

Un morphisme $f : D \to F(X)$ est *couvert* par le générique s ssi il existe $\varphi : C \to X$ tel que f se décompose en $D \xrightarrow{s} F(C) \xrightarrow{F(\varphi)} F(X)$.

On dit de plus que F admet des factorisations génériques relativement à D si pour tout $X \in \mathscr{C}$, tout morphisme $D \to F(X)$ est couvert par un générique. F admet des factorisations génériques (tout court) si F en admet pour tout objet D. Enfin, un générique $D \to F(X)$ est *finitaire* ssi X est finiment présentable.

Bien sûr, cette définition est mot pour mot la définition 4.2.5 moins l'unicité.

Pour montrer la correspondance, on a besoin du lemme crucial suivant :

Lemme 4.5.18. Pour tout $F : \mathscr{C} \to \mathscr{D}$ et tout diagramme commutatif

$$\begin{tikzcd} 1 \arrow[r, "t"] \arrow[d, "s"'] & F(Y) \arrow[d, "F(\psi)"] \\ F(X) \arrow[r, "F(\varphi)"'] \arrow[ur, "F(\delta)"] & F(Z), \end{tikzcd}$$

(tel que $F(\delta) \circ s = t$ et $\psi \circ \delta = \varphi$) avec s et t génériques, δ est un isomorphisme.

Démonstration. Par généricité de s on trouve un $\gamma : Y \to X$ faisant commuter le diagramme de gauche ci-dessous

$$\begin{tikzcd} 1 \arrow[r, "s"] \arrow[d, "t"'] & F(X) \arrow[d, "F(\delta)"] \\ F(Y) \arrow[r, "F(\mathrm{id}_Y)"'] \arrow[ur, "F(\gamma)"] & F(Y) \end{tikzcd} \qquad \begin{tikzcd} 1 \arrow[r, "t"] \arrow[d, "s"'] & F(Y) \arrow[d, "F(\gamma)"] \\ F(X) \arrow[r, "F(\mathrm{id}_X)"'] \arrow[ur, "F(\theta)"] & F(X), \end{tikzcd}$$

avec en particulier $\delta \circ \gamma = \mathrm{id}_Y$: δ est donc un épimorphisme scindé et γ un monomorphisme scindé. Mais par le même raisonnement on montre que γ est aussi un épimorphisme scindé (voir le diagramme de droite ci-dessus) : c'est donc un isomorphisme, ainsi que δ. $\square$

Proposition 4.5.19. Un foncteur $F : \mathrm{Set} \to \mathrm{Set}$ est analytique ssi il admet des factorisations génériques finitaires relativement à 1.

Démonstration. Soit F analytique, engendré par $B : \mathbb{B} \to \mathrm{Set}$. Alors on peut factoriser tout élément $[n, x, \varphi] : 1 \to \int^n B(n) \times \mathrm{Set}(n, X)$ comme

$$1 \xrightarrow{[n,x,\mathrm{id}]} \int^{n'} B(n') \times \mathrm{Set}(n', n) \xrightarrow{F(\varphi)} \int^{n'} B(n') \times \mathrm{Set}(n', X)$$

et montrer que $[n, x, \mathrm{id}]$ est générique. En effet, considérons un carré commutatif arbitraire

$$\begin{tikzcd} 1 \arrow[r, "{[m,y,\psi]}"] \arrow[d, "{[n,x,\mathrm{id}]}"'] & F(X) \arrow[d, "F(\gamma)"] \\ F(n) \arrow[r, "F(\delta)"'] & F(Y). \end{tikzcd}$$

Par définition de F, la commutativité du carré signifie qu'on a

$$[m, y, \gamma \circ \psi] = [n, x, \delta],$$

c'est-à-dire, comme $\mathbb{B}$ est un groupoïde, qu'il existe $\sigma \in \mathbb{B}(m, n)$ tel que $y = \sigma \cdot x$ et $\gamma \circ \psi \circ \sigma = \delta$. En particulier, $m = n$ et on montre alors que $\psi \circ \sigma$ fait commuter

$$\begin{tikzcd} 1 \arrow[r, "{[m,y,\psi]}"] \arrow[d, "{[n,x,\mathrm{id}]}"'] & F(X) \arrow[d, "F(\gamma)"] \\ F(n) \arrow[r, "F(\delta)"'] \arrow[ur, "F(\psi\circ\sigma)"] & F(Y). \end{tikzcd}$$

En effet, pour le triangle du bas, on a $\gamma \circ \psi \circ \sigma = \delta$ par hypothèse et, pour le triangle du haut, on a

$$(\psi \circ \sigma) \cdot [n, x, \mathrm{id}] = [n, x, \psi \circ \sigma] = [n, \sigma \cdot x, \psi] = [n, y, \psi].$$

Réciproquement, si F admet des factorisations génériques relativement à 1, on définit $B(n)$ comme étant l'ensemble des génériques $1 \to F(n)$, munis de l'action de F. En

définissant F_B comme étant l'extension à gauche de B le long de $\mathbb{B} \hookrightarrow \mathrm{Set}$, on a alors une transformation naturelle canonique $\lambda : F_B \to F$ envoyant, en X, $[n, x : 1 \to F(n), \varphi]$ sur $F(\varphi)(x)$. Mais si $[n, x, \varphi]$ et $[m, y, \psi]$ ont la même image par λ_X, par généricité de x on trouve un morphisme δ faisant commuter le diagramme

$$\begin{tikzcd} 1 \arrow[r, "y"] \arrow[d, "x"'] & F(m) \arrow[d, "F(\psi)"] \\ F(n) \arrow[r, "F(\varphi)"'] \arrow[ur, "F(\delta)"] & F(X). \end{tikzcd}$$

Par le lemme, δ est un isomorphisme, donc on a $\delta \cdot x = y$ et $\psi \circ \delta = \varphi$, soit $[n, x, \varphi] = [m, y, \psi]$. □

La démonstration du lemme 4.2.5 s'applique à l'identique et on a :

Définition 4.5.15. On note $\mathrm{Ana}(F)$ la classe des objets relativement auxquels un foncteur $F : \mathscr{C} \to \mathscr{D}$ admet des factorisations génériques.

Lemme 4.5.20. Pour tout foncteur F, $\mathrm{Ana}(F)$ est stable par coproduits.

Corollaire 4.5.21. Un endofoncteur sur les ensembles est analytique ssi il admet des factorisations génériques finitaires.

On le voit venir, le lemme ci-dessus ne parle pas de coégalisateurs et en général c'est bien parce que $\mathrm{Ana}(F)$ n'est pas stable par coégalisateurs. On donne un contre-exemple dans l'article. Nous reviendrons à cette question dans la partie suivante sur la stabilité par composition des foncteurs analytiques. Ici, nous nous concentrons sur l'existence d'un objet terminal pour les foncteurs analytiques et l'extension de cette propriété des ensembles aux catégories de préfaisceaux.

Définition 4.5.16. Une transformation naturelle entre deux foncteurs analytiques est *analytique* ssi elle préserve les génériques finitaires.

Lemme 4.5.22. L'assignation

$$\begin{aligned} \mathrm{ob}(\mathrm{CAT}(\mathbb{B}, \mathrm{Set})) &\to \mathrm{ob}(\mathrm{Ana}(\mathrm{Set}, \mathrm{Set})) \\ B &\mapsto X \mapsto \textstyle\int^n B(n) \times X^n \end{aligned}$$

s'étend en un foncteur, qui est en fait une équivalence de catégories.

Démonstration. Toute transformation naturelle $\alpha : B \to C$ dans $\mathrm{CAT}(\mathbb{B}, \mathrm{Set})$ donne lieu à une famille de fonctions $\sum_n B(n) \times X^n \to \sum_n C(n) \times X^n$, en envoyant (n, x, φ) sur $(n,$

$\alpha_n(x), \varphi)$. Ces fonctions passent au quotient, par naturalité de α, donc on a bien un foncteur comme souhaité.

Pour prouver qu'il est fidèle, considérons deux telles transformations naturelles α et β et supposons que les transformations $F_B \to F_C$ induites, disons a et b sont égales. On a alors pour tout n et $x \in B(n)$ que $[n, \alpha_n(x), \mathrm{id}] \sim [n, \beta_n(x), \mathrm{id}]$, donc qu'il existe $\sigma \in \mathfrak{S}_n$ tel que $\sigma \cdot \alpha_n(x) = \beta_n(x)$ et $\mathrm{id} \circ \sigma = \mathrm{id}$. Ainsi, $\sigma = \mathrm{id}$ et $\alpha_n(x) = \beta_n(x)$. On a prouvé que $\alpha = \beta$ et donc notre foncteur est bien fidèle.

Pour montrer la plénitude de notre foncteur, montrons que pour tout générique finitaire $[n, x, \varphi] \in F_B(X)$, il existe $x' \in B(n)$ tel que $[n, x', \mathrm{id}] = [n, x, \varphi]$. En effet, étant donnés x et φ, par généricité de $[n, x, \mathrm{id}]$ on obtient ψ tel que

$$\begin{tikzcd}
1 \arrow[r, "{[n,x,\varphi]}"] \arrow[d, "{[n,x,\mathrm{id}]}"'] & F_B(X) \arrow[d, "{F_B(\mathrm{id}_X)}"] \\
F_B(X) \arrow[r, "{F_B(\varphi)}"'] \arrow[ur, "{F_B(\psi)}"] & F_B(X)
\end{tikzcd}$$

commute. Par la démonstration de la proposition 4.5.19, $[n, x, \mathrm{id}]$ est générique donc (par le lemme 4.5.18) ψ est un isomorphisme et donc φ aussi. On a donc $[n, x, \varphi] = [n, x, \psi] = [n, \psi \cdot x, \mathrm{id}]$ comme souhaité.

Considérons à présent une transformation naturelle analytique $a : F_B \to F_C$. Pour tout $x \in B(n)$, $a[n, x, \mathrm{id}]$ est générique donc est de la forme $[n, x', \mathrm{id}]$ pour un certain $x' \in C(n)$. Choisissons pour chaque classe de connexité O dans $B(n)$ un représentant canonique o_O et pour chaque $[n, o_O, \mathrm{id}]$ un représentant $[n, o'_O, \mathrm{id}]$ dans $a[n, o_O, \mathrm{id}]$. Définissons enfin α_n comme envoyant chaque o_O sur o'_O et chaque autre $x \in B(n)$ sur $C(\sigma)\left(o'_{O_x}\right)$, où O_x est la classe de connexité de x dans B et $x = \sigma \cdot O_{O_x}$. On a ainsi pour chaque o_O et σ que $\alpha_n(\sigma \cdot o_O) = \sigma \cdot o'_O$, donc α est naturelle et s'envoie sur a par construction. On a donc un foncteur plein et fidèle, qui est essentiellement surjectif par construction : c'est bien une équivalence. □

Proposition 4.5.23. La sous-catégorie $\mathrm{Ana}(\mathrm{Set}, \mathrm{Set})$ des foncteurs analytiques admet un objet terminal, donné par le foncteur $\mathscr{C}$ associant à chaque ensemble le monoïde commutatif libre associé.

Démonstration. La catégorie $\mathrm{CAT}(\mathbb{B}, \mathrm{Set})$ admet évidemment un objet terminal qui est préservé par l'équivalence avec $\mathrm{Ana}(\mathrm{Set}, \mathrm{Set})$. Cet objet terminal est donc donné par

$$F_1(X) = \int^n X^n,$$

qui est bien le foncteur attendu. □

Une autre façon de voir le dernier point est que les éléments $o \in 1_{\mathrm{CAT}(\mathbb{B},\,\mathrm{Set})}(n)$ du préfaisceau covariant terminal sur $\mathbb{B}$ sont fixés par tous les isomorphismes et vérifient donc $\mathfrak{S}_o = \mathfrak{S}_n$. Ainsi, $F_1(X) = \sum_n X^n/\mathfrak{S}_n = \mathscr{C}(X)$.

Comment, à présent, étendre ce résultat aux catégories de préfaisceaux ? Comme pour les foncteurs familiaux, on pose :

Définition 4.5.17. Un foncteur $F : \mathscr{C} \to \widehat{\mathscr{D}}$ est *analytique* ssi il admet des factorisations génériques finitaires relativement aux représentables. Une transformation naturelle entre deux foncteurs analytiques est *analytique* ssi elle préserve ces génériques. On désigne par $\mathrm{Ana}(\mathscr{C}, \widehat{\mathscr{D}}) \subseteq \mathrm{CAT}(\mathscr{C}, \widehat{\mathscr{D}})$ la catégorie des foncteurs analytiques munis des transformations naturelles analytiques.

Comment montrer que cette sous-catégorie admet un objet terminal ?

L'idée fondamentale, peut-être la plus importante de ce chapitre, est de considérer une notion plus précise d'arité prenant en compte les groupes d'automorphismes : on considère une catégorie de départ légèrement augmentée, notée $\mathscr{O}(\mathscr{C})$, et on montre que les foncteurs analytiques $\mathscr{C} \to \widehat{\mathscr{D}}$ correspondent aux foncteurs familiaux $\mathscr{O}(\mathscr{C}) \to \widehat{\mathscr{D}}$. Là où un foncteur familial finitaire $\mathscr{C} \to \widehat{\mathscr{D}}$ correspond à la donnée d'un préfaisceau S sur $\mathscr{D}$ muni d'un foncteur $\mathbf{y}/S \to \mathscr{C}_f$, on remplace ici ce dernier par un foncteur $\mathbf{y}/S \to \mathscr{O}(\mathscr{C}_f)$.

Exemple 4.5.6. Comparons les génériques pour les monades $\mathscr{L}$ et $\mathscr{C}$ envoyant respectivement chaque ensemble sur le monoïde et le monoïde commutatif libres associés. Les deux foncteurs sont analytiques et possèdent tous deux des génériques de toutes les arités finies. Pour les distinguer, on observe que les génériques $1 \to \mathscr{L}(n)$ de $\mathscr{L}$ ne sont fixés par aucun automorphisme non-trivial de n, alors que ceux de $\mathscr{C}$ sont fixés par tous. Dans notre nouvelle catégorie d'arités $\mathscr{O}(\mathrm{Set})$, on considère ainsi (entre autres) deux objets différents pour chaque n :

- d'une part l'objet $(n, 1)$, constitué de n et du sous-groupe trivial de $\mathfrak{S}_n$, et
- d'autre part l'objet $(n, \mathfrak{S}_n)$, constitué de n et du groupe total $\mathfrak{S}_n$.

Définition 4.5.18. Etant donnée une catégorie $\mathscr{A}$, soit $\mathscr{O}(\mathscr{A})$ la catégorie

- dont les objets sont les paires (A, G) d'un objet $A \in \mathrm{ob}(\mathscr{A})$ et d'un sous-groupe G du groupe $\mathfrak{S}_A$ d'automorphismes de A dans $\mathscr{A}$
- et dont les morphismes $(A, G) \to (B, H)$ sont les classes d'équivalences de morphismes $f : A \to B$
  - tels que pour tout $h \in H$ il existe $g \in G$ tel que $h \circ f = f \circ g$,
  - pour la relation engendrée par $f \sim f \circ g$ pour tout $g \in G$.

Pour comprendre cette définition, rappelons que par la formule donnée en début de partie, on identifie (n, o, φ) et $(n, o, \psi) \in F_B(X)$ dès qu'il existe $\sigma \in \mathfrak{S}_o$ tel que $\varphi \circ \sigma = \psi$. On obtient ainsi une nouvelle formulation équivalente :

Proposition 4.5.24. Un foncteur $F : \mathrm{Set} \to \mathrm{Set}$ est analytique ssi il existe $B \in \mathrm{CAT}(\mathbb{B}, \mathrm{Set})$ tel que

$$F(X) \cong \sum_{n} \sum_{o \in \lfloor \pi_0(B(n)) \rfloor} \mathscr{O}(\mathrm{Set})((n, \mathfrak{S}_o), (X, 1)).$$

Cette formule est un indice que les foncteurs analytiques sur $\mathscr{A}$ peuvent se voir comme familiaux sur $\mathscr{O}(\mathscr{A})$: on se ramène à une somme de représentables.

Explorons cette piste, qui va nous mener à une équivalence de catégories. On commence par construire et étudier le plongement canonique $i : \mathscr{A} \hookrightarrow \mathscr{O}(\mathscr{A})$, pour toute catégorie $\mathscr{A}$. Ce dernier associe à chaque objet A la paire $(A, 1)$ de A et du groupe trivial, vu comme un sous-groupe de $\mathfrak{S}_A$. Sur les morphismes, on associe à chaque $f : A \to B$ sa classe d'équivalence pour la relation adéquate, qui se trouve ici être le singleton $\{f\}$ du fait de la trivialité des groupes concernés. On a prouvé :

Proposition 4.5.25. Le plongement $i : \mathscr{A} \hookrightarrow \mathscr{O}(\mathscr{A})$ est plein et fidèle.

On peut en fait voir $\mathscr{O}(\mathscr{A})$ comme une complétion de $\mathscr{A}$ par une certaine sorte de limites :

Définition 4.5.19. Un *invariant de groupe* dans une catégorie $\mathscr{C}$ est la limite d'un foncteur $g : G \to \mathscr{C}$ d'un groupe vu comme une catégorie à un objet. On note $C^{\backslash g}$ la limite correspondante, où C est l'image de l'unique objet de G ; lorsque le foncteur se déduit du contexte, on note $C^{\backslash G}$. On dit que $\mathscr{C}$ admet les invariants de groupes lorsque tous les foncteurs $G \to \mathscr{C}$ admettent une limite.

Proposition 4.5.26. Pour tout $A \in \mathrm{ob}(\mathscr{A})$ et tout sous-groupe $G \leqslant \mathfrak{S}_A$, le morphisme $[\mathrm{id}] : (A, G) \to (A, 1)$, qu'on notera dans la suite q_G, est un invariant de groupe pour le foncteur canonique $G \to \mathscr{A}$.

Démonstration. Le point non-trivial est que tout morphisme $[f] : (B, H) \to (A, 1)$ formant un cone sur G se factorise de manière unique par (A, G). Mais par hypothèse, un tel morphisme vérifie $[f] = [g \circ f]$ pour tout $g \in G$, donc on trouve $h \in H$ tel que $f \circ h = g \circ f$ et donc f donne aussi lieu à un morphisme $(B, H) \to (A, G)$. Enfin, tout $[h] : (B, H) \to (A, G)$ dont la composition avec q_G donne $[f]$ vérifie par définition que $[h] = [f]$, ce qui conclut la démonstration. □

En fait, on a plus : soit Invar la sous-2-catégorie localement pleine de CAT obtenue en se restreignant aux catégories pourvues d'invariants de groupes et aux foncteurs les préservant.

Proposition 4.5.27. La catégorie $\mathscr{O}(\mathscr{A})$ admet tous les invariants de groupe et pour toute catégorie $\mathscr{B}$ admettant les invariants de groupes le foncteur

$$\mathrm{Invar}(\mathscr{O}(\mathscr{A}), \mathscr{B}) \to \mathrm{CAT}(\mathscr{A}, \mathscr{B})$$

de précomposition par i est une équivalence.

Démonstration. Commençons par montrer que $\mathscr{O}(\mathscr{A})$ admet tous les invariants de groupes. Soit (A, G) un objet et $H \leqslant \mathfrak{S}_{(A,G)}$. Considérons le sous-groupe K de $\mathfrak{S}_A$ constitué des automorphismes σ tels que $[\sigma] \in H$. En particulier, on a $G \leqslant K$ puisque pour tout $g \in G$, $[g] = [\mathrm{id}] \in H$. Ceci implique que l'identité sur A induit un morphisme $q_{K,G} : (A, K) \to (A, G)$ qui est l'invariant de groupe recherché. En effet, si $[f] : (B, L) \to (A, G)$ forme un cône sur H, on a que pour tout $k \in K$, $[k] \circ [f] = [f]$, c'est-à-dire qu'il existe $l \in L$ tel que $k \circ f = f \circ l$. Le morphisme $f : B \to A$ induit donc un morphisme $[f] : (B, L) \to (A, K)$. Or ce morphisme est unique puisque tout autre morphisme $[f']$ satisfaisant cette propriété vérifie aussi

$$q_{K,G} \circ [f'] = [f] = q_{K,G} \circ [f]$$

et donc $[f'] = [f]$, comme souhaité.

On montre relativement facilement que le foncteur de précomposition par i est plein et fidèle. Pour prouver sa surjectivité à iso près, on associe à chaque $F : \mathscr{A} \to \mathscr{B}$ le foncteur $F' : \mathscr{O}(\mathscr{A}) \to \mathscr{B}$ envoyant chaque (A, G) sur $F(A)^{\backslash F(G)}$, ce qui s'étend aux morphismes. Plus de détails sont donnés dans l'article. □

Théorème 4.5.28. Pour toute catégorie $\mathscr{C}$, l'équivalence

$$\mathrm{Invar}\left(\mathscr{O}(\mathscr{A}), \widehat{\mathscr{C}}\right) \to \mathrm{CAT}\left(\mathscr{A}, \widehat{\mathscr{C}}\right)$$

se restreint à une équivalence

$$\mathrm{Fam}'\left(\mathscr{O}(\mathscr{A}), \widehat{\mathscr{C}}\right) \to \mathrm{Ana}\left(\mathscr{A}, \widehat{\mathscr{C}}\right),$$

où $\mathrm{Fam}'(\mathscr{O}(\mathscr{A}), \widehat{\mathscr{C}})$ désigne la catégorie

- dont les objets sont les foncteurs familiaux $\mathscr{O}(\mathscr{A}) \to \widehat{\mathscr{C}}$ et

- dont les morphismes $F \to G$ sont les transformations naturelles α telles que pour tout F-générique strict $s : \mathbf{y}_c \to F(A, H)$, il existe une factorisation strictement G-générique du composé $\alpha_{(A,H)} \circ s$ de la forme

$$\mathbf{y}_c \xrightarrow{s'} G(A, K) \xrightarrow{G[\mathrm{id}_A]} G(A, H),$$

avec $H \leqslant K$.

Notons que si on obtient une bonne correspondance attendue sur les objets, on doit considérer des transformations naturelles non familiales pour capturer les transformations naturelles analytiques correspondantes. Ceci ne devrait finalement pas être trop surprenant : une transformation α est analytique lorsque pour tout s générique le composé $\mathbf{y}_c \xrightarrow{s} F(A) \xrightarrow{\alpha_A} G(A)$ l'est aussi, ce qui ne force en aucun cas leurs groupes d'automorphismes (respectivement en tant qu'objets de $\mathbf{y}_c/F$ et $\mathbf{y}_c/G$) à coïncider.

Démonstration. On se contente ici d'une esquisse, le lecteur intéressé pouvant consulter l'article pour plus de détails. Le point crucial est que pour tout $F : \mathscr{A} \to \widehat{\mathscr{C}}$, un morphisme $s : \mathbf{y}_c \to FX$ est générique ssi le morphisme induit $s' : \mathbf{y}_c \to F'(X, \mathfrak{S}_s)$ est strictement générique, où F' est le foncteur $\mathscr{O}(\mathscr{A}) \to \widehat{\mathscr{C}}$ correspondant à F comme dans la démonstration de la proposition 4.5.27. En effet, considérons un tel s générique et un carré commutatif

$$\begin{tikzcd}
\mathbf{y}_c \arrow[r, "t'"] \arrow[d, "s'"'] & F'(A, G) \arrow[d, "F'[\psi]"] \\
F'(X, \mathfrak{S}_s) \arrow[r, "F'[\varphi]"'] & F'(B, H).
\end{tikzcd}$$

On considère le carré extérieur de

$$\begin{tikzcd}
\mathbf{y}_c \arrow[r, "t'"] \arrow[d, "s'"'] \arrow[rr, bend left, "t"] \arrow[dd, bend right, "s"'] & F'(A, G) \arrow[r, "F'(q_G)"] \arrow[d, "F'[\psi]"] & F(A) \arrow[dd, "F(\psi)"] \\
F'(X, \mathfrak{S}_s) \arrow[r, "F'[\varphi]"'] \arrow[d, "q_{\mathfrak{S}_s}"'] & F'(B, H) \arrow[ur, dashed, "F(\gamma)"] \arrow[dr, "F'(q_H)"] & \\
F(X) \arrow[rr, "F(\varphi)"'] \arrow[uurr, dashed] & & F(B)
\end{tikzcd}$$

La généricité de s nous donne un γ comme indiqué en pointillés, faisant commuter les deux triangles. On va montrer que $[\gamma]$ est un relèvement pour le carré de départ. Pour commencer, il nous faut montrer que $[\gamma]$ est bien morphisme $(X, \mathfrak{S}_s) \to (A, G)$.

