DIAGRAMMES

A. BURRONI

Automates et grammaires polygraphiques

Diagrammes, tome S67-68 (2012), p. 9-32

<http://www.numdam.org/item?id=DIA_2012__S67-68__9_0>

© Université Paris 7, UER math., 2012, tous droits réservés.

L'accès aux archives de la revue « Diagrammes » implique l'accord avec les conditions générales d'utilisation (http://www.numdam.org/legal.php). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
http://www.numdam.org/

DIAGRAMMES , SUPPLEMENT AUX VOLUMES 67 + 68 , 2012, PARIS, pp. 9-32

AUTOMATES ET GRAMMAIRES POLYGRAPHIQUES

A. BURRONI

EDITE PAR L. COPPEY , F. FOLTZ , R. GUITART, C. HENRY , C. LAIR

Automates et grammaires polygraphiques

Albert Burroni

En hommage à Andrée Ehresmann à l'occasion de ses 75 ans

Sommaire

- Introduction
- En guise de prolégomènes
- 1. Catégories globulaires monoïdales
- 2. Polygraphes
- 3. transiteurs
- 4. Plexes
- 5. Schéma d'automates et automates n-D
- 6. Schéma de grammaires et grammaires n-D
- Remerciements
- Références

Introduction

Le but de ce travail n'est pas d'introduire le langage et la théorie des catégories en informatique, ceci a été réalisé sous diverses formes et dans différentes branches de cette discipline depuis longtemps, il s'agit plutôt d'une incitation à utiliser les (petites) catégories en dimensions supérieures comme unification des structures de calculs en informatique et de leur permettre de trouver des généralisations en dimensions supérieures. Ces (petites) catégories généralisant le rôle classique des monoïdes en informatique. Son but n'est pas non plus de développer de façon étendue ce qu'annonce le titre, mais seulement d'assurer quelques points d'ancrage à ce thème.

Cet article commence par un prologue qui explique en termes personnels et épistémologiques les motivations générales d'un projet : donner une description unifiée et géométrique des structures élémentaires[1] de calcul de l'informatique théorique (réécriture de mots, réécriture de termes, automates finis, automates à pile, automates d'arbres, machines de Turing, réseaux de Petri, *etc.*) et, surtout, de poser les bases d'une généralisation de ces structures en dimensions supérieures. Nous nous limitons ici, d'une part, à donner quelques définitions générales, à introduire quelques éléments techniques utiles à la réalisation de ce projet et, d'autre part, à quelques indications brèves.

[1]Nous disons "élémentaire" parce que nous n'abordons que des structures à "effet local". Ainsi, dans ce premier travail, sont absents de cette analyse la réécriture en lambda-calcul, les automates cellulaires, les réseaux d'automates et, de façon générale, le parallélisme et la concurrence — encore qu'il y ait une forme élémentaire de parallélisme dès la dimension 2 de nos automates

La section 1 rappelle la définition des ω-catégories, appelées "catégories globulaires monoïdales" lorsque cette structure est réalisée dans les catégories au lieu de l'être dans les ensembles. Du moins, il s'agit de la version stricte, car la définition générale introduite par Batanin dans [Bat 98], est passablement complexe, mais heureusement équivalente à la version stricte utlilsée ici. Le modèle principal inspirateur de cette structure est celle formée par les n-spans et leurs morphismes.

La section 2 est consacrée à la définition des polygraphes. Ceux-ci, sous le nom de "computads" ont été introduits, en dimension 2 par R. Street [Str 78], puis généralisés en toute dimension par J. Power [Pow 91]. De manière indépendante ils ont été introduits dans [Bu 93] (avec une première version [Bu 91]) et ont fait l'objet de conférences en particulier dans les congrès suivants : [Bu 87], [Bu 89], [Bu 89']. Notre motivation était d'énoncer un théorème qui montre que la réécriture de termes est une réécriture 2-dimensionnelle en un sens précis qui peut se prolonger en toutes dimensions.

la section 3 introduit des structures dérivées de la précédente, les transiteurs[2]. On pourrait les décrire informellement comme des "polygraphes à bords". Ils sont définis comme des cospans de polygraphes (avec une filtration dimensionnelle). Ces structures sont l'élément clé de notre projet de revisitation des structures de l'informatique théorique.

la section 4 est consacrée à une construction des plexes qui jouent le rôle de générateurs dans la catégorie des polygraphes, mais qui donnent aussi, en les composants et sous le nom de polyplexes, les formes de "mots" dans la théorie de la réécriture de même que dans celle des automates et des grammaires, tout cela en dimensions supérieures.