Or, par définition de (A, G), tout $g \in G$ fixe q_G, c'est-à-dire qu'on a $[g] \circ q_G = q_G$. Ceci entraîne que

$$F(\gamma) \circ s = t = F'(q_G) \circ t' = F'[g] \circ F'(q_G) \circ t' = F(g) \circ t = F(g) \circ F(\gamma) \circ s,$$

donc que l'extérieur du diagramme suivant commute :

$$\begin{tikzcd}
\mathbf{y}_c \arrow[rr, "s"] \arrow[dd, "s"'] & & F(X) \arrow[d, "F(\gamma)"] \\
& & F(A) \arrow[d, "F(g)"] \\
F(X) \arrow[rr, "F(\gamma)"'] \arrow[uurr, dashed, "F(\delta)"] & & F(A).
\end{tikzcd}$$

Par généricité de s on trouve δ comme indiqué en pointillés, faisant commuter les deux triangles. Par le lemme 4.5.18, δ est un iso et par construction il fixe s. On a donc trouvé $\delta \in \mathfrak{S}_s$ tel que $g \circ \gamma \circ \delta = \gamma$, ce qui entraîne que $[g]$ est bien un morphisme $(X, \mathfrak{S}_s) \to (A, G)$.

On a enfin $\psi \circ \gamma = \varphi$ par construction de γ et $F'[\gamma] \circ s' = t'$ par monicité de $F'(A, G) \to F(A)$. $\square$

On consacre le reste de cette partie à expliquer :

Théorème 4.5.29. La catégorie $\mathrm{Ana}(\mathscr{C}, \widehat{\mathscr{D}})$ des foncteurs analytiques admet un objet terminal.

L'idée est d'utiliser une catégorie équivalente à $\mathrm{Fam}'(\mathscr{O}(\mathscr{C}), \widehat{\mathscr{D}})$. Commençons par décrire le processus sur la catégorie plus simple $\mathrm{Fam}(\mathscr{O}(\mathscr{C}), \widehat{\mathscr{D}})$.

Définition 4.5.20. Soit, pour toutes catégories $\mathscr{A}$ et $\mathscr{B}$ avec $\mathscr{B}$ petite, $\mathrm{el}_{\mathscr{B}} /\!/ \mathscr{A}$ la catégorie ayant pour objets les paires (B, M) d'un préfaisceau $B \in \widehat{\mathscr{B}}$ et d'un foncteur $M : \mathrm{el}\, B \to \mathscr{A}$, dont les morphismes $(B, M) \to (C, N)$ sont les paires (α, φ) de transformations naturelles $\alpha : B \to C$ et

Lemme 4.5.30. L'assignation

$$\begin{tikzcd}
\mathrm{el}\, B \arrow[rr, "\mathrm{el}\,\alpha"] \arrow[dr, "M"'] & \arrow[d, phantom, "\overset{\varphi}{\underset{\cong}{\Leftarrow}}"] & \mathrm{el}\, C \arrow[dl, "N"] \\
& \mathscr{A}. &
\end{tikzcd}$$

$$\begin{aligned}
\mathrm{Sp} \colon (\mathrm{el}_{\mathscr{B}} /\!/ \mathscr{A}) &\to \mathrm{Fam}(\mathscr{A}, \widehat{\mathscr{B}}) \\
(B, M) &\mapsto X \mapsto b \mapsto \textstyle\sum_{x \in B(1)} \mathscr{A}(M(x), X)
\end{aligned}$$

s'étend en une équivalence de catégories.

Démonstration. Le point subtil de la démonstration est que, bien qu'une transformation familiale $\beta : F \to G$ préserve les génériques stricts par définition, elle ne préserve pas forcément le choix de factorisations génériques induit par les représentations comme coproduits de représentables. On peut par exemple avoir deux objets différents mais isomorphes, disons via $k : A \xrightarrow{\cong} A'$, dans $\mathscr{A}$, auquel cas chaque générique strict $s : \mathbf{y}_b \to F(A)$ donne aussi lieu à un générique $F(k) \circ s$. Dans ce cas, E_F fait un choix : soit $x = F(!) \circ s : \mathbf{y}_b \to F(1)$; comme $F(X)(b) \cong \sum_{x \in F(1)(b)} \mathscr{A}(E_F(x), X)$, on a une factorisation générique stricte via $E_F(x)$, canoniquement isomorphe aux autres mais mise en avant par la représentation $(F(1), E_F)$.

Commençons par définir le foncteur Sp sur les morphismes. Etant donné un morphisme $(\alpha, \varphi) : (B, M) \to (C, N)$ dans $\mathrm{el}/\!/\mathscr{A}$, on définit $\left(\mathrm{Sp}(\alpha, \varphi)_X\right)_b : \mathrm{Sp}(B, M)(X)(b) \to \mathrm{Sp}(C, N)(X)(b)$ comme envoyant tout $(x, f) \in \sum_{x \in B(b)} \widehat{\mathscr{B}}(M(x), X)$ sur la paire constituée de $\alpha_b(x) \in C(b)$ et du morphisme

$$N(\alpha_b(x)) \xrightarrow{\varphi_{(b,x)}} M(x) \xrightarrow{f} X.$$

La naturalité en b des $\left(\mathrm{Sp}(\alpha, \varphi)_X\right)_b$ découle facilement de celles de α et φ. La naturalité en X est facile, ainsi que la fonctorialité. La familialité de $\mathrm{Sp}(\alpha, \varphi)$ est aussi évidente par construction.

On a déjà plus ou moins montré que le foncteur Sp ainsi défini est essentiellement surjectif : on choisit pour tout F familial une factorisation strictement générique

$$y_b \xrightarrow{s_x} F(A_x) \xrightarrow{F!} F(1)$$

pour chaque $b \in \mathscr{B}$ et $x \in F(1)(b)$. On montre que l'assignation $x \mapsto A_x$ s'étend par généricité strict en un foncteur $E_F : \mathrm{el}(F(1)) \to \mathscr{A}$ et qu'on a $F \cong \mathrm{Sp}(F(1), E_F)$.

Montrons à présent que Sp est fidèle, considérons deux morphismes $(\alpha, \varphi), (\beta, \psi) : (B, M) \to (C, N)$ dans $\mathrm{el}/\!/\mathscr{A}$, envoyés sur la même transformation naturelle, disons γ. Pour tout $b \in \mathscr{B}$ et $x \in B(b)$, la composante

$$\sum_{x \in B(b)} \mathscr{A}(M(x), M(x)) \to \sum_{x \in C(b)} \mathscr{A}(N(x), M(x))$$

de $\mathrm{Sp}(\alpha, \varphi)$ en $M(x)$ et b envoie l'identité sur $(\alpha_b(x), \varphi_{(b,x)})$, alors que $\mathrm{Sp}(\beta, \psi)$ l'envoie sur $(\beta_b(x), \varphi_{(b,x)})$, ce qui implique $(\alpha, \varphi) = (\beta, \psi)$.

Pour montrer que Sp est plein, considérons une transformation naturelle familiale $\gamma : \mathrm{Sp}(B, M) \to \mathrm{Sp}(C, N)$. Pour tout $b \in \mathscr{B}$ et $x \in B(b)$, $\gamma(M(x))$ envoie $(x, \mathrm{id}_{M(x)})$ sur une paire constituée d'un certain $\alpha_b(x) \in C(b)$ et d'un morphisme $\varphi_{(b,x)} : N(\alpha_b(x)) \to M(x)$. Par construction, le diagramme

$$\begin{tikzcd}
b \arrow[r, "{(x,\mathrm{id})}"] \arrow[d, "{(\alpha_b(x),\mathrm{id})}"'] & F(M(x)) \arrow[r, "\alpha_{M(x)}"] & G(M(x)) \arrow[d, equal] \\
G(N(\alpha_b(x))) \arrow[rr, "G(\varphi_x)"'] \arrow[urr, "G(\varphi_x)"] & & G(M(x))
\end{tikzcd}$$

commute. Or par familialité de γ, les côtés gauche et haut de ce diagramme sont génériques, donc φ_x est un isomorphisme par le lemme 4.5.18.

Pour tout $x_0 \in B(b)$ et $f : M(x_0) \to X$, le carré de naturalité de γ en f et en b, pris en (x_0, id), donne :

$$(\gamma_X)_b(x_0, f) = (\alpha_b(x_0), N(\alpha_b(x_0)) \xrightarrow{\varphi_{(b,x_0)}} M(x_0) \xrightarrow{f} X). \tag{5}$$

Si on montre que α et φ sont naturelles, ce résultat entraîne que $\gamma = \mathrm{Sp}(\alpha, \varphi)$. En utilisant *(5)*, pour tout $h : b' \to b$ et $x_0 \in B(b)$, la commutativité du carré de naturalité

$$\begin{tikzcd}
\sum_{x \in B(b)} \widehat{\mathscr{B}}(M(x), M(x_0)) \arrow[r, "{(\gamma_{M(x_0)})_b}"] \arrow[d, "{\mathrm{Sp}(B,M)(M(x_0))(h)}"'] & \sum_{x' \in C(b)} \widehat{\mathscr{B}}(N(x'), M(x_0)) \arrow[d, "{\mathrm{Sp}(C,N)(M(x_0))(h)}"] \\
\sum_{y \in B(b')} \widehat{\mathscr{B}}(M(y), M(x_0)) \arrow[r, "{(\gamma_{M(x_0)})_{b'}}"'] & \sum_{y' \in C(b')} \widehat{\mathscr{B}}(N(y'), M(x_0))
\end{tikzcd}$$

prise en (x_0, id) donne

- $(\alpha_b(x_0) \cdot h, N(\alpha_b(x_0) \cdot h) \xrightarrow{N(h)} N(\alpha_b(x_0)) \xrightarrow{\varphi_{(b,x_0)}} M(x_0))$ par en haut et
- $(\alpha_{b'}(x_0 \cdot h), N(\alpha_{b'}(x_0 \cdot h)) \xrightarrow{\varphi_{b',x_0 \cdot h}} M(x_0 \cdot h) \xrightarrow{M(h)} M(x_0))$ par en bas,

soit précisément la naturalité de α et φ. □

On peut étendre ce résultat à $\mathrm{Fam}'(\mathscr{O}(\mathscr{C}), \widehat{\mathscr{D}})$:

Définition 4.5.21. Soit, pour toutes catégories $\mathscr{A}$ et $\mathscr{B}$, $\mathrm{el}_{\mathscr{B}} /\!/_{\nu} \mathscr{O}(\mathscr{A})$ la catégorie définie comme $\mathrm{el}_{\mathscr{B}} /\!/ \mathscr{O}(\mathscr{A})$, sauf qu'on relâche la condition que φ soit un isomorphisme en demandant seulement que φ soit *verticale*, c'est-à-dire que pour tout $(b, x) \in \mathrm{el}\, B$, $\varphi_{(b,x)} \in \mathscr{O}(\mathscr{A})(M(b, x), N(b, \alpha_b(x)))$, vue comme une classe d'équivalence de morphismes

$|M(b, x)| \to |N(b, \alpha_b(x))|$, ne contienne que des isomorphismes (en notant $|(A, G)| = A$).

Lemme 4.5.31. L'assignation

$$\begin{aligned} \mathrm{Sp}' \colon \left(\mathrm{el}_{\mathscr{B}} /\!/_{\nu} \mathscr{O}(\mathscr{A})\right) &\to \mathrm{Fam}'(\mathscr{O}(\mathscr{A}), \widehat{\mathscr{B}}) \\ (B, M) &\mapsto X \mapsto b \mapsto \textstyle\sum_{x \in B(1)} \mathscr{O}(\mathscr{A})(M(x), X) \end{aligned}$$

s'étend en une équivalence de catégories.

Preuve du théorème 4.5.29. On se limite encore une fois à décrire les idées essentielles, les détails étant disponibles dans l'article. Le point principal ici consiste à réaliser qu'étant donné un foncteur analytique F, la forme d'une opération de type $b \in \mathscr{B}$ et d'arité $A \in \mathscr{A}$ est constituée non seulement d'un morphisme strictement générique $s : \mathbf{y}_b \to F'(A, G)$, mais également de ses « bords », c'est-à-dire, pour chaque $f : b' \to b$, la factorisation générique du composé $\mathbf{y}_{b'} \xrightarrow{\mathbf{y}_f} \mathbf{y}_b \xrightarrow{s} F'(A, G)$. En d'autres termes, la forme d'une opération est un foncteur $\mathscr{B}/b \to \mathscr{O}(\mathscr{A})$. L'objet terminal cherché est censé posséder une opération de chaque forme : si on choisit $\mathscr{A}$ squelettique, on peut ainsi tenter de définir $U(1)(d)$ comme l'ensemble des foncteurs $\mathscr{B}/b \to \mathscr{O}(\mathscr{A})$. Cet essai se révèle cependant trop naïf : même dans le cas ensembliste, il laisse la place à des formes possédant des automorphismes non triviaux, par exemple tout ensemble fini muni de son groupe trivial $\{\mathrm{id}\}$ de permutations. On se restreint donc aux foncteurs $\mathscr{B}/b \to \mathscr{O}(\mathscr{A})$ « maximalement quotientés ». □

Exemple 4.5.7. Dans le cas ensembliste, on arrive à l'ensemble U des paires $(n, \mathfrak{S}_n)$.

Nous avons ainsi suffisamment généralisé les foncteurs familiaux pour obtenir un objet terminal. Cependant, on peut aussi montrer :

Proposition 4.5.32. Les foncteurs analytiques ne sont en général pas stables par composition.

Le problème est ici dû au fait que les génériques pour un composé $\widehat{\mathscr{C}} \xrightarrow{F} \widehat{\mathscr{D}} \xrightarrow{G} \widehat{\mathscr{E}}$ sont de la forme

$$\mathbf{y}_e \xrightarrow{s} G(D) \xrightarrow{G(t)} G(F(C))$$

avec s et t respectivement G et F-génériques. Or, les F-génériques relatifs à D sont obtenus comme colimites de foncteurs $s : \mathrm{el}\, D \to \widehat{\mathscr{D}}/F$ associant à chaque $x \in D(d)$ un F-générique relatif à $\mathbf{y}_d$. Si F était familial, on saurait que les génériques stricts sont stables par de telles colimites. Mais dans le cas analytique, un F-générique étant déterminé par un

F'-générique strict, l'existence de F-génériques est conditionnée par la cocomplétude (finie) de $\mathscr{O}(\widehat{\mathscr{C}})$, qui s'avère tout simplement fausse.

Considérons par exemple dans $\mathscr{O}(\mathrm{Set})$ le span

$$(2,1) \xleftarrow{[\mathrm{id}]} (2, \mathfrak{S}_2) \xrightarrow{[\mathrm{id}]} (2,1).$$

On peut montrer que ce span n'admet aucun pushout. Intuitivement, le seul candidat plausible est $(2,1)$ lui-même, mais on montre qu'il ne convient pas en considérant le carré

$$\begin{tikzcd} (2, \mathfrak{S}_2) \arrow[r, "{[\mathrm{id}]}"] \arrow[d, "{[\mathrm{id}]}"'] & (2,1) \arrow[d, "{[\mathrm{id}]}"] \\ (2,1) \arrow[r, "{[\sigma]}"'] & (2,1), \end{tikzcd}$$

où σ désigne l'automorphisme non trivial de 2.

En effet, si $(2,1)$ était le pushout recherché, on aurait une fonction $m : 2 \to 2$ telle que $\sigma = m = \mathrm{id}$ ce qui est clairement impossible. En partie suivante, on répare le problème en restreignant les arités à certaines colimites particulières.

4.6 Foncteurs analytiques cellulaires

Dans l'exemple des multicatégories, les arités nécessaires s'obtiennent toutes à partir du multigraphe vide en « recollant des représentables le long de leur bord ». Autrement dit, tout arbre A est codomaine d'une composée de la forme

$$\emptyset = A_0 \xrightarrow{f_1} A_1 \ldots A_{n-1} \xrightarrow{f_n} A_n = A, \tag{6}$$

où chaque f_i est obtenu par un pushout de la forme

$$\begin{tikzcd} \emptyset \arrow[r, "!"] \arrow[d] & \star \arrow[d] \\ A_{i-1} \arrow[r, "f_i"'] & A_i \arrow[ul, phantom, "\ulcorner"] \end{tikzcd} \quad \text{ou} \quad \begin{tikzcd} \star \arrow[r, "\partial"] \arrow[d] & {[p]} \arrow[d] \\ A_{i-1} \arrow[r, "f_i"'] & A_i. \arrow[ul, phantom, "\ulcorner"] \end{tikzcd} \tag{7}$$

Ici, par le lemme de Yoneda, ∂ est soit l'un des $s_j : \star \to [p]$, soit $t : \star \to [p]$. En fait pour le cas des arbres on pourrait même se restreindre à t seul puisqu'on peut d'abord introduire la racine et ensuite ajouter les nœuds successifs en « descendant » dans l'arbre.

Or on peut montrer que pour tout (A, G) dans $\mathscr{O}(\widehat{\mathbb{M}})$, $\partial : \star \to [p]$ et $x : (\star, 1) \to (A, G)$, le span $(A, G) \xleftarrow{x} (\star, 1) \xrightarrow{\partial} ([p], 1)$ admet un pushout.

Par analogie avec la topologie on pose :

Définition 4.6.22. Un préfaisceau $A \in \widehat{\mathbb{M}}$ est un *complexe cellulaire* s'il est de la forme *(6)*.

Grâce aux propriétés de clôture des complexes cellulaires et modulo quelques hypothèses supplémentaires, on peut montrer que tout endofoncteur analytique F sur $\widehat{\mathbb{M}}$ dont les arités sont des complexes cellulaires admet des génériques non seulement relativement aux représentables, mais aussi à tous les complexes cellulaires. De tels foncteurs sont donc stables par composition !

Définition 4.6.23. On appelle *cellulaire* tout endofoncteur analytique F sur $\widehat{\mathbb{M}}$ dont les arités sont des complexes cellulaires et tels que la factorisation générique de tout composé $\star \xrightarrow{\partial} [p] \xrightarrow{s} F(A)$ en

$$\begin{tikzcd}
\star \arrow[r, "\partial"] \arrow[d, "t"'] & {[p]} \arrow[d, "s"] \\
F(B) \arrow[r, "F(E_F(\partial))"'] & F(A)
\end{tikzcd}$$

satisfait les deux conditions suivantes :

1. $E_F(\partial)$ est un composé de pushouts d'une des formes ci-dessus;
2. tout automorphisme $\gamma \in \mathfrak{S}_t$ admet une *extension minimale* dans $\mathfrak{S}_s$ le long de $E_F(\partial)$, c'est-à-dire un $\delta \in \mathfrak{S}_s$ tel que $\delta \circ E_F(\partial) = E_F(\partial) \circ \gamma$ et pour tout $f : A \to C$ tel que $f \circ E_F(\partial) = f \circ E_F(\partial) \circ \gamma$ on ait $f \circ \delta = f$.

Remarque 4.6.4. La première condition demande que le morphisme d'arités induit par le morphisme de bord ∂ corresponde aussi à un recollement de représentables le long de leur bord. La seconde condition est satisfaite automatiquement dès que les arités des opérations de type $\star$ sont triviales, c'est-à-dire que pour tout générique $\star \to F(A)$ on a $A \cong \star$. En effet, $\star$ n'ayant pas d'automorphismes non triviaux, la condition est vraie faute de combattants. La proposition suivante reste donc vraie si on remplace cette condition par la trivialité des opérations de type $\star$.

On peut montrer :

Proposition 4.6.33. Les endofoncteurs cellulaires sont stables par composition.

On peut généraliser la notion de cellularité à des catégories arbitraires. Ceci repose sur la notion de *bordage* décrite dans l'article. Cependant, on montre aussi que les foncteurs cellulaires n'ont pas d'objet terminal en général. On se rabat donc sur une classe

de foncteurs plus restreinte, obtenue en généralisant bêtement la condition alternative de la remarque *4.6.4*.

Définition 4.6.24. Une *catégorie à sens unique* est une catégorie $\mathscr{C}$ telle qu'aucun objet n'admet d'autmorphisme non trivial, munie d'un foncteur vers $\mathbf{2} = 0 \leqslant 1$, dont la fibre en 1 est discrète. .

Ces conditions interdisent tout morphisme non-identité entre deux objets de dimension 1. Elles forcent de plus les morphismes à aller de la dimension 0 à la dimension 1.

Exemple 4.6.8. La catégorie $\mathbb{M}$ donne lieu à une catégorie à sens unique en envoyant $\star$ sur 0 et tous les $[p]$ sur 1.

On appelle *dimension* de c, notée $\dim(c)$, l'image de tout objet $c \in \mathrm{ob}(\mathscr{C})$ par le foncteur $\mathscr{C} \to \mathbf{2}$ en question. Pour $i \in \{0, 1\}$, on note $\mathscr{C}_{|i}$ un choix de pullback

$$\begin{tikzcd} \mathscr{C}_{|i} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner"] & \mathscr{C} \arrow[d] \\ 1 \arrow[r, "\ulcorner i \urcorner"'] & \mathbf{2}, \end{tikzcd}$$

ou autrement dit la sous-catégorie pleine de $\mathscr{C}$ obtenue en se restreignant à la dimension i.

Définition 4.6.25. Pour toute catégorie $\mathscr{C}$ à sens unique, un endofoncteur F sur $\widehat{\mathscr{C}}$ est dit *cadré* ssi sa restriction à $\mathscr{C}_{|0}$ est naturellement isomorphe à un foncteur de la forme $X \mapsto A \times X$, pour un préfaisceau $A : \mathscr{C}_{|0}^{\mathrm{op}} \to \mathrm{Set}$.

Autrement dit, pour tout $c \in \mathrm{ob}(\mathscr{C})$ de dimension 0, il existe un ensemble A_c tel que pour tout $X \in \widehat{\mathscr{C}}$ on ait $F(X)(c) = A_c \times X(c)$, avec en plus une contrainte sur l'action des morphismes.

Si F est cadré, la seule arité permise relativement à chaque $\mathbf{y}_c$ avec $\dim(c) = 0$ est $\mathbf{y}_c$ lui-même, à unique isomorphisme près. La seconde condition de cellularité est donc automatiquement vérifiée pour tout morphisme *de bord*, c'est-à-dire s'envoyant sur $0 \leqslant 1$.

Pour chaque catégorie à sens unique, on peut définir un bordage qui, sans donner la définition, revient à remplacer *(7)* par l'ensemble des morphismes $\emptyset \to c$ pour tout c de dimension 0 et des morphismes de bords.

On obtient :

Proposition 4.6.34. Un endofoncteur analytique cadré sur une catégorie à sens unique est cellulaire ssi

- ses arités sont des complexes cellulaires et

– il vérifie la condition *(1)* de la définition 4.6.23.

Ces foncteurs sont stables par composition et possèdent un objet terminal (vis-à-vis des transformations naturelles analytiques).

La stabilité par composition est facile ; détaillons un peu l'existence d'un objet terminal. On a d'abord :

Lemme 4.6.35. Pour tout endofoncteur cellulaire et cadré, tout objet c de dimension 1 et tout morphisme générique $s : c \to F(A)$, tout automorphisme γ de A fixant s fixe aussi son bord, c'est-à-dire que pour tout morphisme de bord δ, on a $\gamma \circ E_F(\delta) = E_F(\delta)$.

Démonstration. On obtient par hypothèse la partie solide de

$$\begin{tikzcd}
c' \arrow[r, "\partial"] \arrow[rr, bend left, "t"] \arrow[d, "t"'] & c \arrow[dr, "s"] \arrow[rr, dashed] & & F(B) \arrow[d, "F(E_F(\partial))"] \\
F(B) \arrow[r, "F(E_F(\partial))"'] \arrow[urr, dashed, "F(h)"] & F(A) \arrow[u, "s"] \arrow[r, "F(\gamma)"'] & & F(A),
\end{tikzcd}$$

puis par généricité de t le morphisme h en pointillés, qui est nécessairement un automorphisme de B. Mais comme F est cadré et $\mathscr{C}$ est à sens unique, $B = c'$ n'a pas d'automorphisme non trivial donc $h = \mathrm{id}$ et on obtient le résultat voulu. □

On arrive à la définition suivante pour l'objet terminal, dans laquelle on note U_0 le foncteur analytique terminal du théorème 4.5.29. On rappelle que $U_0(1)(c)$ est défini comme l'ensemble (modulo isomorphisme) des foncteurs $\mathscr{C}/c \to \mathscr{O}(\widehat{\mathscr{C}}_f)$ maximalement quotientés.

Définition 4.6.26. Soit U le sous-endofoncteur analytique cadré de U_0 défini par le fait que pour tout $c \in \mathrm{ob}(\mathscr{C})$ de dimension 1, $U(1)(c)$ contient précisément les foncteurs $A : \mathscr{C}/c \to \mathscr{O}(\widehat{\mathscr{C}}_f)$ valant $(c', 1)$ sur chaque morphisme de bord $\partial : c' \to c$, tels que $A(\mathrm{id}_c) = (B, G)$, où G est le groupe de tous les automorphismes de B fixant le bord, c'est-à-dire les g tels que $g \circ F\langle\partial\rangle = F\langle\partial\rangle$ pour tout ∂.

Ici, chaque morphisme de bord $\partial : c' \to c$ est vu comme un morphisme $\langle\partial\rangle : \partial \to \mathrm{id}_c$ dans la tranche $\mathscr{C}/c$. Or F envoie ∂ sur $(c', 1)$, donc $F\langle\partial\rangle : (c', 1) \to (B, G)$ détermine un unique morphisme $c' \to B$, qu'on appelle aussi $F\langle\partial\rangle$. La condition revient à demander que G soit constitué de tous les automorphismes g de B tels que le diagramme suivant commute :

$$\begin{tikzcd} & c' \arrow[dl, "F\langle\partial\rangle"'] \arrow[dr, "F\langle\partial\rangle"] & \\ B \arrow[rr, "g"'] & & B\,. \end{tikzcd}$$

Observons au passage que ces foncteurs sont maximalement quotientés, puisque sur chaque morphisme de bord le groupe trivial est aussi le groupe total (parce que $\mathscr{C}$ est à sens unique) et sur id_c, il faut fournir un morphisme dans $\mathscr{O}(\widehat{\mathscr{C}}_f)$, donc en particulier respecter la condition de la définition 4.5.18. Aucun groupe valide ne peut donc contenir de morphisme ne fixant pas le bord.

Résumons : on a d'abord considéré les foncteurs familiaux, qui étaient stables par composition mais ne possédaient qu'un objet faiblement terminal ; on a ensuite examiné les foncteurs analytiques, plus généraux, dont on a montré qu'ils possédaient bien un objet terminal, mais qui se sont avérés ne pas être stables par composition ; en se restreignant à la notion intermédiaire de foncteur cellulaire et cadré, on obtient enfin une sous-catégorie monoïdale de $\mathrm{CAT}(\widehat{\mathscr{C}}, \widehat{\mathscr{C}})$ munie d'un objet terminal. On a donc automatiquement :

Proposition 4.6.36. L'objet terminal U défini ci-dessus est une monade.

Du coup, tout sous-objet T de U tel que $T \circ T \to U$ et $\mathrm{id} \to U$ se factorisent par T est automatiquement une sous-monade de U. On peut enfin poser :

Définition 4.6.27. Pour toute catégorie $\mathscr{C}$ à sens unique, une monade *morphologique* sur $\widehat{\mathscr{C}}$ est une sous-monade de U.

On peut en fait caractériser les monades morphologiques en termes de l'ordre partiel $\mathrm{Sub}(U)$ des sous-objets de U. En effet, l'identité est déjà un sous-objet, donc la condition que $\mathrm{id} \to U$ se factorise par T est équivalente à $\mathrm{id} \leqslant T$. De plus, en définissant pour tous F, $G \in \mathrm{Sub}(U)$ le *produit morphologique* $F \boxtimes G$ comme étant l'image du composé

$$F \circ G \xrightarrow{!\circ!} U \circ U \xrightarrow{!} U\,,$$

la condition que $T \circ T \to U$ se factorise par T revient à demander $T \boxtimes T \leqslant T$. On peut être ici tenté de dire que T est une monade morphologique ssi T est un monoïde dans $\mathrm{Sub}(U)$ (pour l'opération $\boxtimes$). En effet, le produit morphologique munit presque $\mathrm{Sub}(U)$ d'une structure de quantale, le problème étant qu'il n'est associatif que dans un sens relâché : on peut montrer par propriété universelle des images que $(F \boxtimes G) \boxtimes H \leqslant F \boxtimes (G \boxtimes H)$ pour tous F, G et H. Cette structure est une instance de la notion de *monoïdale penchée* (*skew monoidale* en anglais) de Lack et Street [46], prise ici dans la 2-catégorie des ordres partiels complets

dirigés, munie du produit tensoriel classifiant les morphismes bilinéaires. Concrètement, cela donne la notion suivante :

Définition 4.6.28. Un sous-ensemble D d'un ordre partiel X est *dirigé* lorsqu'il est non vide et tel que pour tous $x, x' \in D$, il existe $y \in D$ tel que $x \leqslant y$ et $x' \leqslant y$.

Une *quantale penchée (à gauche)* est un ordre partiel complet dirigé X muni d'un élément $1 \in X$ et d'une loi binaire $\cdot$ préservant les sups dirigés des deux côtés et telle que $(x \cdot y) \cdot z \leqslant x \cdot (y \cdot z)$, $1 \cdot x \leqslant x$ et $x \leqslant x \cdot 1$ pour tous $x, y, z \in X$. Une quantale penchée est *normale* quand les deux dernières inéquations sont en fait des égalités.

Comme les foncteurs considérés sont finitaires, on montre facilement que le produit morphologique $\boxtimes$ préserve les sups dirigés des deux côtés. De plus, on a le résultat suivant, dont on omet la démonstration :

Lemme 4.6.37. L'ordre partiel $\mathrm{Sub}(U)$ est isomorphe à l'ordre partiel des sous-objets de $U(1)$ dans $\widehat{\mathscr{C}}$.

On obtient comme corollaire que $\mathrm{Sub}(U)$ est un treillis complet et par suite :

Proposition 4.6.38. L'identité et le produit morphologique $\boxtimes$ munissent $\mathrm{Sub}(U)$ d'une structure de quantale penchée normale.

L'avantage de ceci est que dans les quantales penchées on peut définir une notion naturelle de monoïde, pour laquelle on peut facilement construire des monoïdes libres.

Définition 4.6.29. Un *monoïde* dans une quantale penchée X est élément t tel que $1 \leqslant t$ et $t \cdot t \leqslant t$.

En adaptant un travail de Lack [45], on obtient une construction inductive des monoïdes :

Proposition 4.6.39. Dans toute quantale penchée X, pour tout $x \in X$, il existe un plus petit monoïde contenant x, donné par la formule

$$\bigvee_{n \in \mathbb{N}} (\mathrm{id} \vee x)^{\cdot n},$$

où $y^{\cdot 0} = 1$ et $y^{\cdot (n+1)} = y \cdot (y^{\cdot n})$ pour tout n.

Corollaire 4.6.40. La monade morphologique librement engendrée par un sous-objet $\Sigma \hookrightarrow U$ existe et est donnée par

$$\bigcup_{n \in \mathbb{N}} (\mathrm{id} \cup x)^{\boxtimes n}.$$

On peut enfin obtenir la monade M des multicatégories comme monade morphologique librement engendrée par la famille gentree. En effet, cette famille donne lieu à un sous-objet du foncteur analytique, cellulaire et cadré terminal U. Pour le voir, observons qu'un foncteur cadré $\mathbb{M}/[n] \to \widehat{\mathbb{M}}$ est déterminé à isomorphisme près par un objet de $(n+1)\cdot\star/\widehat{\mathbb{M}}$, c'est-à-dire un élément de $\mathrm{gentree}_n$. On pose donc $\Sigma_M(\star) = U(\star)$ et $\Sigma_M[n] = \mathrm{gentree}_n$ et on obtient après un peu de travail :

Théorème 4.6.41. M est la monade morphologique librement engendrée par Σ_M.

Remarque 4.6.5. L'article propose une notion de cellularité beaucoup plus générale que celle utilisée ici, qui permet dans une certaine mesure de choisir ce qu'on considère comme le bord d'une arité. Cette approche permet de restaurer la stabilité par composition, mais pas toujours d'obtenir un objet terminal. Les seuls exemples où on réussit à concilier les deux, donc à engendrer des monades morphologiques de manière satisfaisante, utilisent la notion de foncteur cadré. On obtient de cette manière, en plus des multicatégories, les multicatégories symétriques et les polycatégories. Des recherches en cours semblent indiquer qu'on peut aussi retrouver les PROPs, ce qui semble prometteur au vu de l'intérêt considérable récemment reçu par ces derniers.

5 De la sémantique des jeux à la syntaxe

[P]our inciter les paysans à produire, il est bon de leur donner le goût du superflu.

Raymon Aron, expliquant Montesquieu dans *Les étapes de la pensée sociologique*, p. 48, Gallimard, 1976.

5.1 Introduction

Cette partie est une introduction aux stratégies innocentes vues comme des faisceaux.

5.1.1 Corpus couvert et historique

Les articles avec D. Pous [35], en solo [31] et avec C. Eberhart et T. Seiller [15, 16] couvrent toutes mes contributions à ce sujet, auxquelles on peut ajouter les ancêtres écrits en famille [33, 34] sur la logique linéaire. Ces derniers mettent déjà en avant l'idée de partie multi-joueurs, mais avec des outils beaucoup plus primitifs, tant pour décrire les parties que pour décrire les morphismes entre elles. Nous avions à l'époque l'intuition que les parties devaient former un *champ* [75], les stratégies devant être définies en gros comme des sous-champs. Les faisceaux n'étaient pas loin, mais cette intuition était trop grossière pour rendre compte de l'innocence. De plus, bien que dans les travaux présentés ici les parties ne forment un champ ni pour CCS ni pour π, je suis convaincu qu'il est possible d'en faire un champ en présentant les choses différemment. Et j'espère que cela pourrait contribuer à éclairer les liens mystérieux entre sémantique par réduction et systèmes de transitions étiquetées [68].

5.1.2 Les traces multi-joueurs comme pont entre modèles de jeux et syntaxe

S'il faut retenir quelque chose de cette partie, ce sont les deux idées suivantes.

- On a vu en partie précédente que les préfaisceaux (finiment présentables) fournissent des outils de haut niveau pour décrire des structures graphiques. Ceci s'applique en particulier aux parties — éventuellement multi-joueurs — dans les modèles de jeux innocents.
- En munissant ces parties multi-joueurs d'une notion suffisamment riche de morphisme, on obtient un site de Grothendieck (toute partie étant couverte par ses vues), sur lequel les faisceaux définissent une notion adéquate de stratégie innocente (chaque joueur n'a accès qu'à son propre historique pour décider de son prochain coup) et concurrente (la partie globale se calcule de manière non-déterministe selon les offres faites par les joueurs en présence).

La structure de cette partie reflète cette construction en deux couches, en se focalisant sur un exemple, le *Calculus of Communicating Systems* (CCS) de Milner [57].

En partie 5.2, on décrit la première couche, les parties multi-joueurs et leur représentation à l'aide de préfaisceaux finiment présentables. Plus précisément, on construit une pseudo catégorie double de parties multi-joueurs, dont les objets sont les positions du jeu, les morphismes verticaux sont les parties. Les morphismes horizontaux et les cellules représentent une forme d'inclusion spatiale. Par exemple, dans un morphisme horizontal $X \to Y$, X peut représenter la restriction de la position Y à seulement certains de ses joueurs. De même, dans une cellule $u \to u'$ reliant deux parties u et u', u peut représenter la restriction de la partie u' à certains de ses joueurs.

En partie 5.3, on s'attaque à la seconde couche. La première étape consiste à construire à partir de la pseudo catégorie double des parties, pour chaque position de départ fixée X, une catégorie de parties $\mathbb{P}(X)$ adéquate. On présente certaines constructions naïves en démontrant leur inadéquation, puis on exhibe la construction retenue. On explique ensuite en quoi les préfaisceaux sur $\mathbb{P}(X)$ offrent une première notion de stratégie et on montre en quoi celle-ci est trop naïve pour représenter fidèlement les processus CCS. On introduit alors la notion de *comportement innocent* sur X, c'est-à-dire de préfaisceaux sur une catégorie *ad hoc* de *vues* sur X. Nous faisons ensuite le lien entre comportements innocents et stratégies naïves en construisant la stratégie naïve librement engendrée par tout comportement innocent. En d'autres termes, on construit une adjonction entre les deux catégories.

Enfin, en partie 5.4, on énonce le résultat d'adéquation, qui décrit en quoi la notion de comportement innocent modélise finement les processus CCS. Ceci repose sur la notion d'*équivalence de test* [13] et la construction d'un système de transitions à partir des comportements innocents. On montre aussi que l'adjonction entre stratégies naïves et comportements innocents permet d'exprimer l'équivalence de test équitable sur ces derniers uniquement en termes ludiques, c'est-à-dire sans référence au système de transitions.

On conclut en partie 5.5 en donnant quelques idées de travaux futurs.

5.2 Parties multi-joueurs

5.2.1 Structure globale : premières intuitions sur les terrains de jeux

Dans les parties suivantes, on décrit plusieurs exemples de jeux pour donner une idée de la généralité de l'approche. En préambule, on tente dans cette partie de donner quelques premières intuitions sur ce qu'on entend ici par « jeu ». Formellement, il s'agit des *terrains de jeux*, introduits et étudiés dans la version longue du travail sur CCS [31].

Comme certains jeux considérés sont multi-joueurs, un terrain de jeux comprend naturellement un ensemble de *positions* censé décrire l'ensemble des configurations possibles.

Exemple 5.2.1. En CCS, une position X consiste en un ensemble fini $X(\star)$ de canaux de communications, un ensemble fini $X[n]$ de joueurs d'arité n pour chaque $n \in \mathbb{N}$, ainsi que, pour chaque n et $x \in X[n]$, d'un choix de n canaux dans $X(\star)$. Autrement dit, une configuration est un préfaisceau fini sur la catégorie $\mathbb{C}_1$ librement engendrée par le graphe constitué

- d'un sommet $\star$ ainsi que d'un sommet $[n]$ pour tout $n \in \mathbb{N}$,
- de n arêtes $s_1, \ldots, s_n : \star \to [n]$.

On tolère ici un conflit de notation : pour chaque n on a un morphisme différent $s_1 : \star \to [n]$, qu'il faudrait distinguer des autres. Pour éviter une lourdeur superflue on s'en dispense, en s'appuyant sur le contexte pour lever les ambiguïtés.

Définition 5.2.1. Munis des transformations naturelles injectives comme morphismes, ces préfaisceaux finis forment une catégorie $\mathbb{D}_h$.

Remarque 5.2.1. On se restreint ici aux morphismes injectifs, parce que cela suffit pour CCS. Le travail sur π fait usage de morphismes légèrement plus généraux.

Exemple 5.2.2. On peut imaginer un terrain de jeu où les positions sont constituées de joueurs jouant simultanément entre eux plusieurs parties d'échecs. Chaque joueur x voit donc un ensemble fini d'échiquiers, chacun dans une position arbitraire. Sur chacun de ces échiquiers, x joue soit les blancs soit les noirs. Une configuration comprend ainsi

- une famille finie de positions, c'est-à-dire une famille (finie) d'ensembles indexée par l'ensemble Pos des positions d'échiquiers ; pour toute configuration X et toute position A, on note $X(A)$ l'ensemble correspondant à A, qui décrit l'ensemble des échiquiers qui sont dans la position A ; l'ensemble de tous les échiquiers de X est donc $\sum_A X(A)$;
- pour chaque *séquent* $S = (A_1, \ldots, A_n \vdash B_1, \ldots, B_p)$, un ensemble $X(S)$ représentant les joueurs dont les échiquiers sont dans les positions respectives $A_1, \ldots, A_n \vdash B_1, \ldots, B_p$, les A_i étant par convention ceux où le joueur en question joue les blancs ;
- pour chaque joueur $x \in X(S)$ et chaque $i \in n$, un échiquier $s_i(x) \in X(A_i)$;

- de même, pour chaque joueur $x \in X(S)$ et chaque $i \in p$, un échiquier $t_i(x) \in X(B_i)$.
  Ces données doivent de plus vérifier la condition suivante :

$$\text{Chaque échiquier } e \in X(A) \text{ est dans l'image d'au plus un } s_i \text{ et un } t_j. \qquad (8)$$

Cela signifie qu'au plus un joueur y joue les blancs et qu'au plus un joueur y joue les noirs.

Autrement dit, les positions sont des préfaisceaux finis sur la catégorie $\mathbb{C}$ librement engendrée par le graphe constitué
- d'un sommet $[A]$ pour chaque position d'échiquier A,
- d'un sommet $[S]$ pour chaque séquent $S = (A_1, \dots, A_n \vdash B_1, \dots, B_p)$,
- d'arêtes $s_i : [A_i] \to S$ pour chaque $i \in n$,
- d'arêtes $t_i : [B_i] \to S$ pour chaque $i \in p$.

Cependant, contrairement à l'exemple précédent, tous les préfaisceaux ne sont pas des positions valides :

Définition 5.2.2. Une *position* est un préfaisceau sur $\mathbb{C}$ satisfaisant la condition *(8)*.

Comme précédemment, on a :

Proposition 5.2.1. Les positions, avec comme morphismes les transformations naturelles, forment une catégorie $\mathbb{D}_h$.

Dans les exemples précédents, les positions s'accompagnent naturellement d'une notion de morphisme. Un morphisme $X \to Y$ décrit un plongement de X dans Y. Or, dans la partie suivante, on s'attache à décrire formellement les parties pour CCS. Le point crucial pour comprendre la notion de partie est qu'en accord avec le principe de catégorification énoncé dans l'introduction, on souhaite décrire plus qu'une simple relation entre les positions de départ et d'arrivée d'une partie : on veut rendre compte du processus ayant mené de l'une à l'autre. Les parties forment ainsi une deuxième notion de morphisme entre positions. Par convention, une partie de position initiale X et de position finale Y sera vue comme un morphisme $Y \nrightarrow X$.

Exemple 5.2.3. Dans l'exemple *5.2.2*, une partie de position initiale X est constituée
- pour chaque joueur $x \in X\left(A_1^{\text{blancs}}, \dots, A_{n_{\text{blancs}}}^{\text{blancs}} \vdash A_1^{\text{noirs}}, \dots, A_{n_{\text{noirs}}}^{\text{noirs}}\right)$, d'une suite $\left(s_1^x, \dots, s_{n_x}^x\right)$ de triplets de la forme $s_i^x = (\lambda_i^x, p_i^x, c_i^x)$, où $\lambda_i^x \in \{\text{blancs}, \text{noirs}\}$, $p_i \in n_{\lambda_i^x}$ et c_i^x est un coup valide sur la position $A_i^{\lambda_i^x}$,
- ainsi que, pour chaque échiquier dans $e \in X(A)$, d'une suite $c_1^e, \dots, c_{n_e}^e$ de coups valides à partir de la position A,

telles que pour chaque joueur, la projection de la suite de coups sur chaque échiquier donne précisément la suite de coups dudit échiquier.

La position d'arrivée d'une telle partie se calcule comme suit :

- chaque échiquier $e \in X(A)$ est remplacé par un échiquier dans la position A'_e obtenue en jouant les coups $c^e_1, \ldots, c^e_{n_e}$ à partir de A;
- chaque position A^λ_i de chaque joueur $x \in X\left(A^{\text{blancs}}_1, \ldots, A^{\text{blancs}}_{n_{\text{blancs}}} \vdash A^{\text{noirs}}_1, \ldots, A^{\text{noirs}}_{n_{\text{noirs}}}\right)$ est remplacée par $A'_{s^\lambda_i(x)}$, où $s^{\text{blancs}} = s$ et $s^{\text{noirs}} = t$.

On est donc en présence de deux notions de morphismes partageant le même ensemble d'objets, ce qui correspond précisément à la description faite par Mac Lane [55] des *catégories doubles*, revues en définition 2.2.1.

En fait, les parties ne formeront pas une catégorie au sens strict, mais plutôt une bicatégorie. Dans notre cadre, un jeu sera donc organisé en une pseudo catégorie double (définition 2.2.2). L'intuition naturelle pour les cellules

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "u"'] \arrow[dr, phantom, "\Swarrow\alpha"] & B \arrow[d, "v"] \\ C \arrow[r, "g"'] & D \end{tikzcd}$$

est que la partie u se plonge dans v. Par exemple, u est la restriction de v à certains joueurs, ceux de l'image de g.

Exemple 5.2.4. Etant donnée une partie comme dans l'exemple *5.2.3*, disons $u : Y \rightarrowtail X$, on peut choisir un joueur $x \in X(\Gamma \vdash \Delta)$, qui, muni des échiquiers adjacents, définit une sous-position X_x de X. Les suites de coups de x et des échiquiers concernés définissent une partie $u_x : Y_x \rightarrowtail X_x$, qui induit en fait une cellule $u_x \to u$ dans $\mathbb{D}_H$.

5.2.2 Exemple détaillé : représentation des parties en CCS

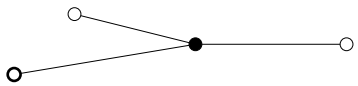
5.2.2.1 Générateurs

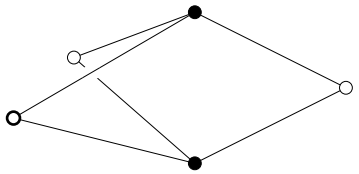
Comme annoncé en partie précédente, les parties de CCS décrivent le processus menant de leur position initiale à leur position finale. Elles sont engendrées par un ensemble de « générateurs » décrivant chacun un type d'étape, ou de coup, atomique dans une partie. On aura ainsi un générateur pour la réception de message, ainsi que d'autres pour l'émission, la synchronisation, la création d'un nouveau canal de communication, le lancement d'un nouveau processus (le *fork* en anglais), etc. Les joueurs de CCS étant munis d'une arité, le nombre de canaux auxquels ils ont accès, il nous faut un générateur pour chaque arité valide.