Les sections 5 et 6 donnent, respectivement, une définition d'automates et de grammaires formels (ainsi qu'une relation générale entre ces deux concepts). Ces définitions sont réduites à un noyau conceptuel très général. En implémentant ces données abstraites dans des catégories de transiteurs, et par des données concrètes de transiteurs, on obtient des automates et des grammaires polygraphiques, On trouvera plus d'indications sur des exemples dans des notes manuscrites d'un exposé fait en 2002 [Bu 02]

En guise de prolégomènes

Dans ce travail, je propose trois objectifs interdépendants : unifier, géométriser, généraliser — spécifiquement, généraliser aux dimensions supérieures les structures de calcul classiques de l'informatique théorique (IT). Ces objectifs sont dans la suite naturelle de mon article [Bu 93] qui proposait à l'époque les mêmes objectifs, mais limités à la réécriture. On trouvera ici une mise en perspective de ce programme.

J'aime mieux le dire d'emblée, les réflexions qui suivent sont celles d'un admirateur de Nicolas Bourbaki sur au moins un point fondamental : sa position structuraliste sur l'organisation des mathématiques. Cette admiration, par la suite — suite qui commence avec l'enseignement de mon professeur, Charles Ehresmann —, s'est muée en admiration pour la théorie des catégories. Et si une rupture (et pas seulement une simple extention) m'est apparue entre ces deux points de vue, ce n'est pas avec la soudaineté d'une révélation, mais au fil du temps. Reste que ce choix fondamental de Bourbaki a facilité pour d'autres mathématiciens (d'ailleurs, pas si "autres" que cela) l'ouverture d'un nouvel horizon. Les propos sur l'IT qui suivent, et qui seront limités à des aspects élémentaires (et anciens) de cette discipline, se situent à ce niveau de réflexion.

[2]Que j'ai appelé "logographes" dans mes exposés antérieurs, notamment dans [Bu 02].

Dans [Bu 93] je proposais une généralisation en dimensions supérieures de la réécriture. Il s'agissait essentiellement, à partir d'une présentation par générateurs et relations d'une n-catégorie (ce que je nommais un n-polygraphe) du calcul des n-cellules de cette n-catégorie. A l'origine de cette idée se trouve, dans l'article cité, un théorème, assez inattendu, qui montre que la logique équationnelle, ou réécriture de termes est, au sens qui vient d'être dit, un problème de réécriture 2-dimensionnelle. Autrement dit, alors que la réécriture de mots est une réécriture 1-dimensionnelle dans les monoïdes, la réécriture de termes est une réécriture 2-dimensionnelle dans les 2-monoïdes (*i.e.* catégories monoïdales strictes). Comme la réécriture est la mère (voir la grand-mère, puisqu'elle est apparue bien avant l'IT, par exemple chez Thue ou indépendamment chez Dehn) de tout les processus de calcul, il était naturel de rechercher quels liens pouvaient entretenir ces processus de l'IT avec la structure de n-catégorie (à peine plus générale que celle de n-monoïde, laquelle ne se caractérise que par le fait de n'avoir qu'une seule 0-cellule).

Quant aux liens entre l'IT et la structure de monoïde (ou celle de groupe), ils sont anciens, étendus et parfaitement bien connus, non seulement en réécriture, mais aussi dans l'étude des langages. En revanche la structure de catégorie, elle, pourtant si proche puisqu'on aurait pu l'appeler structure de "monoïde à plusieurs objets"[3], est restée généralement ignorée. Du moins, c'est ainsi qu'en surface les choses se présentent car, comme nous le verrons, la structure de catégorie, et plus largement celle de n-catégorie, est la partie cachée d'un iceberg dont la partie visible est la structure de monoïde. En réalité, concrètement ou potentiellement, la structure de catégorie est omniprésente en IT, même si sa présence est masquée. On peut même se risquer à dire plus : il faut faire un effort, conscient ou non, pour ne pas la voir. Certes, de nombreux auteurs ont œuvré pour introduire cette structure dans la théorie des automates, mais outre que, à notre connaissance, ces travaux ont un caractère sporadique et/ou généraliste, il faut dire que ces travaux restent en marge du corpus d'études concrètes faites par les informaticiens (parmi les catégoriciens ayant introduit la structure de catégorie dans la théorie des automates, on peut citer, en particulier — et avec sûrement beaucoup d'oublis[4] —, Adamek, Bozapalidès, Ehrig, Eilenberg, Guitart, Hoehnke, Latch, Lawvere, Manes, Riguet, Trnková, Walters, ...).

Il semblera, à beaucoup, que ces affirmations, sur l'absence ou la présence de la structure de catégorie en IT, soient démesurément inexactes. Inexactes, parce que la théorie des catégories et son langage ont été largement adoptés dans divers secteurs importants de l'IT. Cependant, on aura sans doute remarqué que, jusqu'ici, nous n'avions pas employé le terme de "théorie des catégories", mais toujours celui de "structure de catégorie". Cette expression de "théorie des catégories" est, en effet, trompeuse si l'on entend par théorie des catégories l'étude, au sens propre, de la structure de catégorie. Ainsi, si l'on compare cette appellation, par exemple, avec celle de théorie des monoïdes, laquelle est vraiment l'étude de la structure de monoïde, l'expression est inadaptée. La théorie des catégories a bien d'autres choses à faire : par exemple, connaît-on un seul théorème fondamental de structure sur les catégories ? (Toujours par exemple — pour dire quelque chose de fantaisiste au hasard —, a-t-on une étude connue de la décomposition d'une catégorie en produit d'éléments simples — que cela ait ou non un intérêt ?) Ce qui est étudié en théorie des catégories ce sont des structures bien plus complexes que de simples catégories (*i.e.* monoïdes à plusieurs objets),

[3] J'aurai envie de faire des remarque analogue pour la linguistique ; cette idée à joué un rôle dans mes choix conceptuels généraux.