Dans le cadre de l'exemple *5.2.1*, pour décrire par exemple une étape de réception par un joueur ternaire sur son premier canal, on part du span $Y \xleftarrow{s} I \xrightarrow{t} X$ où X et Y sont respectivement les positions initiale et finale, et I est la partie de X censée rester stable au cours de la réception. Ici, $X = Y = [3]$ sont le préfaisceau représentable sur l'objet $[3] \in \mathrm{ob}(\mathbb{C}_1)$, alors que $I = \star + \star + \star$ est juste constituée des trois canaux de $[3]$. En procédant comme en récriture de graphes [3], on considère le pushout

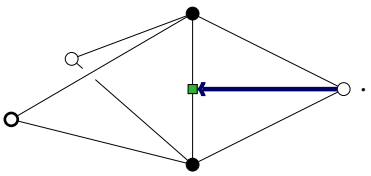
$$\begin{tikzcd} I \arrow[r] \arrow[d] & Y \arrow[d] \\ X \arrow[r] & Y +_I X . \end{tikzcd}$$

Intuitivement, ce dernier décrit la relation entre X et Y, mais ne décrit pas comment on passe de l'un à l'autre. Pour combler cette lacune, on constate qu'on peut représenter le pushout de manière « sphérique », un peu comme une coquille. En effet, si la position initiale est le diagramme de gauche ci-dessous alors le pushout est celui de droite, avec par convention le joueur final en haut :





L'idée est alors de « remplir » cette coquille avec l'information comme quoi le joueur du bas a effectué une réception sur son premier canal. Intuitivement, on souhaite arriver à la représentation graphique suivante :



(9)

Pour cela, on va ajouter un objet à notre catégorie de base $\mathbb{C}_1$, dont le dessin ci-dessus figurera le préfaisceau représentable. Pour représenter le lien entre joueurs et canaux dans la catégorie de base $\mathbb{C}_1$, on a traduit le fait qu'un joueur n-aire soit connecté à n canaux en introduisant les n morphismes $s_1, \ldots, s_n : \star \to [n]$. Procédons ici de manière similaire : chaque réception sera connectée à deux joueurs, le joueur prêt à recevoir et le joueur venant de recevoir. Les deux ayant (en CCS) la même arité, on introduit un nouvel objet $\iota_{3,1}$ (3 parce que le joueur considéré est ternaire, 1 parce qu'il reçoit sur son premier canal), muni de deux morphismes $s, t : [3] \to \iota_{3,1}$, t représentant le joueur initial et s le joueur

final. Cependant, pour correspondre à la représentation graphique souhaitée, ces deux morphismes doivent satisfaire quelques équations. Par exemple, on s'attend à ce que, pour chaque réception ternaire, le premier canal du joueur final coïncide avec celui du joueur initial. On peut incorporer cette contrainte à notre catégorie étendue en imposant que le diagramme

$$\begin{tikzcd} \bigstar \arrow[r, "s_1"] \arrow[d, "s_1"'] & {[3]} \arrow[d, "s"] \\ {[3]} \arrow[r, "t"'] & \iota_{3,1} \end{tikzcd}$$

commute. On impose bien sûr une contrainte similaire pour s_2 et s_3. On peut procéder sans problème de manière identique pour la réception sur le canal i d'un joueur n-aire, pour $i \in n$. On note l'objet ajouté $\iota_{n,i}$. Cette technique fonctionne aussi pour l'émission d'un message sur le canal i d'un joueur n-aire.

Le cas d'une création de nom est similaire. On ajoute un objet ν_n pour chaque arité n, muni de deux morphismes s et t, sauf que le morphisme s doit avoir le type $[n+1] \to \nu_n$. On demande de même $s \circ s_i = t \circ t_i$ pour tout $i \in n$.

$$\begin{tikzcd} {[n]} \arrow[r, "t"] & \pi_n & {[n]} \arrow[l, shift right, "s_1"'] \arrow[l, shift left, "s_2"] \end{tikzcd} \tag{10}$$

Le cas d'un *fork* est plus compliqué. Une première approche consiste à introduire pour chaque arité n un nouvel objet π_n muni cette fois de trois morphismes (puisqu'en *forkant*, le joueur considéré crée en fait deux avatars). Cela fonctionnerait parfaitement sans la question des vues. Pour définir les stratégies innocentes ci-dessous, nous aurons besoin d'une partie représentant, par exemple, ce que le premier avatar d'un *fork* retient du coup. Pour cela, on introduit d'abord deux coups qui formeront des vues, un pour l'avatar gauche, π_n^l, l'autre pour le droit, π_n^r munis chacun de morphismes $[n] \xrightarrow{t} \pi_n^\lambda \xleftarrow{s} [n]$ tels que $t \circ s_i = s \circ s_i$. On obtient donc un raffinement de *(10)* :

$$\begin{tikzcd} & \pi_n^l \arrow[d, "l"] & {[n]} \arrow[l, "s"'] \\ {[n]} \arrow[ur, "t"] \arrow[dr, "t"'] & \pi_n & \\ & \pi_n^r \arrow[u, "r"'] & {[n]}. \arrow[l, "s"] \end{tikzcd}$$

Remarque 5.2.2. Comme pour $\mathbb{C}_1$, on tolère un conflit de notation entre les différents morphismes notés s et t.

Le *fork* est une sorte de réunion de ces *forks partiels* gauche et droit π^l_n et π^r_n. Pour le représenter, on ajoute encore un objet π_n muni de morphismes $\pi^l_n \xrightarrow{l} \pi_n \xleftarrow{r} \pi^r_n$. A nouveau, ces morphismes doivent satisfaire une équation pour correspondre à CCS : pour tout *fork*, les *forks* gauche et droit associés doivent être joués par le même joueur. On impose ceci en ajoutant l'équation

$$l \circ t = r \circ s. \tag{11}$$

Le seul cas difficile restant est celui de la synchronisation. En fait, il ressemble beaucoup à celui du *fork* : une synchronisation doit avoir deux vues, une réception et une émission, disons $\iota_{n,i}$ et $o_{m,j}$, donc on ajoute un objet $\tau_{n,i,m,j}$ muni de morphismes $\iota_{n,i} \xrightarrow{\rho} \tau_{n,i,m,j} \xleftarrow{\varepsilon} o_{m,j}$. Comme le récepteur et l'émetteur sont deux joueurs différents, on n'impose pas l'équation analogue à *(11)*. En revanche, pour que la synchronisation fasse sens, il faut que le canal i du récepteur coïncide avec le canal j de l'émetteur. On impose donc l'équation $\rho \circ t \circ s_i = \varepsilon \circ t \circ s_j$.

En procédant ainsi pour tous les autres coups, à toutes les arités possibles on obtient :

Définition 5.2.3. Soit $\mathbb{C}$ la catégorie librement engendrée par le graphe ayant comme sommets

- ceux de $\mathbb{C}_1$,

ainsi que pour tout entier n

- des sommets $\iota_{n,i}$ et $o_{n,i}$ pour chaque $i \in n$, pour la réception et l'émission,
- un sommet ν_n, pour la création de nouveaux canaux,
- des sommets π^l_n, π^r_n et π_n, pour le *fork*,
- un sommet $\tau_{n,i,m,j}$ pour chaque m avec $i \in n$ et $j \in m$, pour la synchronisation,
- ainsi qu'un sommet $\heartsuit_n$ (prononcé « tick »), qui sert à définir les équivalences de tests à la manière du démon en ludique [28],

et pour arêtes

- celles de $\mathbb{C}_1$,

ainsi que pour tout entier n des arêtes

- $[n] \xrightarrow{t} \iota_{n,i} \xleftarrow{s} [n]$ et $[n] \xrightarrow{t} o_{n,i} \xleftarrow{s} [n]$ pour chaque $i \in n$,
- $[n] \xrightarrow{t} \nu_n \xleftarrow{s} [n+1]$,
- $[n] \xrightarrow{t} \pi^\lambda_n \xleftarrow{s} [n]$ pour $\lambda \in \{l, r\}$ et $\pi^l_n \xrightarrow{l} \pi_n \xleftarrow{r} \pi^r_n$,
- $\iota_{n,i} \xrightarrow{\rho} \tau_{n,i,m,j} \xleftarrow{\varepsilon} o_{m,j}$ pour tout $m, i \in n$ et $j \in m$
- et enfin $[n] \xrightarrow{t} \heartsuit_n \xleftarrow{s} [n]$,

avec les équations

$$\begin{tikzcd} \bigstar \arrow[r, "s_i"] \arrow[d, "s_i"'] & {[n]} \arrow[d, "s"] \\ {[n]} \arrow[r, "t"'] & c \end{tikzcd}$$

$\forall n \in \mathbb{N}, i, j \in n$ et $c \in \{\iota_{n,j}, o_{n,j}, \nu_n, \pi^l_n, \pi^r_n, \heartsuit_n\}$

$$\begin{tikzcd} {[n]} \arrow[r, "t"] \arrow[d, "t"'] & \pi^r_n \arrow[d, "r"] \\ \pi^l_n \arrow[r, "l"'] & \pi_n \end{tikzcd}$$

$\forall n \in \mathbb{N}$

$$\begin{tikzcd} {[n]} \arrow[rr, "t"] & & \iota_{n,i} \arrow[dr, "\rho"] & \\ & \bigstar \arrow[ul, "s_i"] \arrow[dl, "s_j"'] & & \tau_{n,i,m,j} \\ & {[m]} \arrow[rr, "t"'] & & \iota_{m,j} \arrow[ul, "\varepsilon"'] \end{tikzcd}$$

$\forall n, m \in \mathbb{N}, i \in n$ et $j \in m$.

La catégorie $\mathbb{C}$ nous fournit une base pour décrire des coups CCS, mais n'assure aucunement le respect de l'orientation temporelle.

Exemple 5.2.5. Considérons le pushout

$$\begin{tikzcd} {[1]} \arrow[r, "s"] \arrow[d, "s"'] & \iota_{1,1} \arrow[d] \\ \iota_{1,1} \arrow[r] & P. \end{tikzcd}$$

Le résultat, P, peut se voir comme une réception suivie d'une « déréception » (avec toutes les responsabilités qui lui décombent [53]), c'est-à-dire d'une réception à rebours dans le temps.

Pour donner la direction du temps, on va considérer les coups et les parties non plus seulement comme des préfaisceaux sur $\mathbb{C}$, mais comme des cospans $X \to M \leftarrow Y$, dont X est la position finale. Cette orientation est en accord avec les notations de la partie précédente, puisque M sera un morphisme $X \nrightarrow Y$. Pour définir ces cospans, on adopte la

Notation 5.2.1. On désigne par $[m]_{a_1,\dots,a_p|c_1,\dots,c_p}[n]$ la position constituée d'un joueur m-aire x et d'un joueur n-aire y, quotientée par les équations $x \cdot s_{a_k} = y \cdot s_{c_k}$ pour $k \in p$. Lorsque les deux listes sont vides, on les omet et par convention $m = n$ et les deux joueurs partagent tous leurs canaux, dans l'ordre.

Par le lemme de Yoneda, $[m]_{a_1,\dots,a_p|c_1,\dots,c_p}[n]$ peut se définir comme le pushout ci-dessous, donc pour tous morphismes $[m] \xrightarrow{f} X \xleftarrow{g} [n]$ faisant commuter l'extérieur du diagramme

$$\begin{tikzcd} p \cdot \star \arrow[r, "{[s_{c_i}]_{i \in p}}"] \arrow[d, "{[s_{a_i}]_{i \in p}}"'] & {[m]} \arrow[d] \arrow[ddr, bend left, "g"] & \\ {[n]} \arrow[r] \arrow[drr, bend right, "f"'] & {[m]_{a_1,\dots,a_p|c_1,\dots,c_p}[n]} \arrow[dr, dashed, "{[f,g]}"] & \\ & & X \end{tikzcd}$$

il existe un unique morphisme médian qu'on notera $[f, g]$ comme un copairage.

Voici les cospans adéquats pour chaque type de coup :

- Réception : $[n] \xrightarrow{s} \iota_{n,i} \xleftarrow{t} [n]$.

- Emission : $[n] \xrightarrow{s} o_{n,i} \xleftarrow{t} [n]$.
- Synchronisaton : $[n]\,{}_j|_i\,[n] \xrightarrow{[\varepsilon\circ s,\,\rho\circ s]} \pi_n \xleftarrow{[\varepsilon\circ t,\,\rho\circ t]} [n]\,{}_j|_i\,[n]$.
- Création : $[n+1] \xrightarrow{s} \nu_n \xleftarrow{t} [n]$.
- Tick : $[n] \xrightarrow{s} \heartsuit_n \xleftarrow{t} [n]$.
- Fork gauche : $[n] \xrightarrow{s} \pi_n^l \xleftarrow{t} [n]$.
- Fork droit : $[n] \xrightarrow{s} \pi_n^r \xleftarrow{t} [n]$.
- Fork : $[n]\,|\,[n] \xrightarrow{[l\circ s,\,r\circ s]} \pi_n \xleftarrow{l\circ t} [n]$.

Définition 5.2.4. On appelle ces cospans les *graines* de notre terrain de jeu.

Nous avons enfin nos cospans de base, qui vont servir de générateurs pour construire nos parties.

5.2.2.2 Coups et parties

Jusqu'ici nos cospans ne représentent que des coups isolés, c'est-à-dire partant d'une position composée seulement des joueurs prenant part au coup. Nous allons maintenant représenter l'inclusion de ces coups isolés dans un contexte plus grand.

L'idée est que pour obtenir un coup on colle une graine à une position, cette dernière représentant la partie passive.

Exemple 5.2.6. La graine de réception $(3\,,1)$ *(9)*, formellement le cospan $[3] \xrightarrow{s} \iota_{3,1} \xleftarrow{t} [3]$, comprend trois canaux de communication. On peut par exemple la recoller à la position constituée d'un joueur binaire, dans laquelle deux de ces trois canaux sont identifiés. Pour cela, on considère le morphisme h défini par

$$3\cdot\star \cong \star + (\star+\star) \xrightarrow{\star+[id,\,id]} \star+\star \xrightarrow{[s_1,\,s_2]} [2]\,.$$

Ce morphisme part de trois canaux, identifie les deux derniers et plonge le résultat dans la position $[2]$ constituée d'un joueur binaire. D'un autre côté, on a l'inclusion $k : 3\cdot\star \hookrightarrow [3]$. Le coup qu'on cherche à construire doit être un cospan ; on peut obtenir ses trois objets en poussant ceux de la graine de départ le long de h. Ainsi, les deux positions obtenues seront le pushout ci-dessous à gauche et l'objet du milieu, le coup, sera le pushout de droite :

$$\begin{tikzcd} 3\cdot\star \arrow[r, "k"] \arrow[d, "h"'] & {[3]} \arrow[d] \\ {[2]} \arrow[r] & X \end{tikzcd} \qquad \begin{tikzcd} 3\cdot\star \arrow[r, "k"] \arrow[d, "h"'] & {[3]} \arrow[r, "s"] & \iota_{3,1} \arrow[d] \\ {[2]} \arrow[rr] & & M\,. \end{tikzcd}$$

En fait, les deux jambes du cospan découlent de la propriété universelle du pushout de gauche, puisqu'elles sont respectivement données par les deux morphismes en pointillés dans

$$\begin{tikzcd} 3\cdot\bigstar \arrow[r, "k"] \arrow[d, "h"'] & {[3]} \arrow[r, "s"] \arrow[d] & \iota_{3,1} \arrow[d] \\ {[2]} \arrow[r] \arrow[rr, bend right] & X \arrow[r, dashed] & M \end{tikzcd} \qquad \text{et} \qquad \begin{tikzcd} 3\cdot\bigstar \arrow[r, "k"] \arrow[d, "h"'] & {[3]} \arrow[r, "t"] \arrow[d] & \iota_{3,1} \arrow[d] \\ {[2]} \arrow[r] \arrow[rr, bend right] & X \arrow[r, dashed] & M. \end{tikzcd}$$

Comme $s \circ k = t \circ k$, on peut définir le coup obtenu comme étant le cospan de droite ci-dessous :

$$\begin{tikzcd} & {[3]} \arrow[rr] \arrow[d, "s"] & & X \arrow[d] \\ & \iota_{3,1} \arrow[rr] & & M \\ 3\cdot\bigstar \arrow[ur, "k"] \arrow[ur] \arrow[rr, "h"] \arrow[dr, "k"'] & & {[2]} \arrow[ur] \arrow[dr] & \\ & {[3]} \arrow[rr] \arrow[uu, "t"] & & X. \arrow[uu] \end{tikzcd}$$

Généralisons maintenant cette construction : chaque graine $Y_0 \xrightarrow{s} M_0 \xleftarrow{t} X_0$ (définition 5.2.4) donne lieu à un pullback

$$\begin{tikzcd} I \arrow[r, "k_s"] \arrow[d, "k_t"'] & Y_0 \arrow[d, "s"] \\ X_0 \arrow[r, "t"'] & M_0. \end{tikzcd}$$

Dans l'exemple précédent, on avait $X_0 = Y_0$ et $k_s = k_t$, mais cela n'est pas toujours le cas, comme le montre la graine ν_n de création de canal.

Définition 5.2.5. Un *coup* est un cospan obtenu en poussant le pullback associé à une graine le long d'un monomorphisme $I \longrightarrow Z$, pour toute position Z, comme ci-dessous :

$$
\begin{tikzcd}
 & Y_0 \arrow[rr] \arrow[dd, "s"] & & X \arrow[dd] \\
 & & & \\
 & M_0 \arrow[rr] & & M \\
I \arrow[uur, "k_s"] \arrow[uur] \arrow[rr, "h"] \arrow[dr, "k_t"'] & & Z \arrow[uur] \arrow[dr] & \\
 & X_0 \arrow[uu, "t"] \arrow[rr] & & X.
\end{tikzcd}
$$

On peut enfin définir la pseudo catégorie double qui nous intéresse. On rappelle la catégorie $\mathbb{C}$ définie en 5.2.3.

Définition 5.2.6. Soit $\mathbb{D}$ la sous-pseudo catégorie double de $\mathrm{Cospan}(\widehat{\mathbb{C}})$ obtenue en restreignant

- les objets aux positions,
- les morphismes horizontaux aux morphismes injectifs,
- les morphismes verticaux aux composés de coups (à isomorphisme près)
- et les cellules aux triplets de morphismes

$$
\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "s"'] & A' \arrow[d, "s'"] \\
B \arrow[r, "g"] & B' \\
C \arrow[u, "t"] \arrow[r, "h"'] & C'. \arrow[u, "t'"']
\end{tikzcd}
$$

avec g injectif (f et h l'étant par défaut en tant que morphismes horizontaux).

Remarque 5.2.3. Les préfaisceaux apparaissant dans $\mathbb{D}$ sont tous finis, dans le sens fort où leur catégorie d'éléments l'est.

5.3 Stratégies

5.3.1 Stratégies naïves

Etant donnée une pseudo catégorie double avec les intuitions de la partie 5.2.1, on peut imiter les *modèles de préfaisceaux* en théorie de la concurrence [11] pour donner une notion naïve de stratégie.

Pour rappel, les modèles de préfaisceaux consistent à partir d'une catégorie $\mathbb{T}$ de *traces*, dont les objets sont intuitivement des traces d'exécution et dont les morphismes

sont donnés par l'ordre préfixe, et à considérer comme notion de processus, en gros, les préfaisceaux sur $\mathbb{T}$.

Exemple 5.3.7. L'exemple tarte à la crème est alimentaire : on ressort la machine à café de Milner. On cherche à représenter les deux machines à café suivantes :

$$\begin{tikzcd}
 & \bigstar \arrow[dl, "a"'] \arrow[dr, "a"] & \\
x \arrow[d, "b"'] & & x' \arrow[d, "c"] \\
y \arrow[d, "d"'] & & y' \arrow[d, "e"] \\
z & & z'
\end{tikzcd}
\qquad
\begin{tikzcd}
\bigstar \arrow[d, "a"'] & \\
x \arrow[dl, "b"'] \arrow[dr, "c"] & \\
y \arrow[d, "d"'] & y' \arrow[d, "e"] \\
z & z',
\end{tikzcd}$$

avec la même interprétation intuitive des actions que dans l'exemple *2.4.4*.

Bien sûr, après avoir mis un euro dans la première machine à café, le client peut se trouver dans l'impossibilité de commander un thé. La première machine, disons M_1 ne modélise donc pas une machine à café raisonnable. La seconde, M_2, au contraire, reçoit toujours la requête du client.

Les deux machines acceptent néanmoins le même ensemble de traces, $\{\varepsilon, a, a\,b, a\,c, a\,b\,d, a\,c\,e\}$, ce que Milner utilise pour motiver la recherche d'une équivalence plus fine.

Nous nous intéressons ici à la représentation de ces deux machines par des préfaisceaux sur l'ensemble ordonné $\mathbb{T}$ (vu comme une catégorie) des mots sur l'ensemble des actions (ici disons un ensemble dénombrable fixé), munis de l'ordre préfixe.

L'idée est que le préfaisceau F_M représentant la machine M enverra toute trace $T \in \mathbb{T}$ sur l'ensemble des états atteignables à partir de l'état initial en suivant T.

De manière analogue à l'exemple *2.4.4*, on obtient pour F_{M_1} :

- $F_{M_1}(\varepsilon) = \{\bigstar\}$,
- $F_{M_1}(a) = \{x, x'\}$,
- $F_{M_1}(ab) = \{y\}$,
- $F_{M_1}(ac) = \{y'\}$,
- $F_{M_1}(abd) = \{z\}$,
- $F_{M_1}(ace) = \{z'\}$,
- $F_{M_1}(T) = \emptyset$ ailleurs,

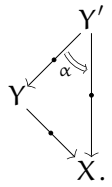
avec, par exemple, $F_{M_1}(ac \leqslant ace)(z') = y'$, etc.

Le préfaisceau F_{M_2} ressemble à F_{M_1}, sauf que $F_{M_2}(a)$ est le singleton $\{x\}$ et qu'on modifie les actions des morphismes $a \leqslant T$ pour envoyer tout le monde sur x.

Evidemment, les modèles de préfaisceaux sont plus généraux que l'exemple précédent, au sens où ils ont été définis dans un cadre abstrait permettant la définition de la bisimilar-

ité. Ce cadre commun englobe à la fois les catégories de traces comme dans l'exemple, les structures d'évènements, les « pomsets », etc.

Dans la pseudo catégorie double $\mathbb{D}$ fixée en début de partie, on peut fixer une position initiale X, pour laquelle on a une structure naturellement analogue à $\mathbb{T}$: il s'agit de la catégorie ayant pour objets les parties de position initiale X — on dira les parties *sur* X — et pour morphismes $u \to u'$ les paires (w, α) comme dans



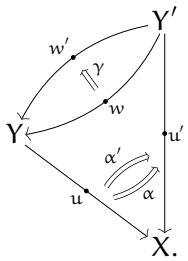
La composition est évidente et l'identité sur u est donnée par $\mathrm{id}_{u:Y \to X} = (\mathrm{id}_Y, \rho_u)$, où $\rho_u : u \bullet \mathrm{id}_Y \to u$ est l'isomorphisme structurel de la composition verticale.

On obtient bien une catégorie, à ceci près que la composition n'est pas strictement associative, ni les identités strictement neutres. En effet, en composant

$$u \xrightarrow{(w,\alpha)} u' \xrightarrow{(w',\alpha')} u'' \xrightarrow{(w'',\alpha'')} u'''$$

des deux manières différentes possibles, on obtient deux paires dont les premières composantes, $w \bullet (w' \bullet w'')$ et $(w \bullet w') \bullet w''$, diffèrent déjà *a priori* puisqu'on est dans une pseudo catégorie double.

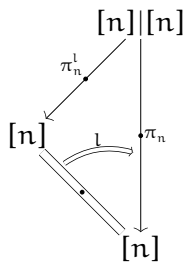
On considère donc comme équivalents deux morphismes (w, α) et (w', α') dès qu'il existe un morphisme spécial $\gamma : w \to w'$ dans $\mathbb{D}_H$ tel que $\alpha' \circ (u \bullet \gamma) = \alpha$. En images :



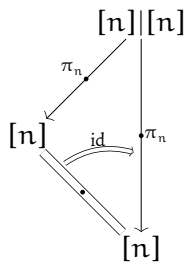
Plus formellement, on quotiente bien sûr par la relation d'équivalence engendrée.

Proposition 5.3.2. On obtient une catégorie $\mathbb{P}(X)$ des *parties sur* X.

Remarque 5.3.4. Rétrospectivement, le quotient qu'on fait pour obtenir une catégorie fait sens. En effet, il assure que les morphismes $u \to u'$ décrivent la manière de plonger u dans u', indépendamment de w. Par exemple, on peut construire deux morphismes



et



décrivant clairement la même manière de plonger l'identité verticale dans π_n. Il semble donc juste d'identifier (π_n^l, l) et (π_n, id). Symétriquement et par transitivité, on identifie aussi (π_n^l, l) à (π_n^r, r).

Les préfaisceaux sur $\mathbb{P}(X)$ satisfont aux contraintes habituelles de clôture par préfixe des stratégies : tout état au-dessus d'une partie u' admet pour tout morphisme $(w, \alpha) : u \to u'$ une restriction à u donnée par l'action de (w, α). Les préfaisceaux satisfont de plus une condition d'invariance par isomorphisme de partie, dans les cas où cela fait sens.

On pose donc:

Définition 5.3.7. La catégorie des *stratégies naïves* sur X est la catégorie des préfaisceaux sur $\mathbb{P}(X)$.

Exemple 5.3.8. Dans le cas de CCS, décrit en partie précédente, toute création de canal de communication impose le choix d'un représentant. Tous les choix menant néanmoins à des parties isomorphes, tout préfaisceau sur $\mathbb{P}(X)$ (pour tout X) devra les envoyer sur des ensembles isomorphes.

Comme elles forment une catégorie de préfaisceaux, on a :

Proposition 5.3.3. Les stratégies naïves forment sur chaque position X une catégorie cartésienne fermée, complète et cocomplète (modulo la remarque *5.3.5* ci-dessous).

On peut organiser les stratégies naïves (vers set) en un pseudo 2-foncteur $\mathbb{D}_v^{\mathrm{coop}} \to \mathrm{Cat}$. En effet, on a d'abord un pseudo 2-foncteur $\mathbb{D}_v \to \mathrm{Cat}$ envoyant chaque X sur $\mathbb{P}(X)$ et chaque $u : Y \nrightarrow X$ sur le foncteur

$$\begin{aligned} \mathbb{P}(Y) &\to \mathbb{P}(X) \\ u' &\mapsto u \bullet u'. \end{aligned}$$

Par dualité, on obtient un foncteur $\mathbb{D}_v^{\mathrm{coop}} \to \mathrm{Bat}^{\mathrm{coop}}$. D'autre part on a le foncteur de dualisation $\mathrm{op} : \mathrm{Cat} \to \mathrm{Cat}^{\mathrm{co}}$, dont le dual est $\mathrm{op}^{\mathrm{coop}} : \mathrm{Cat}^{\mathrm{coop}} \to \mathrm{Cat}^{\mathrm{op}}$. Enfin, on a le pseudo 2-foncteur représentable $\mathbf{y}_{\mathrm{set}}$ sur set. Le composé

$$\mathbb{D}_v^{\mathrm{coop}} \to \mathrm{Cat}^{\mathrm{coop}} \to \mathrm{Cat}^{\mathrm{op}} \to \mathrm{Cat}$$

envoie bien tout $X \in \mathbb{D}_v$ sur $\mathrm{Cat}(\mathbb{P}(X)^{\mathrm{op}}, \mathrm{set}) \subseteq \widehat{\mathbb{P}(X)}$. On a prouvé :

Proposition 5.3.4. L'assignation $X \mapsto \mathrm{Cat}(\mathbb{P}(X)^{\mathrm{op}}, \mathrm{set})$ s'étend en un pseudo 2-foncteur $\mathbb{D}_v^{\mathrm{coop}} \to \mathrm{Cat}$.

Remarque 5.3.5.

On a ici un problème de taille puisque $\widehat{\mathbb{P}(X)}$ n'est pas une petite catégorie. Les articles sur lesquels s'appuie cette introduction sont plus rigoureux en la matière et considèrent en fait directement les préfaisceaux de « petits ensembles », dont on trouve plusieurs variantes, la plus souple étant sans doute les parties finies de $\mathbb{N}$. Dans la suite, on ignore ce problème

et $\widehat{\mathbb{P}(X)}$ désigne en fait $\mathrm{Cat}(\mathbb{P}(-)^{\mathrm{op}}, \mathrm{set})$.

Pour le pseudo 2-foncteur $\widehat{\mathbb{P}(-)}$, l'action d'une partie $u : Y \rightarrowtail X$ envoie $S \in \widehat{\mathbb{P}(X)}$ sur X sur le préfaisceau défini par $(S \cdot u)(u') = S(u \bullet u')$.

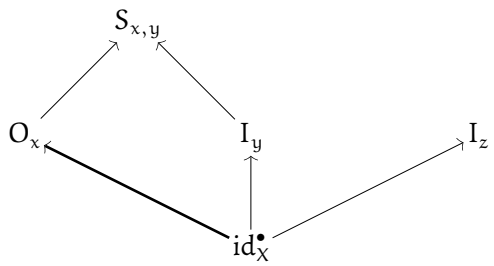
Si les préfaisceaux jouissent d'une structure raisonnable, ils n'assurent en aucun cas les contraintes d'indépendance entre les joueurs qu'on aimerait imposer au vu de la sémantique opérationnelle de CCS.

Exemple 5.3.9.

On peut définir un préfaisceau définissant le comportement suivant. On considère une position constituée de trois joueurs x, y et z partageant un canal a. Le joueur x accepte d'émettre sur a et y et z acceptent tous les deux de recevoir. La synchronisation entre x et y est acceptée, mais pas celle entre x et z.

Définir un tel préfaisceau à la main n'est *a priori* pas complètement évident, puisqu'il faut gérer les isomorphismes entre parties. En revanche, la combinatoire catégorique nous fournit un guide pour le faire, les extensions de Kan. Ces dernières nous mènent à une première définition, qu'on peut ensuite simplifier. Commençons par donner des noms : soit $X = ([1]|[1]|[1])$ la position complète, dont on décrète par convention que les joueurs sont dans l'ordre x, y et z. Soient ensuite O_x la partie où x émet sur a, I_y celle où y reçoit et I_z celle où z reçoit. Soient enfin $S_{x,y}$ et $S_{x,z}$ les deux synchronisations.

L'idée est de considérer la sous-catégorie pleine $\mathbb{F}$ de $\mathbb{P}(X)$ contenant $\mathrm{id}_X^{\bullet}$, O_x, I_y, I_z et $S_{x,y}$ — mais pas $S_{x,z}$! C'est en fait l'ordre partiel engendré par



vu comme une catégorie. Restreinte à cette sous-catégorie, notre préfaisceau est très simple à définir puisque c'est le préfaisceaux terminal, constant égal au singleton 1, qu'on appellera T dans la suite de l'exemple. On construit ensuite l'extension à gauche de $T : \mathbb{F}^{op} \to \mathrm{Set}$ le long du plongement $\mathbb{F}^{op} \hookrightarrow \mathbb{P}(X)^{op}$. Par l'expression des extensions à gauche comme des cofins (proposition 2.4.23), on obtient le préfaisceau

$$S(u) = \int^{u' \in \mathbb{F}} \mathbb{P}(X)(u, u') \cdot T(u') \cong \int^{u' \in \mathbb{F}} \mathbb{P}(X)(u, u').$$

En fait, $\mathbb{P}(X)$ est un préordre et $\mathbb{F}$ est connexe, donc on obtient

$$S(u) = \begin{pmatrix} 1 & \text{s'il existe un morphisme } u \to u' \text{ avec } u' \in \mathbb{F} \\ \emptyset & \text{sinon.} \end{pmatrix}$$

On obtient clairement comme souhaité que S accepte toutes les parties de $\mathbb{F}$, mais pas $S_{x,z}$. Tout se passe donc comme si x et y se mettaient d'accord pour écarter z du jeu. Un tel comportement n'est pas souhaitable dans un modèle de CCS, où on s'attend à ce que les joueurs n'aient pas d'autre manière de communiquer entre eux que les canaux eux-mêmes. Par exemple, x ne devrait avoir aucun moyen de détecter si c'est y ou z qui reçoit son message.

Pour remédier à ce problème, on définit dans la partie suivante une notion de stratégie *innocente*.

5.3.2 L'idée des stratégies innocentes

L'intuition des stratégies innocentes est que pour chaque partie u sur X, un état acceptant u est constitué d'une famille compatible d'états acceptant les *vues* de chaque joueur en présence, où les vues sont des parties distinguées représentant les informations auxquelles un joueur peut avoir accès.

Définition 5.3.8. En CCS, une vue est une partie dont les domaine et codomaine sont tous deux des positions représentables, c'est-à-dire de la forme $[n]$. Soit $\mathbb{V}$ la sous-pseudo catégorie double de $\mathbb{D}$ obtenue en restreignant les morphismes verticaux aux vues.

En pratique, en plus des parties identités sur les $[n]$, ceci inclut toutes les graines hormis les π_n et les $\tau_{n,i,m,j}$. Les vues étant stables par composition verticale, on obtient bien une sous-pseudo catégorie double.

Définir formellement l'ensemble des vues des joueurs en présence dans une partie s'avère quelque peu subtil, de la même manière que faire de $\mathbb{P}(X)$ une catégorie le fut. En effet, commençons par constater qu'une vue peut avoir plusieurs occurrences dans une partie.

Notation 5.3.2. Etendons d'abord de la manière naturelle la notation *(5.2.1)* aux parties, comme dans :

$$\left(X_{a_1,\dots,a_p|b_1,\dots,b_p}Y\right) \xrightarrow{t_{X_{a_1,\dots,a_p|b_1,\dots,b_p}t_Y}} \left(u_{a_1,\dots,a_p|b_1,\dots,b_p}v\right) \xleftarrow{s_{X_{a_1,\dots,a_p|b_1,\dots,b_p}s_Y}} \left(X'_{a_1,\dots,a_p|b_1,\dots,b_p}Y'\right).$$

Exemple 5.3.10. Dans la partie $\pi_n|\pi_n$, on trouve deux occurrences de la vue π^l_n, une pour le premier joueur, l'autre pour le second.

Une définition naïve consiste à considérer qu'une vue de $u \in \mathbb{P}(X)$ est un morphisme $v \to u$ dans $\mathbb{D}_H$. Une stratégie innocente S pour cette définition pourrait alors être donnée en associant un ensemble à chaque vue $v : [n'] \nrightarrow [n]$ liée à un joueur de X. Formellement, on arrive rapidement à considérer les préfaisceaux sur la catégorie $\mathbb{V}_H/X$ définie comme la catégorie comma :

$$\begin{tikzcd}
\mathbb{V}_H/X \arrow[r] \arrow[d] \arrow[dr, phantom, "\lambda"] & 1 \arrow[d, "\ulcorner X \urcorner"] \\
\mathbb{V}_H \arrow[r, "\mathrm{cod}"'] & \mathbb{D}_h
\end{tikzcd}$$

dont les objets sont paires (v, x) d'une vue v et d'un morphisme $x : \mathrm{cod}(v) \to X$ dans $\mathbb{D}_h$. D'un préfaisceau S sur $\mathbb{V}_H/X$ on en obtient un sur $\mathbb{P}(X)$ en posant

$$S'(u) = \prod_{\alpha : v \to u} S(v, \mathrm{cod}(\alpha))$$

(on devrait en fait se restreindre à un squelette $\mathbb{V}^s_H$ de $\mathbb{V}_H$; prendre une fin ici ne changerait alors rien parce que $\mathbb{V}^s_H/u$ est en fait discrète).

Un préfaisceau sur $\mathbb{P}(X)$ serait alors innocent s'il est dans l'image essentielle de cette fonction $S \mapsto S'$. En particulier, pour qu'un tel préfaisceau S sur $\mathbb{P}(X)$ soit innocent, il doit satisfaire la condition suivante :

Condition 5.3.1. Pour toute partie u sur X, si pour toute vue v et tout morphisme $\alpha : v \to u$ il existe une partie u'_α et un morphisme $\beta_\alpha : v \to u'_\alpha$ tel que $S(u'_\alpha) \neq \emptyset$, alors $S(u) \neq \emptyset$.

Cependant, cela ne suffit pas pour imposer la compatibilité des états de deux joueurs admettant une part de passé commun. Le critère obtenu est trop restrictif, comme le montre l'exemple suivant.

Exemple 5.3.11.
Considérons la partie $u = (\nu_0 \bullet \pi_1 \bullet (\tau_{1,1,1,1})) : ([1]|[1]) \nrightarrow [0]$, où le joueur x de départ crée un canal a puis se sépare en x_1 et x_2, qui se synchronisent ensuite sur a — x_1 étant l'émetteur. On va construire un préfaisceau qui refuse cette partie mais accepte suffisamment de vues pour ne pas être innocent dans le sens naïf proposé. Pour définir ce préfaisceau sur $\mathbb{P}[0]$, on commence par en définir deux sur $\mathbb{P}([1]|[1])$. Soit S_1 le plus petit préfaisceau dans lequel x_1 accepte d'émettre sur a, mais x_2 refuse de recevoir. Comme dans l'exemple *5.3.9*, on peut formaliser cela en posant

$$S_1(u) = \begin{pmatrix} 1 & \text{s'il existe un morphisme } u \to (o_{1,1}|[1]) \\ \emptyset & \text{sinon,} \end{pmatrix}$$

(voir la notation *(5.3.2)*). Soit maintenant S_2 celui dans lequel x_2 accepte de recevoir mais x_1 refuse d'émettre, défini de manière analogue.

On utilise maintenant une construction qui, étant donnée une partie $u : Y \nrightarrow X$ et une stratégie S sur Y, construit une stratégie notée $u \bullet S$ sur X, telle que

$$(u \bullet S)(\mathrm{id}^\bullet_X) = 1$$

et, pour tout $u' : Z \nrightarrow Y$,

$$(u \bullet S)(u \bullet u') = S(u').$$

Cette construction est définie formellement en partie 5.3.3, mais l'intuition devrait être claire et le seul intérêt de cette partie est sans doute d'être un exemple combinatoire de colimite pondérée, une notion de théorie des catégories enrichies.

On définit grâce à cette construction $S'_i = \pi_1 \bullet S_i$ sur $[1]$, pour $i \in 2$, puis $S = \nu_0 \bullet (S'_1 + S'_2)$ sur $[0]$. Intuitivement, ce préfaisceau S correspond au processus CCS $\nu a\,.\,((\overline{a}|0) + (0|a))$, qui est parfaitement légitime. Sauf qu'il ne satisfait pas la condition naïve d'innocence donnée ci-dessus. En effet, on a par construction $S(\mathrm{id}^\bullet_{[0]}) = 1$, $S(\nu_0) = 2 = 1+1 = S'_1(\mathrm{id}^\bullet_{[1]}) + S'_2(\mathrm{id}^\bullet_{[1]})$ et pour toute partie $u : X \nrightarrow ([1]|[1])$:

$$S(\nu_0 \bullet \pi_1 \bullet u) = S'_1(\pi_1 \bullet u) + S'_2(\pi_1 \bullet u) = S_1(u) + S_2(u).$$

En particulier, on a

$$S(\nu_0 \bullet \pi_1 \bullet (o_{1,1}|[1])) = S_1(o_{1,1}|[1]) + S_2(o_{1,1}|[1]) = 1 + \emptyset = 1$$

et

$$S(\nu_0 \bullet \pi_1 \bullet ([1]|\iota_{1,1})) = S_1([1]|o_{1,1}) + S_2([1]|o_{1,1}) = \emptyset + 1 = 1.$$

On a enfin

$$S(\nu_0 \bullet \pi_1 \bullet \tau_{1,1,1,1}) = \emptyset,$$

ce qui contredit la condition *(11)*.

5.3.3 Interlude : un exemple d'application des colimites pondérées

Revenons à présent à la notation $u \bullet S$, pour laquelle la combinatoire catégorique nous fournit à nouveau une définition rigoureuse, que le lecteur pressé pourra ignorer sans dommage, tant l'intuition est claire. On n'inclut cette définition qu'afin d'illustrer le niveau de rigueur gagné sans trop d'efforts grâce aux catégories. On fixe pour commencer une partie $u : Y \nrightarrow X$ et un préfaisceau $S : \mathbb{P}(Y)^{op} \to \mathrm{Set}$ arbitraires.

Premièrement, le moyen catégorique d'ajouter formellement un objet initial à une catégorie $\mathscr{C}$ consiste à prendre la *colimite relâchée* (*lax colimit* en anglais) [44] de l'unique foncteur $\mathscr{C} \to 1$ dans la Cat-catégorie Cat, c'est-à-dire le diagramme universel de la forme

$$\begin{tikzcd} \mathscr{C} \arrow[dd, "!"'] \arrow[dr, "p"] & \\ & \mathscr{D} \arrow[dl, phantom, "\Downarrow\alpha"] \\ 1. \arrow[ur, "q"'] & \end{tikzcd}$$

Ici, l'universalité signifie que pour tout autre diagramme $(\mathscr{D}', p', q', \alpha')$ de la même forme, on trouvera un unique foncteur $m : \mathscr{D} \to \mathscr{D}'$ tel que $m \circ p = p'$, $m \circ q = q'$ et $m \circ \alpha = \alpha'$. Dans notre exemple, notons $\mathbb{P}(Y)_\perp$ la colimite relâchée de $\mathbb{P}(Y) \to 1$. On procède en plusieurs étapes :

- on utilise la propriété universelle de $\mathbb{P}(Y)_\perp$ pour définir une extension $S_\perp$ de S à cette dernière ;
- par la même propriété universelle, on définit un foncteur $\mathbb{P}(Y)_\perp \to \mathbb{P}(X)$ étendant le foncteur canonique de composition verticale avec u, $(u \bullet -) : \mathbb{P}(Y) \to \mathbb{P}(X)$;
- on définit $u \bullet S$ comme l'extension à gauche de $S_\perp$ le long de (l'opposé de) $\mathbb{P}(Y)_\perp \to \mathbb{P}(X)$.

Pour le premier point, bien sûr, en prenant les catégories opposées on obtient un diagramme universel de la même forme avec α dans l'autre sens (sa version *co*) : on a ajouté un objet terminal à $\mathbb{P}(Y)^{op}$. Pour tout $u' : Z \nrightarrow Y$, on a un morphisme canonique $!_{S(u')}$: $S(u') \to 1$ qui donne une transformation naturelle $!_{S(-)} : S \to (\ulcorner 1 \urcorner \circ !)$ dans $\mathrm{Cat}(\mathbb{P}(Y)^{op}$,

Set). Par propriété universelle de $\mathbb{P}(Y)_\perp$ on obtient l'extension $S_\perp : \mathbb{P}(Y)_\perp^{op} \to \mathrm{Set}$ de S souhaitée, qui fait commuter le diagramme

$$\begin{tikzcd}
\mathbb{P}(Y)^{op} \arrow[dr, "p^{op}"] \arrow[dd, "!"'] \arrow[drr, bend left, "S"] & & \\
& \mathbb{P}(Y)_\perp^{op} \arrow[r, dashed, "S_\perp"] \arrow[dr, phantom, "\Downarrow !_{S(-)}"] & \mathrm{Set} \\
1 \arrow[ur, "q^{op}"'] \arrow[urr, bend right, "\ulcorner 1 \urcorner"'] \arrow[ur, phantom, shift left=3, "\Downarrow \alpha"] & &
\end{tikzcd}$$

(parce qu'on considère les catégories opposées). En particulier, on a bien $S_\perp(q(\star)) = 1$ et $S_\perp(p(u')) = S(u')$ pour tout $u' \in \mathbb{P}(Y)$.

Pour le second point, pour tout $u' : Z \rightarrowtail Y$, on a un morphisme canonique $\lambda_{u'} : \mathrm{id}_X^\bullet \to u \bullet u'$ qui donne une transformation naturelle $\lambda : \mathrm{id}_X^\bullet \to (u \bullet -)$ dans $\mathrm{Cat}(\mathbb{P}(Y), \mathbb{P}(X))$, donc par propriété universelle de $\mathbb{P}(Y)_\perp$ un foncteur m faisant commuter

$$\begin{tikzcd}
\mathbb{P}(Y) \arrow[dr, "p"] \arrow[dd, "!"'] \arrow[drr, bend left, "u\bullet -"] & & \\
& \mathbb{P}(Y)_\perp \arrow[r, dashed, "m"] \arrow[dr, phantom, "\Downarrow \lambda"] & \mathbb{P}(X). \\
1 \arrow[ur, "q"'] \arrow[urr, bend right, "\mathrm{id}_X^\bullet"'] \arrow[ur, phantom, shift left=3, "\Downarrow \alpha"] & &
\end{tikzcd}$$

On note au passage qu'on a $m(p(u')) = u \bullet u'$ pour tout u'.

Enfin, on définit $u \bullet S$ comme l'extension à gauche de $S_\perp$ le long de m^{op}. De plus, on peut montrer que m est plein et fidèle, essentiellement parce que les parties de CCS n'ont pas d'endomorphismes non triviaux. Donc on peut choisir $u \bullet S$ tel que $(u \bullet S) \circ m^{op} = S_\perp$. Mais cela entraîne grâce aux résultats précédents que l'extérieur du diagramme suivant commute aussi :

$$\begin{tikzcd}
\mathbb{P}(Y)^{op} \arrow[rr, "(u\bullet -)^{op}"] \arrow[dr, "p^{op}"'] \arrow[ddrr, bend right, "S"'] & & \mathbb{P}(X)^{op} \arrow[dd, "u\bullet S"] \\
& \mathbb{P}(Y)_\perp^{op} \arrow[ur, "m^{op}"'] \arrow[dr, "S_\perp"] & \\
& & \mathrm{Set}.
\end{tikzcd}$$

On a donc en particulier

$$(u \bullet S)(\mathrm{id}_X^\bullet) = (u \bullet S)(m(q(\star))) = S_\perp(q(\star)) = 1$$

et

$$(u \bullet S)(u \bullet u') = (u \bullet S)(m(p(u'))) = S_{\perp}(p(u')) = S(u'),$$

comme souhaité.

5.3.4 Comportements innocents et lien avec stratégies naïves

L'exemple *5.3.11*, s'il pointe un défaut dans la condition *(11)*, indique aussi une solution intuitive. Etant données deux morphismes de vues $v_1 \xrightarrow{\alpha_1} u \xleftarrow{\alpha_2} v_2$, on ne souhaite imposer l'existence d'un état dans $S'(u)$ à partir de $x_1 \in S(v_1)$ et $x_2 \in S(v_2)$ que si pour tout préfixe commun v de v_1 et v_2, les « restrictions » de x_1 et x_2 à v coïncident.

Pour réaliser cette intuition, on commence par étendre $\mathbb{V}_H/X$ pour prendre en compte l'ordre préfixe — exactement comme on l'a fait en définissant $\mathbb{P}(X)$. Pour cela, on tente de former une catégorie $\mathbb{V}_X$

- dont les objets sont les paires (v, x) avec v une vue et $x : \mathrm{cod}(v) \to X$ et
- dont les morphismes $(v, x) \to (v', x')$, avec $v : [n'_v] \rightarrowtail [n_v]$ et $v' : [n'_{v'}] \rightarrowtail [n_{v'}]$ sont les vues $w : [n_w] \rightarrowtail [n'_v]$ munies d'un morphisme $\alpha : (v \bullet w) \to v'$ dans $\mathbb{D}_H$ tel que $x' \circ \mathrm{cod}(\alpha) = x$.

En images, un morphisme correspond à un diagramme commutatif

$$\begin{tikzcd}
{[n_w]} \arrow[rr] \arrow[d, tail, "w"'] \arrow[dr, phantom, "\Nearrow\alpha"] & & {[n'_{v'}]} \arrow[dd, tail, "v'"] \\
{[n'_v]} \arrow[d, tail, "v"'] & & \\
{[n_v]} \arrow[rr] \arrow[dr, "x"'] & & {[n_{v'}]} \arrow[dl, "x'"] \\
& X. &
\end{tikzcd}$$

Sauf qu'en fait la sous-catégorie pleine de $\mathbb{D}_h$ formée par les positions représentables s'avère discrète. Donc les deux morphismes horizontaux ci-dessus sont forcément des identités et on peut réduire la notion de morphisme à celle de $\mathbb{P}(X)$. La catégorie $\mathbb{V}_X$ contient donc pour chaque $x : [n] \to X$ une copie de la sous-catégorie pleine de $\mathbb{P}[n]$ restreinte aux vues, disons $\mathbb{V}[n]$. On pose donc :

Définition 5.3.9. Soit $\mathbb{V}_X = \sum_{x:[n]\to X} \mathbb{V}[n]$. La catégorie des *comportements innocents* sur X est celle des préfaisceaux sur $\mathbb{V}_X$.

Se pose à présent la question du rapport entre un comportement innocent en ce sens et les préfaisceaux sur $\mathbb{P}(X)$.

Dans l'article sur CCS [31], on introduit une catégorie intermédiaire $\mathbb{E}_X$ incluant à la fois les vues et les parties, dont les morphismes sont proches de ceux de $\mathbb{V}_X$. On construit

ainsi un cospan $\mathbb{V}_X \to \mathbb{E}_X \leftarrow \mathbb{P}(X)$ de foncteurs et le passage d'un comportement innocent au préfaisceau correspondant sur $\mathbb{P}(X)$ se fait par extension de Kan à droite le long du premier foncteur, puis par restriction le long du second.

Nous utilisons ici une autre approche, qui requiert des hypothèses légèrement différentes. Pour cette construction en particulier, l'article utilise

- le fait que le foncteur codomaine $\mathbb{D}_H \to \mathbb{D}_h$ soit une fibration et
- un certain axiome dit de *décomposition à droite*.

Par contraste, l'approche exposée ici repose

- sur une version limitée de l'axiome de fibration, ne demandant l'existence de relèvements cartésiens que le long de *joueurs*, c'est-à-dire de morphismes horizontaux de la forme $[n] \to X$, et
- un renforcement de l'axiome de décomposition à droite, auquel on ajoute une condition de minimalité faible (voir ci-dessous).

L'intérêt de cette approche reste bien sûr à démontrer sur de nouveaux exemples.

L'idée en est de définir la notion d'*occurrence* d'une vue sur X dans une partie sur X, puis d'en tirer un foncteur $\mathrm{EInj} : \mathbb{P}(X) \to \widehat{\mathbb{V}_X}$, autrement dit un profoncteur $\mathrm{EInj} : \mathbb{P}(X) \nrightarrow \mathbb{V}_X$. On peut ensuite appliquer la gymnastique catégorique archi-rebattue du foncteur singulier, revue en partie 2.4.5. Dans notre cas, on obtient

$$\begin{tikzcd}
\mathbb{P}(X) \arrow[rr, "y"] \arrow[dr, "\mathrm{EInj}"'] & & \widehat{\mathbb{P}(X)} \arrow[dl, bend left, "\mathrm{lan}_y \mathrm{EInj}"] \arrow[dl, phantom, "\top"] \\
& \widehat{\mathbb{V}_X}. \arrow[ur, bend left, "\mathrm{Sing}"] & \\
\end{tikzcd}$$

Le foncteur qui nous intéresse ici est bien sûr Sing. Une interprétation de son adjoint à gauche $\mathrm{lan}_y \mathrm{EInj}$ est qu'il calcule un comportement innocent minimal associé à son argument.

Remarque 5.3.6. Le cœur de la différence entre l'approche publiée et celle décrite ici réside dans le fait que le *collage* [10] du profoncteur EInj définit seulement une partie de la catégorie $\mathbb{E}_X$, qui suffit à faire le pont entre comportement innocents et stratégies.

La notion cruciale ici est donc celle d'occurrence d'une vue de $\mathbb{V}_X$ dans une partie $u : Y \rightarrowtail X$. Une première idée consiste à définir une occurrence de (x, v) dans u, pour $v : [n'_v] \rightarrowtail [n_v]$, comme étant une paire (w, α) avec $w : Z \rightarrowtail [n'_v]$ une partie et $\alpha : (v \bullet w) \to u$ dans $\mathbb{D}_H$ tel que $\mathrm{cod}(\alpha) = x$. Cependant, un examen rapide d'exemples nous rappelle le quotient définissant les morphismes de $\mathbb{P}(X)$. On identifie donc deux telles paires (w, α) et (w', α') dès que α se décompose en

$$
\begin{tikzcd}
{[n_w]} \arrow[r, "h"] \arrow[d, "w"'] \arrow[dr, phantom, "\Downarrow\gamma"] & {[n_{w'}]} \arrow[r, "l"] \arrow[dl, "w'"'] \arrow[dr, phantom, "\Downarrow\alpha'"] & Y \arrow[dd, "u"] \\
{[n'_v]} \arrow[d, "v"'] & & \\
{[n_v]} \arrow[rr, "x"'] & & X
\end{tikzcd}
$$

pour un certain γ. On note $[w, \alpha]$ la classe d'équivalence de (w, α) pour cette relation.

Définition 5.3.10. Soit $\mathrm{EInj}(v, u)$ le quotient obtenu, dont on appelle les éléments des *extensions-injections*, ou *extinjections*, en référence au fait qu'on étend v d'abord et qu'on injecte ensuite le résultat dans u.

Notre but est à présent de montrer que l'assignation $((v, x), u) \mapsto \mathrm{EInj}((v, x), u)$ définit un profoncteur $\mathbb{P}(X) \xrightarrow{\mathrm{EInj}} \mathbb{V}_X$. L'action de $\mathbb{V}_X$ est claire : étant donné $(w, \alpha) \in \mathrm{EInj}(V, u)$ et $(w', \gamma) : V' \to V$, on peut poser $(w, \alpha) \cdot (w', \gamma) = (w' \bullet w, \alpha \circ (\gamma \bullet w))$ et on obtient bien :

Proposition 5.3.5. Pour tout $u : Y \nrightarrow X$, les assignations

$$
\begin{array}{rrcl}
\mathrm{EInj}(-, u): & \mathbb{V}_X^{op} & \to & \mathrm{Set} \\
& (v, x) & \mapsto & \mathrm{EInj}((v, x), u) \\
& (w', \gamma) & \mapsto & (w, \alpha) \mapsto (w' \bullet w, \alpha \circ (\gamma \bullet w))
\end{array}
$$

définissent un foncteur.

Démonstration. Facile, grâce en particulier au quotient définissant EInj. □

En revanche, l'action de $\mathbb{P}(X)$ est moins claire.

Dans les articles sur CCS et π, on démontre qu'étant donné $(w', \gamma) : u \to u'$, on peut construire une *restriction* canonique

$$
\begin{tikzcd}
Y'_{|\mathrm{dom}(\alpha)} \arrow[r] \arrow[d, "w'_{|\mathrm{dom}(\alpha)}"'] \arrow[dr, phantom, "\Downarrow\alpha_{w', \mathrm{dom}(\alpha)}"] & Y' \arrow[d, "w'"] \\
{[n'_v]} \arrow[r, "\mathrm{dom}(\alpha)"'] & Y
\end{tikzcd}
$$

de w' le long de $\mathrm{dom}(\alpha)$ et définir $(w', \gamma) \cdot (w, \alpha)$ comme $(w \bullet w'_{|\mathrm{dom}(\alpha)}, \gamma \circ (\alpha \bullet \alpha_{w', \mathrm{dom}(\alpha)}))$, graphiquement :

$$
\begin{tikzcd}
Z_{|k} \arrow[r, "l"] \arrow[d, "w'_{|k}"'] \arrow[dr, phantom, "\Uparrow \alpha_{w',k}"] & Z \arrow[d, "w'"] \arrow[ddd, bend left] \\
Y' \arrow[r, "k"] \arrow[d, "w"'] \arrow[dr, phantom, "\Uparrow \alpha"] & Y \arrow[d, "u"] \arrow[r, Rightarrow, "\gamma"] & u' \\
{[n'_v]} \arrow[d, "v"'] & \\
{[n_v]} \arrow[r, "x"'] & X.
\end{tikzcd}
\tag{12}
$$

La canonicité de cette restriction signifie qu'elle est cartésienne au sens des fibrations de Grothendieck, c'est-à-dire qu'elle fournit un relèvement cartésien de w' le long de k pour le foncteur $\mathrm{cod} : \mathbb{D}_H \to \mathbb{D}_h$.

Proposition 5.3.6. On obtient un profoncteur $\mathrm{EInj} : \mathbb{P}(X) \nrightarrow \mathbb{V}_X$.

Démonstration. On montre tout d'abord que l'action ci-dessus définit pour tout (v, x) un foncteur

$$
\begin{aligned}
\mathrm{EInj}((v, x), -) : \quad \mathbb{P}(X) &\to \mathrm{Set} \\
u &\mapsto \mathrm{EInj}((v, x), u) \\
(w'', \delta) &\mapsto (w, \alpha) \mapsto (w \bullet w''_{|\mathrm{dom}(\alpha)}, \alpha \bullet \alpha_{w'', \mathrm{dom}(\alpha)}).
\end{aligned}
$$

On montre ensuite que les deux actions sont compatibles, au sens où le carré

$$
\begin{tikzcd}
\mathrm{EInj}((v, x), u) \arrow[r, "\mathrm{EInj}((w', \gamma), u)"] \arrow[d, "\mathrm{EInj}((v, x), (w'', \delta))"'] & \mathrm{EInj}((v', x), u) \arrow[d, "\mathrm{EInj}((v', x), (w'', \delta))"] \\
\mathrm{EInj}((v, x), u') \arrow[r, "\mathrm{EInj}((w', \gamma), u')"'] & \mathrm{EInj}((v', x), u')
\end{tikzcd}
$$

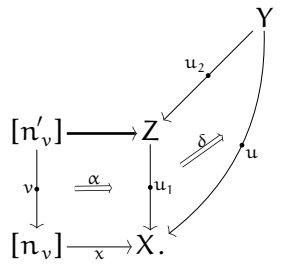
commute. □

Nous enrichissons ici ce point de vue en donnant une définition équivalente de $\mathrm{EInj}(v, u)$, dont l'idée, au lieu d'envoyer une extension de v dans u, est d'envoyer v dans un préfixe de u. Pour cette définition, l'action de u est claire.

Définition 5.3.11. Soit $\mathrm{IExt}((v, x), u)$ l'ensemble des quadruplets $q = (u_1, u_2, \delta, \alpha)$ avec δ un isomorphisme $(u_1 \bullet u_2) \to u$ et $\alpha : v \to u_1$ tel que $\mathrm{cod}(\alpha) = x$, quotienté par la relation d'équivalence identifiant q avec $q' = (u'_1, u'_2, \delta', \alpha')$ dès qu'il existe u' et des isomorphismes $\upsilon : (u_1 \bullet u') \to u'_1$, $\mu : u_2 \to (u' \bullet u'_2)$ et $\xi : \mathrm{id}^{\bullet}_{\mathrm{dom}(v)} \to u'$ tels que $\upsilon \circ (\alpha \bullet$

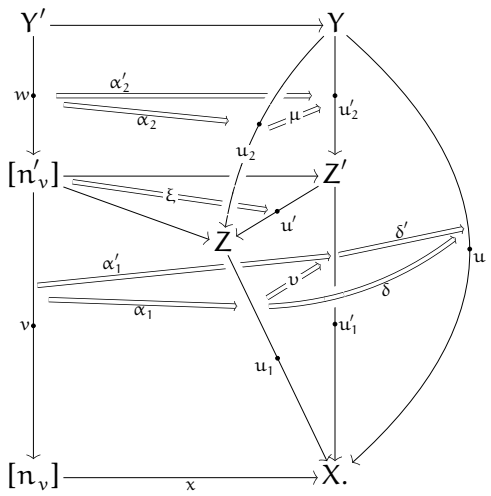
$\xi) = \alpha'$ et $\delta = \delta' \circ (\upsilon \bullet u'_2) \circ (u_1 \bullet \mu)$. On appelle ces classes d'équivalence des *injections-extensions*, ou *injextensions*, en référence au fait qu'on injecte d'abord v dans une partie de u, que l'on étend ensuite.

Graphiquement, un quadruplet q ressemble à



(13)

La minimalité faible signifie ici que pour toute autre décomposition $(u'_1, u'_2, \delta', \alpha'_1, \alpha'_2)$, on peut trouver u' et des isomorphismes $\upsilon : u_1 \bullet u' \to u'_1$, $\mu : u_2 \to u' \bullet u'_2$ et $\xi : \mathrm{id}^{\bullet}_{\mathrm{dom}(v)} \to u'$ tels que $\alpha_1 \bullet \xi = \alpha'_1$, $\mu \circ \alpha_2 = \xi \bullet \alpha'_2$ et $\delta = \delta' \circ (\upsilon \bullet u'_2) \circ (u_1 \circ \mu)$, comme dans



Preuve du théorème. Commençons par noter que la fonction compl est bien définie, ce qui est une conséquence facile de la cartésianité de $\alpha_{u_2,k}$.

Le lemme fournit ensuite une fonction candidate pour l'inverse de compl, qu'on appelle scind et qui envoie (w, α) sur $(u_1, u_2, \alpha_1, \delta)$ comme dans le lemme ??. Cette fonction est bien définie par minimalité faible : étant donnés (w, α) et (w', α') dans $\mathrm{EInj}((v, x), u)$, si $\alpha = \alpha' \circ (v \bullet \gamma)$, alors $\mathrm{scind}(w, \alpha) \sim \mathrm{scind}(w', \alpha')$ au sens de IExt. On vérifie

facilement que $\mathrm{compl} \circ \mathrm{scind} = \mathrm{id}_{\mathrm{EInj}}$ par cartésianité du relèvement adéquat. De même, $\mathrm{scind} \circ \mathrm{compl} = \mathrm{id}_{\mathrm{IExt}}$ par minimalité faible. $\square$

La famille d'ensembles IExt hérite ainsi d'une structure de profoncteur $\mathbb{P}(X) \nrightarrow \mathbb{V}_X$. On peut de plus vérifier facilement que l'action de $\mathbb{P}(X)$ définie sur EInj par le diagramme *(12)* correspond via la bijection à l'action évidente sur IExt :

Proposition 5.3.7. L'action de $\mathbb{P}(X)$ sur IExt induite par celle sur EInj via la bijection associe simplement à $q = (u_1, u_2, \alpha, \delta)$ et (w', γ) avec $\gamma : (u \bullet w) \to u'$ le quadruplet $(w', \gamma) \cdot q = (u_1, u_2 \bullet w, \alpha, \gamma \circ (\delta \bullet w))$, comme dans

$$\begin{tikzcd}
 & & & Y' \arrow[dl, "w"'] \arrow[ddl, bend left] \\
 & & Y \arrow[dl, "u_2"'] \arrow[ddl, bend left] & \\
{[n'_v]} \arrow[r] \arrow[d, "v"'] \arrow[dr, phantom, "\overset{\alpha}{\Longrightarrow}"] & Z \arrow[d, "u_1"] \arrow[dr, phantom, "\overset{\delta}{\Longrightarrow}"] & & \\
{[n_v]} \arrow[r, "x"'] & X. & &
\end{tikzcd}$$

On vérifie enfin :

Proposition 5.3.8. Le foncteur $\mathrm{Sing} : \widehat{\mathbb{V}_X} \to \widehat{\mathbb{P}(X)}$ obtenu coïncide bien avec celui de l'article.

Démonstration. Le premier envoie un comportement $B \in \widehat{\mathbb{V}_X}$ sur

$$u \mapsto \widehat{\mathbb{V}_X}(\mathrm{EInj}(-, u), B).$$

Par l'expression bien connue des transformations naturelle comme des fins, cela revient à

$$u \mapsto \int_{V \in \mathbb{V}_X} \mathrm{Set}(\mathrm{EInj}(V, u), B(V)),$$

soit exactement la formule obtenue dans l'article, puisqu'on a par construction $\mathrm{EInj}(V, u) = \mathbb{E}_X(V, (u, \mathrm{id}_X))$. $\square$

Remarque 5.3.7. Un autre point de vue sur le passage des comportements innocents aux stratégies est que l'inclusion $\mathbb{V}_X \hookrightarrow \mathbb{E}_X$ induit une topologie de Grothendieck sur $\mathbb{E}_X$

et que la bonne notion de stratégie est celle de faisceau sur $\mathbb{E}_X$ pour cette topologie. L'intuition est qu'une partie est couverte par ses vues, dans un sens généralisant le recouvrement d'un espace topologique par des ouverts, et que l'acceptation d'une partie u par un faisceau est locale, c'est-à-dire qu'elle ne dépend que de l'acceptation de chacune des vues de u. Autrement dit, une partie est acceptée si et seulement si toutes ses vues le sont. Comme les stratégies peuvent ici accepter une partie de plusieurs manières, ceci se raffine en : une manière d'accepter une partie est précisément une famille compatible de manières d'accepter chacune de ses (occurrences de) vues. Bien que ce point de vue ait guidé l'ensemble du développement, il a rencontré une résistance très forte auprès des relecteurs d'articles, ainsi que des collègues à qui j'ai tenté de vendre la mobylette. Il n'est donc mentionné que de façon anecdotique dans les articles publiés. Récemment, néanmoins, plusieurs semblent l'avoir adopté : Paul-André Melliès l'a mis en avant dans un exposé à l'IHP en 2014 et Tsukada et Ong l'ont appliqué à une variante non-déterministe de PCF [73].

On conclut cette partie en définissant les stratégies innocentes :

Définition 5.3.12. Une stratégie $S \in \widehat{\mathbb{P}(X)}$ est *innocente* ssi elle est dans l'image essentielle du foncteur Sing : $\widehat{\mathbb{V}_X} \to \widehat{\mathbb{P}(X)}$.

5.4 Adéquation

5.4.1 Adéquation intentionnelle pleine et équivalences de test

On s'intéresse dans cette partie à démontrer que les comportements innocents mènent à un modèle adéquat de CCS. Premier point : l'interprétation des processus comme des comportements innocents saute aux yeux. On le montre en partie 5.4.4. Démontrer l'adéquation de cette interpretation est plus délicat, d'abord parce qu'il faut définir soigneusement le sens d'« adéquation ».

Commençons par expliquer en quoi les comportements innocents sont une représentation résolument intentionnelle des processus.

Exemple 5.4.12. Considérons la loi d'expansion de Milner :

$$a \,|\, b \sim a \,.\, b + b \,.\, a \,,$$

qui caractérise l'interprétation par entrelacement des programmes concurrents. Cette loi est satisfaite par toute équivalence comportementale raisonnable sur les processus. Or, les deux comportements innocents diffèrent. En effet, par exemple, le comportement correspondant

au premier processus ci-dessus refuse de recevoir sur a mais accepte le *fork*, alors que c'est le contraire pour le comportement correspondant au second processus.

On ne peut donc pas espérer obtenir de modèle dénotationnel d'équivalences comportementales en interprétant les processus comme des comportements innocents. Au lieu de cela, on met à profit leur riche structure pour imiter une large classe d'équivalences comportementales. On démontre ensuite que pour chacune de ces équivalences, la traduction induit une bijection sur les quotients. Plus précisément :

Définition 5.4.13. Etant donnés deux ensembles A et B munis chacun d'une relation d'équivalence, disons $\sim_A$ et $\sim_B$, une fonction $A \to B$ est dite *intentionnellement pleinement adéquate* ssi elle induit une fonction $A/{\sim_A} \to B/{\sim_B}$ bien définie, qui est de plus une bijection.

Cette notion semble avoir été introduite dans l'article originel de Hyland et Ong [38].

Proposition 5.4.9. Une telle fonction $f : A \to B$ est intentionnellement pleinement adéquate ssi

- **compatibilité et injectivité essentielle :** pour tous $a, a' \in A$, $(a \sim_A a') \iff (f(a) \sim_B f(a'))$ et
- **surjectivité essentielle :** pour tout $b \in B$, il existe $a \in A$ tel que $f(a) \sim_B b$.

Démonstration. La condition $(a \sim_A a') \implies (f(a) \sim_B f(a'))$ est équivalente au fait que f induise une fonction bien définie entre les quotients. La réciproque est équivalente à l'injectivité de cette fonction. Enfin, la condition de surjectivité essentielle est équivalente à la surjectivité de la fonction induite. □

Remarque 5.4.8. Une démonstration similaire mais plus rigoureuse : les conditions de la proposition reviennent précisément au fait que la fonction $A \to B$ donnée s'étend en une équivalence entre les groupoïdes induits respectivement par $\sim_A$ et $\sim_B$. On a donc les deux côtés verticaux et l'un des deux côtés horizontaux d'un carré commutatif d'équivalences

$$\begin{tikzcd}
(A, \sim_A) \arrow[r] \arrow[d, "\simeq"] & (B, \sim_B) \arrow[d, "\simeq"] \\
A/{\sim_A} \arrow[r] & B/{\sim_B},
\end{tikzcd}$$

qui induisent évidemment le quatrième côté par composition.

Concluons cette partie introductive en expliquant quelle classe d'équivalences comportementales est gérée par notre approche. L'idée de base est qu'il s'agit des

équivalences de test, dont les premiers exemples ont été introduits par De Nicola et Hennessy [13] et dont nous donnons ici une définition générale — sans prétendre à l'universalité. Une équivalence de test se définit en deux temps : on fixe d'abord le *mode d'interaction*, puis la notion de *réussite*. Ici, nous utilisons le même mode d'interaction pour toutes, la composition parallèle. La notion de réussite est donnée par un pôle, dans le sens suivant.

Définition 5.4.14. Soit Σ le graphe réflexif avec un sommet $\star$ et $\Sigma(\star, \star) = \{\tau, \heartsuit\}$, avec comme identité sur $\star$ l'arête τ.

Un *pôle* est une propriété des états de systèmes de transitions sur $\mathrm{fc}(\Sigma)$, qui est stable par bisimilarité forte.

Cette définition appelle quelques explications. D'abord, elle présente un problème de taille, puisqu'on a une classe propre d'états de systèmes de transitions sur $\mathrm{fc}(\Sigma)$. On laisse le soin au lecteur de résoudre cette question de sa manière préférée, par exemple en utilisant un univers, ou en définissant « propriété » par « formule de logique modale ». Ensuite, il peut sembler étrange de considérer la bisimilarité forte sur $\mathrm{fc}(\Sigma)$, mais si le système de transitions considéré est de la forme $\mathrm{fc}(p) : \mathrm{fc}(G) \to \mathrm{fc}(\Sigma)$, cela revient à la bisimilarité faible sur G.

Exemple 5.4.13. L'ensemble des états x tels que pour toute transition silencieuse $x \leftarrow x'$ il existe une transition $x' \xleftarrow{\heartsuit} x''$ est un pôle, noté $\perp_f$. L'équivalence induite (voir ci-dessous) s'appelle l'équivalence de *test équitable* et a été introduite simultanément par Brinksma et al. [6] et Natarajan et Cleaveland [60].

Un autre pôle sensé, noté $\perp_+$ est constitué de tous les états x tels que pour toute transition $x \leftarrow x'$ éventuellement non silencieuse, il existe une transition $x' \xleftarrow{\heartsuit} x''$. En comparaison avec $\perp_f$, il requiert que x ne perde jamais la capacité d'acceptation (c'est-à-dire d'effectuer une transition $\heartsuit$), même après avoir déjà accepté.

Enfin, le pôle constitué des états acceptants, c'est-à-dire les x tels qu'il existe une transition $x \xleftarrow{\heartsuit} x'$, donnera l'équivalence de test dite *angélique* (*may testing* en anglais).

On définit maintenant la notion d'équivalence de test, en utilisant le matériel introduit en partie 2.3.

Définition 5.4.15. Un *graphe testable* est un système de transitions $p : G \to \mathrm{fc}(\Sigma)$ sur $\mathrm{fc}(\Sigma)$, muni d'une relation $\rhd_G : \mathrm{ob}(G)^2 \nrightarrow \mathrm{ob}(G)$ dont le domaine, noté $\smile_G$, est une relation d'équivalence et qui est de plus partiellement fonctionnelle à bisimilarité forte près.

Demander que le domaine soit une relation d'équivalence signifie que l'ensemble des $(x, y) \in \mathrm{ob}(G)^2$ tels qu'il existe un $z \in \mathrm{ob}(G)$ vérifiant $((x, y), z) \in \rhd_G$ forme une relation

d'équivalence sur $\mathrm{ob}(G)$. La fonctionnalité partielle à bisimilarité forte près signifie que si (x, y) est relié à deux états z et z', alors $z \sim z'$.

Notation 5.4.3. On note $x^{\smile}$ l'ensemble des $y \in G$ tel que $x \smile_G y$. On note de plus la relation $\rhd_G$ en infixe, c'est-à-dire que (x, y) est relié à z s'écrit $(x, y) \rhd_G z$. On fixe de plus un choix global, pour tous $x \smile_G y$, de z tel que $(x, y) \rhd_G z$. On obtient une fonction partielle $\mathrm{ob}(G)^2 \to \mathrm{ob}(G)$, qu'on note $(x, y) \mapsto (x \| y)$, dont le domaine est celui de $\rhd_G$.

Remarque 5.4.9. Notre définition, bien que formellement différente de celle de De Nicola et Hennessy, est dans le même esprit et nous ne prétendons nullement à l'originalité en la matière.

Définition 5.4.16. Etant donné un pôle $\perp$ et un graphe testable $p : G \to \mathrm{fc}(\Sigma)$, on définit l'*équivalence de test* $\sim_G^{\perp}$ induite par $\perp$ sur G par $x \sim_G^{\perp} y$ ssi $x \smile_G y$ et pour tout $z \smile_G x$,

$$((x \| z) \in \perp) \iff ((y \| z) \in \perp).$$

Intuitivement, x et y sont équivalents s'ils passent exactement les mêmes tests.

Notre approche consiste d'abord à voir à la fois CCS et les comportements innocents en des graphes testables. Nous définissons ensuite la fonction de traduction $[\![-]\!] : \mathrm{CCS} \to \mathrm{Comp}$, où $\mathrm{Comp} = \sum_n \widehat{\mathbb{V}_{[n]}}$, puis nous prouvons que pour tout pôle $\perp$, la traduction est intentionnellement pleinement adéquate pour $\sim_{\mathrm{CCS}}^{\perp}$ et $\sim_{\mathrm{Comp}}^{\perp}$.

5.4.2 Processus

Jusqu'ici nous avons défini notre modèle sans référence explicite à CCS. Il nous faut donc préciser quelle variante on considère. Pour alléger la syntaxe, mais aussi pour s'épargner la question ouverte posée à la fin du premier article sur CCS [35], on travaille avec des termes infinis. L'ensemble des termes est ainsi défini coïnductivement par les règles en figure 5.3.

On y présente deux jugements, $\vdash$ pour les termes et $\vdash_g$ pour les termes *gardés*, qui servent pour définir les sommes. Une particularité de notre présentation est que tous les préfixes, y compris la création de canal $\nu\, a\, .\, P$, sont acceptés dans les sommes (gardées). Un mot sur le traitement des variables liées : Γ désigne un ensemble fini de canaux pris dans un ensemble infini, disons les entiers.

Comme dans la littérature, $\nu\, a\, .\, P$ lie a dans P et on utilise les conventions habituelles de renommage sans capture.

Remarque 5.4.10. On pourrait formaliser la syntaxe de CCS dans le cadre de la partie 3, comme engendrée par une signature à deux sortes, t pour les processus et c pour les canaux, dans laquelle ν serait une constante de type $t^c \vdash t$.

Notation 5.4.4. La notation Γ, a est définie ssi $a \notin \Gamma$ et dans ce cas désigne $\Gamma \uplus \{a\}$.

On construit maintenant un système de transitions dont les sommets sont les termes CCS sur le graphe Σ de la définition 5.4.14. On donne une présentation dans le style de la « machine chimique » de Berry et Boudol [4]. Pour cela, on définit à partir des termes l'ensemble suivant :

Définition 5.4.17. Une *configuration* est une paire constituée d'une partie finie de $\mathbb{N}$, Γ, et d'un multi-ensemble fini de processus P tels que $\Gamma \vdash P$.

Notation 5.4.5. On note une telle paire $\langle \Gamma \| M \rangle$ avec M un multi-ensemble, que l'on note comme une liste $M = [P_1, \ldots, P_n]$. On omet les crochets pour alléger la lecture lorsque cela ne crée pas d'ambiguïté, comme par exemple dans $\langle \Gamma \| P_1, \ldots, P_n \rangle$. On note M, N l'union multi-ensembliste de M et N.

On ne prend pas la peine de définir le renommage (sans capture), mais on retient :

Proposition 5.4.10. Le renommage étend la famille

$$\begin{aligned} \mathrm{set} &\to \mathrm{Set} \\ \Gamma &\mapsto \mathrm{CCS}_\Gamma, \end{aligned}$$

où CCS_Γ désigne l'ensemble des termes CCS typés dans Γ, en un foncteur.

Remarque 5.4.11. Si Σ désigne la signature esquissée dans la remarque *5.4.10* et $\mathscr{C}_\Sigma$ la catégorie cartésienne fermée engendrée, le foncteur $\Gamma \mapsto \mathrm{CCS}_\Gamma$ peut se définir à partir du représentable $\mathbf{y}_t : \mathscr{C}_\Sigma^{\mathrm{op}} \to \mathrm{Set}$. En effet, le plongement $\mathrm{set}^{\mathrm{op}} \hookrightarrow \mathscr{C}_\Sigma$, standard dans les théories de Lawvere, induit un foncteur $F : \mathrm{set} \to \mathrm{Set}$ et on a $\mathbf{y}_t \circ F^{\mathrm{op}} \cong \mathrm{CCS}_{\_}$.

Notation 5.4.6. Pour tous $\Gamma \vdash_g P$ et $\Gamma \vdash Q$ avec $Q = \sum_{i \in n} Q_i$, ainsi que pour toute injection $h : n \hookrightarrow n+1$, on note $P +_h Q$ pour la somme $\sum_{j \in n+1} P_j$, où $P_{h(i)} = Q_i$ pour tout $i \in n$ et $P_{j_0} = P$ pour j_0 l'unique élément de $(n+1) \setminus \mathrm{im}(h)$.

$$\frac{\Gamma \vdash_g P_1 \quad \ldots \quad \Gamma \vdash_g P_n}{\Gamma \vdash \sum_{i \in n} P_i} \qquad \frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P | Q}$$

$$\frac{\Gamma, a \vdash P}{\Gamma \vdash_g \nu a \,.\, P} \qquad \frac{\Gamma \vdash P}{\Gamma \vdash_g \heartsuit .\, P} \qquad \frac{\Gamma \vdash P}{\Gamma \vdash_g \tau .\, P} \qquad \frac{a \in \Gamma \quad \Gamma \vdash P}{\Gamma \vdash a \,.\, P} \qquad \frac{a \in \Gamma \quad \Gamma \vdash P}{\Gamma \vdash \bar{a} .\, P}$$

Figure 5.3 – Termes de CCS

Les transitions pour CCS sont définies en figure 5.4, en utilisant la notation *(2.3.2)*. Ceci forme un graphe réflexif CCS, muni d'une projection $p_{CCS} : CCS \to \Sigma$. Nous munissons à présent ce graphe avec d'une structure de graphe testable :

Proposition 5.4.11. La relation $\rhd_{CCS}$ définie comme l'ensemble des triplets $(\langle \Gamma \| M \rangle, \langle \Gamma \| N \rangle, \langle \Gamma \| M, N \rangle)$ fait de CCS un graphe testable.

Démonstration. La relation donnée est partiellement fonctionnelle, donc *a fortiori* partiellement fonctionnelle modulo bisimilarité forte. □

$$\frac{}{\langle \Gamma \| \tau.\, P +_h Q \rangle \xleftarrow{\tau} \langle \Gamma \| P \rangle} \qquad \frac{}{\langle \Gamma \| \heartsuit.\, P +_h Q \rangle \xleftarrow{\heartsuit} \langle \Gamma \| P \rangle}$$

$$\frac{}{\langle \Gamma \| \nu a.\, P +_h Q \rangle \xleftarrow{\tau} \langle \Gamma, a \| P \rangle} \qquad \frac{}{\langle \Gamma \| (a.\, P +_h Q), (\overline{a}.\, P' +_k Q') \rangle \xleftarrow{\tau} \langle \Gamma \| P, P' \rangle}$$

$$\frac{}{\langle \Gamma \| (P | Q) \rangle \xleftarrow{\tau} \langle \Gamma \| P, Q \rangle} \qquad \frac{\langle \Gamma_1 \| S_1 \rangle \xleftarrow{\alpha} \langle \Gamma_2 \| S_2 \rangle}{\langle \Gamma_1 \| S \cup S_1 \rangle \xleftarrow{\alpha} \langle \Gamma_2 \| S \cup S_2 \rangle}$$

Figure 5.4 – Système de transitions étiquetées pour CCS

5.4.3 Un système de transitions pour les comportements innocents

On s'occupe maintenant de la définition du système de transitions pour les comportements innocents, sur le même alphabet Σ. Cette construction repose sur deux éléments clefs :

- la notion de *résidu* $B \cdot M$ d'un comportement B selon un coup M, qui mène à une syntaxe confortable pour les comportements innocents, et
- la notion de restriction $B_{|\sigma}$ d'un comportement B à un *état* σ.

L'intuition de ce qu'est le résidu $B \cdot M$ est très simple : c'est le comportement obtenu en continuant à jouer comme B, après que le coup M a été joué. Avec les processus, c'est impossible à définir en général. Par exemple, pour le processus $a.\, 0$ de réception sur a, aucun processus ne décrit son résidu après, par exemple une synchronisation. L'ingrédient manquant est l'analogue du zéro des entiers naturels, la stratégie vide $\emptyset$!

Techniquement, tout repose sur le lemme « des vues » :

Lemme 5.4.12. Pour toute partie $u : Y \twoheadrightarrow X$ et tout joueur $y : [n] \to Y$, il existe une cellule

$$\begin{tikzcd}
{[n]} \arrow[r, "y"] \arrow[d, "\nu^{y,u}"'] \arrow[dr, phantom, "\Downarrow \alpha^{y,u}"] & Y \arrow[d, "u"] \\
{[n]^{y,u}} \arrow[r, "y^u"'] & X
\end{tikzcd}$$

avec $v^{y,u}$ une vue, unique à unique isomorphisme près.

Démonstration. On s'attaque d'abord à la question de l'existence. On décompose u en coups pour se réduire au cas où u est un coup. Alors, si y est créé par u, une analyse par cas nous fournit les données souhaitées. Par exemple, si u est un *fork*, alors si y désigne l'avatar de gauche, on prend pour $v^{y,u}$ un *fork* gauche ; sinon un *fork* droit convient. Si y n'est pas créé par u, la vue identité $\mathrm{id}^{\bullet}_{[n]}$ convient.

On se réfère à l'article sur CCS pour une définition plus précise de la propriété d'unicité, mais l'intuition est que pour chaque coup, le choix de la graine est déterminé par u et y. Toute autre vue doit donc être isomorphe à celle construite ci-dessus. □

Définition 5.4.18. Etant donné un comportement $B \in \widehat{\mathbb{V}_X}$ une partie $u : Y \rightarrowtail X$, soit $B \cdot u$ le comportement défini par $(B \cdot u)(w, y) = B(v^{y,u} \bullet w, y^u)$, pour tout $(w, y) \in \mathbb{V}_Y$. On appelle $B \cdot u$ le *résidu* de B selon u.

Définissons maintenant la notion de restriction. En fait, la question principale à ce sujet est : « restriction à quoi » ? Dans notre cas, étant donné un comportement innocent B sur X, il suffit de considérer l'ensemble

$$\mathrm{Initiaux}(B) = \prod_{n \in \mathbb{N}} \prod_{x : [n] \to X} B(\mathrm{id}^{\bullet}_{[n]}, x)$$

des états initiaux de B.

Proposition 5.4.13. Pour tout $B \in \widehat{\mathbb{V}_X}$, on a $\mathrm{Initiaux}(B) \cong \mathrm{Sing}(B)(\mathrm{id}^{\bullet}_X)$.

On a ensuite :

Définition 5.4.19. La *restriction* de B à un état $\sigma \in \mathrm{Initiaux}(B)$, notée $B_{|\sigma}$, est définie pour tout $x : [n] \to X$ et $v : [n'] \rightarrowtail [n]$ par le pullback

$$\begin{tikzcd} B_{|\sigma}(v, x) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B(v, x) \arrow[d] \\ 1 \arrow[r, "\ulcorner \sigma(n,x) \urcorner"'] & B(\mathrm{id}^{\bullet}_{[n]}, x), \end{tikzcd}$$

où la fonction $B(v, x) \to B(\mathrm{id}^{\bullet}_{[n]}, x)$ est l'image par B de l'unique morphisme $(\mathrm{id}^{\bullet}_{[n]}, x) \to (v, x)$ dans $\mathbb{V}_X$. L'action de $B_{|\sigma}$ sur les morphismes se construit par propriété universelle du pullback.

Un travail conséquent est fait dans l'article [31] pour justifier cette définition dans le cas d'un résidu $B \cdot M$ avec M un coup. Contentons-nous ici de remarquer que si par exemple X est un représentable $[n]$ et M est un *fork*, et si B a deux états initiaux e et e' acceptant

chacun π^l_n et π^r_n d'une manière, alors $\mathrm{Initiaux}(B \cdot M) = B(\pi^l_n, \mathrm{id}_{[n]}) \times B(\pi^l_n, \mathrm{id}_{[n]}) \cong 4$, ce qui diffère du nombre d'états globaux après M, puisque $\mathrm{Sing}(B)(M) \cong 2$ contient la paire d'états issus de e et celle issue de e'. Néanmoins, on a

Proposition 5.4.14. Pour tout coup $M : Y \rightarrowtail X$ et $B \in \widehat{\mathbb{V}_X}$, si B est *défini*, c'est-à-dire que chaque joueur a exactement un état initial (formellement, on a pour tout $x : [n] \to X$ que $B(\mathrm{id}^\bullet_{[n]}, x) \cong 1$), alors $\mathrm{Sing}(B)(M) \cong \mathrm{Initiaux}(B \cdot M)$.

Démonstration. Chaque élément de $\mathrm{Sing}(B)(M)$ est une famille compatible d'états de B sur les vues de M. Mais comme sur les vues identité B est constant et isomorphe à 1, la compatibilité est gratuite et on se réduit à un produit

$$\prod_{x : [n] \to X,\, v : [n'] \rightarrowtail [n],\, v \neq \mathrm{id}^\bullet_{[n]}} B((v, x))^{\mathrm{EInj}((v,x), M)}.$$

Pour chaque (x, v) et $\alpha \in \mathrm{EInj}((v, x), M)$, en posant $y(x, v, \alpha) = \mathrm{dom}(\alpha)$ on a par le lemme des vues que $v = v^{y(x,v,\alpha), M}$ et $\alpha = \alpha^{y(x,v,\alpha), M}$. On en arrive donc au produit $\prod_{y : [n'] \to Y} B((v^{y,M}, y^M), M)$, c'est-à-dire $\prod_{y : [n'] \to Y} (B \cdot M)(\mathrm{id}^\bullet_{[n]}, y)$ comme souhaité. □

On peut enfin définir le système de transitions sur les comportements innocents :

Définition 5.4.20. Un coup $X \rightarrowtail Y$ est *clos* si c'est un fork, un tick, une création, ou une synchronisation.

Soit $\mathscr{S}$ le graphe ayant

- pour sommets les paires d'une position X et d'un comportement innocent défini D sur X, ainsi que
- pour arêtes $(X, D) \leftarrow (X', D')$ les paires d'un coup clos $M : X' \rightarrowtail X$ et d'un état σ de $D \cdot M$ tel que $D \cdot M_{|\sigma} \cong D'$.

Soit de plus $p^{\mathscr{S}} : \mathscr{S} \to \Sigma$ le morphisme de graphes envoyant toute paire (M, σ) sur $\heartsuit$ si $M(\heartsuit) \neq \emptyset$ et sur τ sinon.

On fait ensuite de $\mathscr{S}$ un graphe testable. On définit pour cela l'action des morphismes horizontaux h sur les comportements innocents B, notée $B \cdot h$ — tout comme l'action $B \cdot u$ d'une partie u, mais le typage ne laisse pas trop d'ambiguïté :

Définition 5.4.21. Etant donné $B \in \widehat{\mathbb{V}_X}$ et $h : Y \to X$, soit $B \cdot h$ le comportement innocent sur Y défini par $(B \cdot h)(v, y) = B(v, h \circ y)$.

Cette action donne un foncteur $\mathbb{D}^{\mathrm{op}}_h \to \mathrm{Cat}$.

On définit enfin la relation $\rhd_{\mathscr{S}}$ comme l'ensemble des $((X, D_X), (Y, D_Y), (Z, D_Z))$ tels que $X(\star) = Y(\star) = I$, Z est un pushout de X et Y le long de I, comme dans

$$\begin{tikzcd} I \arrow[r] \arrow[d] & Y \arrow[d, "h_Y"] \\ X \arrow[r, "h_X"'] & Z \arrow[ul, phantom, "\ulcorner"] \end{tikzcd},$$

et de plus $D_X = D_Z \cdot h_X$ et $D_Y = D_Z \cdot h_Y$.

Le sens de cette définition est que (Z, D_Z) est partagé en deux équipes, jouant respectivement comme prescrit par D_X et D_Y. On obtient clairement une relation d'équivalence partiellement fonctionnelle à bisimilarité fort près, donc

Proposition 5.4.15. La relation $\rhd_{\mathscr{S}}$ fait de $\mathscr{S}$ un graphe testable.

5.4.4 Interprétation des processus et adéquation

Pour traduire les processus CCS en comportements innocents, nous commençons par exhiber une présentation syntaxique de ces derniers. Le principe de cette présentation est qu'un comportement innocent B sur un seul joueur $[n]$ est entièrement déterminé par son ensemble d'états initiaux $B_0 = B(\mathrm{id}^{\bullet}_{[n]}, \mathrm{id}_{[n]})$, ainsi, pour chaque $\sigma \in B_0$, que par la fonction associant à chaque coup $M : [n'] \rightarrowtail [n]$ le comportement $(B_0)_{|\sigma} \cdot M$. Puisqu'on raisonne dans $\mathbb{V}_X$, M est aussi une vue. On donne un nom à ce genre de coups :

Définition 5.4.22. Un coup qui est aussi une vue est dit *basique*. On note ces coups avec un indice b sur la flèche, comme dans $M : [n'] \rightarrowtail_b [n]$.

Intuitivement, comme $[n]$ n'a qu'un joueur, $\mathbb{V}_{[n]}$ est isomorphe à $\mathbb{V}([n])$, la sous-catégorie pleine de $\mathbb{P}([n])$ constituée des vues. Sur $\mathrm{id}^{\bullet}_{[n]}$, B vaut B_0 par définition ; et sur $v = b \bullet v'$ avec $b : [n'] \rightarrowtail [n]$, on a

$$\begin{aligned} B(b \bullet v', \mathrm{id}_{[n]}) &= \textstyle\sum_{\sigma \in B_0} B_{|\sigma}(b \bullet v', \mathrm{id}_{[n]}) \\ &= \textstyle\sum_{\sigma \in B_0} (B_{|\sigma} \cdot b)(v', \mathrm{id}_{[n']}). \end{aligned}$$

Ce principe mène directement à la présentation syntaxique souhaitée, dont l'énoncé formel est que la famille $[n] \mapsto \mathrm{ob}\left(\widehat{\mathbb{V}_{[n]}}\right)$ (ou plus précisément sa restriction aux foncteurs $\mathbb{V}_X^{\mathrm{op}} \to \mathrm{ford}$ (voir partie 2.1) est une coalgèbre terminale pour l'endofoncteur

$$\begin{aligned} \mathrm{Set}^{\mathbb{N}} &\to \mathrm{Set}^{\mathbb{N}} \\ (U_n)_{n \in \mathbb{N}} &\mapsto \textstyle\sum_{B_0 \in \mathrm{ford}} \left(\prod_{b : [n'] \rightarrowtail_b [n]} U_{n'}\right)^{B_0} \\ &\cong \left(\textstyle\prod_{b : [n'] \rightarrowtail_b [n]} U_{n'}\right)^{\star}. \end{aligned}$$

Comme toute stratégie à valeurs dans set est isomorphe à une stratégie à valeurs dans ford, ce résultat signifie en pratique que les comportements innocents sont en bijection à isomorphisme près avec les termes infinis de la syntaxe suivante :

$$\frac{n \vdash_D D_1 \quad \dots \quad n \vdash_D D_p}{n \vdash \sum_{i \in p} D_i}_{(p \in \mathbb{N})} \qquad \frac{\dots \quad n_b \vdash B_b \quad \dots}{n \vdash_D \left\langle (B_b)_{b : [n_b] \to_b [n]} \right\rangle} .$$

Cette syntaxe comporte deux jugements, $\vdash_D$ pour former des comportements innocents définis et $\vdash$ pour en former des généraux. Sémantiquement, la première règle forme un coproduit dans $\widehat{\mathbb{V}_{[n]}}$: de p comportements définis, on forme un comportement à p états initiaux. La seconde règle décrit un comportement défini en donnant pour chaque coup basique b son résidu selon b.

Bien que très simple, cette syntaxe est très expressive, comme on va le voir en traduisant les processus.

Exemple 5.4.14. Par exemple, pour former l'analogue de $P|Q$, on considère $\langle \pi^l_n \mapsto P, \pi^r_n \mapsto Q \rangle$, en adoptant ici la convention suivante :

Notation 5.4.7. Les coups basiques non mentionnés entre piquants sont envoyés sur la stratégie vide $\emptyset$, syntaxiquement désignée par la somme vide ($p = 0$ dans la première règle).

Exemple 5.4.15. De même, pour désigner une somme gardée, disons $a\,.\,P + a\,.\,Q + \overline{b}.\,R$, on forme

$$\langle \iota_{n,a} \mapsto (P + Q)\,,\, o_{n,b} \mapsto R \rangle.$$

Détaillons un peu : le coproduit $+$ de comportements innocents s'apparente sémantiquement à l'opération de choix interne, habituellement notée $\oplus$, qu'on n'a pas incluse dans notre présentation de CCS par souci de simplicité. Sa sémantique est simplement définie par la règle $P \oplus P' \xleftarrow{\tau} P$ et sa symétrique. Notre interprétation du processus ci-dessus identifie $a\,.\,P + a\,.\,Q$ et $a\,.\,(P \oplus Q)$. Cette identité est valide dans toutes les équivalences de test, dans le sens de la définition 5.4.16, même si elle est fausse du point de vue de la bisimilarité (forte ou faible).

Bien que très expressive, notre syntaxe présente un défaut d'expressivité vis-à-vis du choix interne. Par exemple, il fait sens de former le processus $(P|P') \oplus (Q|Q')$. Cependant si on applique naïvement les méthodes utilisées dans les deux exemples précédents, on obtient $\langle \pi^l_n \mapsto P \oplus Q\,,\, \pi^r_n \mapsto P' \oplus Q' \rangle$. Ce comportement correspond plutôt à $(P \oplus Q)|(P' \oplus Q')$, qui diffère clairement du processus de départ puisqu'il admet une transition vers $P|Q'$. Si on souhaitait inclure le choix interne dans notre interprétation de CCS, on contournerait le problème en traduisant $P \oplus Q$ comme la somme gardée $\tau\,.\,P + \tau\,.\,Q$.

Nous sommes maintenant en mesure de donner notre traduction des processus, définie par coïnduction sur les processus (infinis). Pour simplifier l'exposé, on se limite aux processus dont le contexte Γ est un ordinal fini. Le travail complet est fait dans l'article sur π, mais donne un résultat nettement plus lourd. (On suppose en particulier que dans $n \vdash \nu a \,.\, P$, $a = (n+1)$.) Notre traduction $[\![-]\!]$ est définie coïnductivement par :

$$\begin{aligned} \mathrm{ob}(\mathrm{CCS}) &\to \mathrm{ob}(\widehat{\mathbb{V}_X}) \\ n \vdash P|Q &\mapsto \langle \pi^l_n \mapsto [\![P]\!] \,,\, \pi^r_n \mapsto [\![Q]\!] \rangle \\ n \vdash \textstyle\sum_{i \in p} \alpha_i \,.\, P_i &\mapsto \left\langle \left(b \mapsto \textstyle\sum_{\{i \in p \mid [\![\alpha_i]\!] = b\}} [\![P_i]\!] \right)_{b : [n'] \twoheadrightarrow_b [n]} \right\rangle, \end{aligned}$$

où on définit $[\![n \vdash \alpha]\!]$ par :

$$\begin{aligned} \tau &\mapsto \tau_n \\ \heartsuit &\mapsto \heartsuit_n \\ \nu a &\mapsto \nu_n \\ a &\mapsto \iota_{n,a} \\ \bar{a} &\mapsto o_{n,a} \,. \end{aligned}$$

On peut enfin énoncer :

Théorème 5.4.16. Pour tout pôle $\perp$ et tous processus $\Gamma \vdash P$ et $\Gamma \vdash Q$, on a

$$(P \sim^{\perp}_{\mathrm{CCS}} Q) \iff \left([\![P]\!] \sim^{\perp}_{\mathscr{P}} [\![Q]\!] \right).$$

De plus, pour tout comportement innocent $B \in \widehat{\mathbb{V}_X}$, il existe un processus P tel que $[\![P]\!] \sim^{\perp}_{\mathscr{P}} B$. Ensemble ces deux résultats signifient que $[\![-]\!]$ est intentionnellement pleinement adéquate.

On omet ici la démonstration en renvoyant le lecteur aux articles publiés.

Mentionnons néanmoins deux remarques importantes.

D'abord, l'équivalence de test équitable sémantique, c'est-à-dire côté comportements innocents, admet grâce au foncteur Sing de la proposition 5.3.8 une formulation entièrement en termes ludiques. On commence par décrire l'appartenance d'une stratégie naïve $S \in \widehat{\mathbb{P}(X)}$ au pôle en demandant que pour toute partie close $u \in \mathbb{P}(X)$ sans tick et tout $\sigma \in S(u)$, il existe une extension $(w, \alpha) : u \to u'$ de u où u' est close et contient au moins un tick, munie d'un $\sigma' \in S(u')$ tel que $\sigma' \cdot (w, \alpha) = \sigma$. On observe enfin que deux comportements innocents $B, C \in \widehat{\mathbb{V}_X}$ sont équitablement équivalents précisément lorsque pour toute position Y telle que $Y(\star) = X(\star) = I$ et tout comportement innocent $T \in \widehat{\mathbb{V}_Y}$, on ait $\mathrm{Sing}[B, T] \in \perp$ ssi $\mathrm{Sing}[C, T] \in \perp$. Ici, $[B, T]$ désigne le comportement innocent sur le pushout

$$\begin{tikzcd} I \arrow[r] \arrow[d] & Y \arrow[d, "h_Y"] \\ X \arrow[r, "h_X"'] & Z \arrow[ul, phantom, "\ulcorner"] \end{tikzcd},$$

déterminé par le fait que $[B, T] \cdot h_X = B$ et $[B, T] \cdot h_Y = T$. Ceci est bien défini en vertu du fait que $\mathbb{V}_Z \cong \mathbb{V}_X + \mathbb{V}_Y$ avec comme injections de coproduit $\mathbb{V}_{h_X}$ et $\mathbb{V}_{h_Y}$, ce qui explique au passage l'adoption de la notation de copairage :

$$\begin{tikzcd} \mathbb{V}_X^{op} \arrow[r, "\mathbb{V}_{h_X}"] \arrow[dr, "B"'] & \mathbb{V}_Z^{op} \arrow[d, "{[B,T]}"] & \mathbb{V}_Y^{op} \arrow[l, "\mathbb{V}_{h_Y}"'] \arrow[dl, "T"] \\ & \mathbf{Set}. & \end{tikzcd}$$

Seconde remarque : les démonstrations données dans l'article sur CCS [31] et dans celui sur π [15, 16] sont très différentes.

En particulier, la méthode utilisée pour CCS ne passe pas à π en l'état actuel de nos connaissances. Ma compréhension de ce phénomène rejoint la vaste question de l'opposition entre relation de réduction et système de transitions étiquetées [69, 68]. Pour résumer brièvement cette question, on connaît deux méthodes principales pour définir les langages concurrents comme CCS ou π :

- Selon la première méthode, on commence par définir une relation $\equiv$ dite de *congruence structurelle*, qui identifie par exemple $P|Q$ et $Q|P$, pour tous P et Q. On donne ensuite une relation simple dite de *réduction*, typiquement pour CCS :

$$a\,.\,P|\bar{a}.\,Q \longrightarrow P|Q,$$

  à laquelle on incorpore la congruence structurelle en ajoutant la règle

$$\frac{P \equiv P' \quad P' \longrightarrow Q' \quad Q' \equiv Q}{P \longrightarrow Q}.$$

  Cette méthode a l'avantage de la simplicité, mais se révèle peu efficace pour raisonner sur la dynamique des programmes.
- La seconde méthode corrige le problème en s'assurant dès le départ que toute étape d'exécution d'un programme P est construite par induction sur la structure de P. Le prix à payer pour cela est que la notion d'exécution devient « ouverte », en ce sens que, par exemple, toute réduction (dans le sens de la première méthode) de la forme $P|Q \longrightarrow R$ se verra décomposée en deux *transitions* $P \xrightarrow{\alpha} P'$ et $Q \xrightarrow{\bar{\alpha}} Q'$, avec $R \equiv P'|Q'$. On enrichit ainsi la relation binaire de départ en une relation ternaire : chaque transition est munie

d'une *étiquette* représentant le contexte manquant pour que la réduction ait lieu. Un exemple typique de transition étiquetée en CCS est

$$a\,.\,P \xrightarrow{a} P.$$

Intuitivement, cette transition dit que si l'environnement est prêt à émettre sur a, alors $a\,.\,P$ se réduit.

CCS et π sont tous deux munis de présentations dans les deux styles, dont on montre qu'elles coïncident, dans le sens où les transitions dites *closes* (c'est-à-dire dont l'étiquette est un certain τ fixé à l'avance) coïncident avec les réductions.

Pour revenir à la question de départ, le point crucial est que, le jeu pour π étant basé sur la réduction plutôt que sur le système de transitions étiquetées, le système de transitions étiquetées $\mathscr{S}$ de la définition 5.4.20 est trop grossier. Par exemple, il lui manque une transition pour la réception d'un canal déjà connu. Cette discrépance avec le système de transitions étiquetées standard pour π le rend inutile pour l'adéquation. La démonstration n'en utilise d'ailleurs que les transitions closes.

Tout cela soulève la question de savoir si la méthode employée pour π s'adaptera facilement à d'autres langages. Je ne sais pas répondre à cette question aujourd'hui. Il est néanmoins prévisible que certaines difficultés (dont l'ampleur reste à découvrir) se poseront dans le cadre des langages fonctionnels, pour lesquels les positions seront plus structurées que les simples « soupes » de CCS et π. Il est de plus probable qu'une explosion combinatoire rende impraticable le passage à l'échelle de cette méthode. En effet, rien que pour π, le lemme crucial, en Annexe A [16] requiert six pages pleines d'une pénible analyse de cas. On peut espérer optimiser la méthode au moyen de techniques de style « up to » [64], mais cela reste entièrement à faire.

Une piste prometteuse est l'intuition que le défaut du système de transitions $\mathscr{S}$ évoqué plus haut est secrètement un symptôme du fait que la fibration des parties au-dessus des positions n'est pas un champ, mentionné en introduction de cette partie. On peut ainsi espérer construire une variante du jeu pour π corrigeant ce défaut, selon un raffinement de la construction bien connue du champ associé à une fibration (chapitre 8.8 du *Stacks Project* [72]). A long terme, cette piste peut mener vers des progrès en matière de construction systématique de systèmes de transitions à partir de relations de réduction, l'intérêt étant qu'en opérant sur les parties plutôt que sur les termes, on simplifie grandement la situation.

5.5 Perspectives

Dans cette partie, on a donné un aperçu de l'approche faisceautique en sémantique des jeux concurrente. On a déjà donné une manière de conclusion en partie 5.1.2, ainsi que quelques perspectives en fin de partie précédente.

Mentionnons encore que la construction des stratégies naïves et des comportements innocents à partir d'une pseudo catégorie double de parties, présentée ici dans le cas de CCS, découle d'une construction générale exposée dans [31]. Cette construction devra encore être légèrement généralisée pour tenir compte des phénomènes de polarité, qui sont importants dans les modèles de jeux des langages fonctionnels. Bien qu'elle n'ait certainement pas atteint sa forme définitive, elle a le mérite d'offrir un certain nombre d'outils pour la définition de nouveaux modèles de jeux.

En revanche, la première couche, qui consiste à construire la pseudo catégorie double des parties, n'est pas encore automatisée. Dans un article en préparation avec Clovis Eberhart, nous retravaillons et généralisons la construction des pseudo catégories doubles pour CCS et π. Ce travail en profondeur nous permet de lever certains verrous et nous pensons bientôt être en mesure d'appliquer cette construction au cas des langages fonctionnels tels que PCF, le λ-calcul pur, etc. Le seul obstacle dans cette direction est que notre construction ne s'applique pas si le jeu permet aux joueurs d'« oublier » des variables, ce qui semble nécessaire dans le cas des langages fonctionnels.

Si nos idées pour résoudre ce problème fonctionnent, nous comptons mettre à profit ce progrès pour explorer les pistes suivantes.

- En premier lieu, nous souhaitons relier formellement la construction de Tsukada et Ong [73] à la nôtre.
- Nous espérons de plus construire des modèles de jeux adéquats de langages fonctionnels avec effets de bord dans le style de ceux de Murawski et Tzevelekos [59], mais bénéficiant de l'intégration de la stabilité par préfixe et de leur *équivariance* dans la notion de préfaisceau. De notre point de vue, la difficulté semble venir des positions du jeu, plus structurées que pour CCS et π donc *a priori* moins faciles à gérer. Dans le même ordre d'idées, nous souhaitons mener une étude des types dépendants dans notre cadre : là encore les positions du jeu semble être le point difficile.
- Enfin, un projet à long terme est d'étudier les traductions entre langages au niveau des pseudo catégories doubles de parties, l'objectif étant de déduire l'adéquation d'une traduction à partir des propriétés de la traduction des parties. Ce dernier point pourrait faciliter les preuves de correction de compilateurs, notamment en les rendant plus modulaires.

6 Bibliographie

[1] Samson Abramsky, Radha Jagadeesan et Pasquale Malacaria. Full Abstraction for PCF. *Information and Computation*, 163(2):409-470, 2000.

[2] Jiří Adámek et Jiří Rosicky. *Locally Presentable and Accessible Categories*. Cambridge University Press, 1994.

[3] Paolo Baldan, Andrea Corradini, Tobias Heindel, Barbara König et Paweł Sobociński. Processes and unfoldings: concurrent computations in adhesive categories. *Mathematical Structures in Computer Science*, 24(4), 2014.

[4] Gérard Berry et Gérard Boudol. The Chemical Abstract Machine. In: *POPL*, pages 81-94. 1990.

[5] Gavin M. Bierman. What is a Categorical Model of Intuitionistic Linear Logic?. In: *TLCA*, volume 902 of Lecture Notes in Computer Science, pages 78-93. Springer, 2009.

[6] Ed Brinksma, Arend Rensink et Walter Vogler. Fair Testing. In: *CONCUR*, volume 962 of Lecture Notes in Computer Science, pages 313-327. Springer, 1995.

[7] Roberto Bruni et Ugo Montanari. Cartesian Closed Double Categories, Their Lambda-Notation, and the Pi-Calculus. In: *LICS*. IEEE, 1999.

[8] Aurelio Carboni et Peter T. Johnstone. Connected Limits, Familial Representability and Artin Glueing. *Mathematical Structures in Computer Science*, 5(4):441-459, 1995.

[9] Aurelio Carboni et Peter T. Johnstone. Corrigenda for `Connected limits, familial representability and Artin glueing'. *Mathematical Structures in Computer Science*, 14(1):185-187, 2004.

[10] Aurelio Carboni, Stefano Kasangian et Robert F. C. Walters. An axiomatics for bicategories of modules. *Journal of Pure and Applied Algebra*, 45:127-141, 1987.

[11] Gian Luca Cattani et Glynn Winskel. Presheaf Models for Concurrency. In: *CSL*, , pages 58-75. Springer, 1997.

[12] Pierre-Louis Curien, Hugo Herbelin, Jean-Louis Krivine et Paul-André Melliès. *Interactive models of computation and program behaviour*. Société Mathématique de France, 2009.

[13] Rocco De Nicola et Matthew Hennessy. Testing Equivalences for Processes. *Theoretical Computer Science*, 34:83-133, 1984.

[14] James Dolan et John C. Baez. Categorification. In: *Higher Category Theory*, pages 1-36. American Mathematical Society, 1998.

[15] Clovis Eberhart, Tom Hirschowitz et Thomas Seiller. An intensionally fully-abstract sheaf model for pi. In: *CALCO*, LIPIcs. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2015.

[16] Clovis Eberhart, Tom Hirschowitz et Thomas Seiller. *An intensionally fully-abstract sheaf model for π (expanded version)*. 2015.
[17] Charles Ehresmann. *Catégories et structures*. Dunod, 1965.
[18] Lisbeth Fajstrup, Martin Raussen et Eric Goubault. Algebraic topology and concurrency. *Theoretical Computer Science*, 357(1-3):241 - 278, 2006.
[19] Marcelo P. Fiore. Fibred Models of Processes: Discrete, Continuous, and Hybrid Systems. In: *IFIP TCS*, pages 457-473. Chapman & Hall, 1998.
[20] Marcelo P. Fiore, Gordon Plotkin et Daniele Turi. Abstract Syntax and Variable Binding. In: *LICS*. IEEE, 1999.
[21] Marcelo P. Fiore et Daniele Turi. Semantics of Name and Value Passing. In: *LICS*, pages 93-104. 2001.
[22] Brendan Fong. *Causal Theories: A Categorical Perspective on Bayesian Networks*. Rapport de Master, University of Oxford, 2012.
[23] Murdoch J. Gabbay et Andrew M. Pitts. A New Approach to Abstract Syntax Involving Binders. In: *LICS*. IEEE, 1999.
[24] Fabio Gadducci et Ugo Montanari. The tile model. In: *Proof, Language, and Interaction, Essays in Honour of Robin Milner*, pages 133-166. The MIT Press, 2000.
[25] Richard H. G. Garner. *Polycategories*. Thèse de doctorat, University of Cambridge, 2006.
[26] Richard H. G. Garner et Tom Hirschowitz. *Shapely monads and analytic functors*. 2015.
[27] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50, 1987.
[28] Jean-Yves Girard. Locus Solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3):301-506, 2001.
[29] Jean-Yves Girard, Paul Taylor et Yves Lafont. *Proofs and types*. Cambridge University Press, 1989.
[30] Yves Guiraud et Philippe Malbos. Higher-dimensional categories with finite derivation type. *Theory and Applications of Categories*, 22(18):420-278, 2009.
[31] Tom Hirschowitz. Full abstraction for fair testing in CCS (expanded version). *Logical Methods in Computer Science*, 10(4), 2014.
[32] Tom Hirschowitz. Cartesian closed 2-categories and permutation equivalence in higher-order rewriting. *Logical Methods in Computer Science*, 9(3), 2013.
[33] André Hirschowitz, Michel Hirschowitz et Tom Hirschowitz. *Topological Observations on Multiplicative Additive Linear Logic*. 2008. `http://arxiv.org/abs/0807.2636`.
[34] André Hirschowitz, Michel Hirschowitz et Tom Hirschowitz. Contraction-Free Proofs and Finitary Games for Linear Logic. In: *MFPS*, volume 249 of Electronic Notes in Computer Science, pages 287-305. Elsevier, 2009.
[35] Tom Hirschowitz et Damien Pous. Innocent Strategies as Presheaves and Interactive Equivalences for CCS. *Scientific Annals of Computer Science*, 22(1):147-199, 2012.
[36] Martin Hofmann et Thomas Streicher. The Groupoid Model Refutes Uniqueness of Identity Proofs. In: *LICS*, pages 208-212. 1994.
[37] Mark Hovey. *Model Categories*. American Mathematical Society, 1999.

[38] J. M. E. Hyland et C.-H. Luke Ong. On Full Abstraction for PCF: I, II, and III. *Inf. Comput.*, 163(2):285-408, 2000.
[39] Ole H. Jensen et Robin Milner. *Bigraphs and mobile processes (revised)*. 2004.
[40] André Joyal. Foncteurs analytiques et espèces de structure. In: *Combinatoire énumérative (Montréal 1985)*, volume 1234 of Lecture Notes in Mathematics, pages 126-159. Springer, 1986.
[41] André Joyal, Mogens Nielsen et Glynn Winskel. Bisimulation and open maps. In: *LICS*, pages 418-427. 1993.
[42] C. Kapulkin, Peter LeFanu Lumsdaine et Vladimir Voevodsky. The Simplicial Model of Univalent Foundations. *ArXiv e-prints*, 2012.
[43] G. M. Kelly. *Basic concepts of enriched category theory*. 1982.
[44] G. M. Kelly. Elementary observations on 2-categorical limits. *Bulletin of the Australian Mathematical Society*, 39:301-317, 1989.
[45] Stephen Lack. Note on the construction of free monoids. *Applied Categorical Structures*, 18(1):17-29, 2010.
[46] Stephen Lack et Ross Street. Skew monoidales, skew warpings, and quantum categories. *Theory and Applications of Categories*, 26(15):385-402, 2012.
[47] Yves Lafont. Interaction Nets. In: *POPL*. ACM, 1990.
[48] Joachim Lambek. *Deductive systems and categories II: standard constructions and closed categories*. 1969.
[49] F. William Lawvere. *Functorial semantics of algebraic theories*. Thèse de doctorat, Columbia University, 1963.
[50] Michel Lazard. Lois de groupes et analyseurs. *Annales scientifiques de l'Ecole Normale Supérieure*, 72(4):299-400, 1955.
[51] Tom Leinster. *Higher Operads, Higher Categories*. Cambridge University Press, 2004.
[52] Tom Leinster. *Basic Category Theory*. Cambridge University Press, 2014.
[53] Les Inconnus. *Les flics*. 1990.
[54] F. E. J. Linton. Some aspects of equational theories. In: *Proc. Conf. on Categorical Algebra at La Jolla*. 1966.
[55] Saunders MacLane. *Categories for the Working Mathematician*. Springer, 1998.
[56] Saunders MacLane et Ieke Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer, 1992.
[57] Robin Milner. *A Calculus of Communicating Systems*. Springer, 1980.
[58] Robin Milner, Joachim Parrow et David Walker. A Calculus of Mobile Processes, I/II. *Information and Computation*, 100(1):1-77, 1992.
[59] Andrzej S. Murawski et Nikos Tzevelekos. Game Semantics for Good General References. In: *LICS*, pages 75-84. IEEE, 2011.
[60] V. Natarajan et Rance Cleaveland. Divergence and Fair Testing. In: *ICALP*, pages 648-659. Springer, 1995.

[61] Hanno Nickau. Hereditarily Sequential Functionals. In: *Logical Foundations of Computer Science*, volume 813 of Lecture Notes in Computer Science, pages 253-264. Springer, 1994.

[62] Tobias Nipkow. Higher-Order Critical Pairs. In: *LICS*, pages 342-349. IEEE, 1991.

[63] Robert Paré. Yoneda theory for double categories. *Theory and Applications of Categories*, 25(17):436-489, 2011.

[64] Damien Pous et Davide Sangiorgi. *Enhancements of the bisimulation proof method*. 2011.

[65] A. John Power. An abstract formulation for rewrite systems. In: *CTCS*, volume 389 of Lecture Notes in Computer Science, pages 300-312. Springer, 1989.

[66] Emily Riehl. *Categorical Homotopy Theory*. Cambridge University Press, 2014.

[67] D. E. Rydeheard et J. G. Stell. Foundations of equational deduction: A categorical treatment of equational proofs and unification algorithms. In: *CTCS*, volume 283 of Lecture Notes in Computer Science, pages 114-139. Springer, 1987.

[68] Vladimiro Sassone et Paweł Sobociński. Deriving Bisimulation Congruences Using 2-categories. *Nordic Journal of Computing*, 10(2), 2003.

[69] Peter Sewell. From Rewrite Rules to Bisimulation Congruences. In: *CONCUR*, volume 1466 of Lecture Notes in Computer Science, pages 269-284. Springer, 1998.

[70] Sam Staton. General Structural Operational Semantics through Categorical Logic. In: *LICS*, pages 166-177. IEEE, 2008.

[71] M. E. Szabo. Polycategories. *Communications in Algebra*, 3(8):663-689, 1975.

[72] The Stacks Project Authors. *The Stacks Project*. 2015.

[73] Takeshi Tsukada et C.-H. Luke Ong. Nondeterminism in Game Semantics via Sheaves. In: *LICS*. IEEE, 2015.

[74] Daniele Turi et Gordon D. Plotkin. Towards a Mathematical Operational Semantics. In: *LICS*, pages 280-291. 1997.

[75] Angelo Vistoli. *Notes on Grothendieck topologies, fibered categories and descent theory*. 2007.

[76] Mark Weber. Generic morphisms, parametric representations and weakly cartesian monads. *Theory and Applications of Categories*, 13:191-234, 2004.

[77] Fabio Zanasi. *Interacting Hopf algebras — the theory of linear systems*. Thèse de doctorat, Ecole Normale Supérieure de Lyon, 2015.