[4] Pour comprendre le choix des auteurs figurant dans cette liste, voir ci-dessous la distinction que je fais entre cette démarche et celle qui consiste à introduire le « langage » des catégories.

même si au hasard des nécessités, ici ou là, cela puisse arriver. Mais j'y reviendrai plus loin à propos du concept de structure.

Mais comment, se demandera mon lecteur, peut-on dire que certains informaticiens théoriciens ont adopté la théorie des catégories et son langage sans avoir, par là même, intégré la structure de catégorie ? C'est là un paradoxe apparent que je ne pourrai éclairer que plus loin, après la mise en place de quelques idées générales.

C'est durant une période relativement courte, mais riche, disons entre l'introduction de la machine de Turing (1936) et celle des réseaux de Petri (1962) que l'IT (qui aurait pu s'appeler "science du calcul") a créé, pour son propre usage, une multitude de structures qui ont toutes pour fonction de produire ou d'analyser du calcul. Cette série de structures, qu'on peut qualifier de "machines à calculer formelles" ont pour représentants, entre autres, les automates finis, les automates à pile, les machines de Turing, les réseaux de Petri, les grammaires, *etc.* ; liste qui n'est pas exhaustive. Ainsi, par exemple, on peut leur adjoindre une quantité de jeux combinatoires à un ou plusieurs joueurs (taquin, Rubik's cube, échecs, ...) qui sont aussi des formes de calcul. Je n'oublie pas non plus la forme la plus ancestrale, si l'on peut dire, de toutes ces machines qui est tout simplement la réécriture dont j'ai parlé au début. C'est cette dernière que l'on retrouvera, sous des masques plus ou moins pittoresques, dans la variété un peu hétéroclite de toutes les autres. C'est donc bien dans la continuité de [Bu 93] qu'est écrit cet article.

Ce qui frappe, à l'examen de ces structures, c'est la différence *implicite* du statut qui leur est attaché par rapport à celui dont bénéficient des structures mathématiques classiques telles que les groupes, les corps, les espaces topologiques et, plus proche de ce travail, les (petites) catégories. Les premières ne sont jamais vraiment traitées pour elles-mêmes, ce sont des auxiliaires (malgré la richesse d'idées et d'intuition qui font leur charme). Elles semblent créées *ad hoc* pour une fonction, tandis que les structures classiques ont plus d'universalité. Autrement dit, les unes ressemblent à ce qu'en ingénierie on appelle des modèles, tandis que les autres sont, disons, des "idées". Mais nous découvrirons plus loin, pour ces dernières, le "secret" qui explique l'une des raisons profondes de leur dynamisme et de leur universalité.

A vrai dire, de cet aspect *ad hoc* des structures de l'IT, on ne devrait pas s'étonner. L'informatique est, de par sa nature, comme de par ses origines historiques, une mathématique appliquée, car, bien que le concept de calcul soit consubstantiel à celui de mathématique, le calcul est du côté de l'action et pas nécessairement du côté du contenu. Toutefois, les mathématiques ont une propriété de réflexivité qui, dans la durée, finit par absorber l'acteur ou, plus simplement son action, pour le faire entrer dans le cercle de ce qui est étudié. Si le calcul est un acte, le calcul sur le calcul devient à son tour un nouveau contenu mathématique.

Que penser alors de cet aphorisme célèbre : "*l'informatique théorique n'est pas plus l'étude des ordinateurs que l'astronomie n'est l'étude des télescopes*"[5] ? Ne milite-t-il pas dans le sens d'une plus grande théorisation de l'IT ? L'interrogation exprimée par cet aphorisme, a certainement une visée assez juste, on voit bien ce qu'il veut dire, quoique ... car, tout-à-coup, un doute nous vient à l'esprit : cet aphorisme est-il si pertinent qu'il le semble à première vue ? au delà de son évidence immédiate, n'avoue-t-il pas autre chose ... son contraire ? Et si l'erreur de l'ID, curieusement assumée par cette phrase, était d'avoir négligé ses "télescopes de l'esprit" ? je veux dire ses machines à calculer formelles dont il est question plus haut ?

En fait, tout se passe comme si une bipolarisation conceptuelle hétérogène (refusant la réflexivité) s'était installée entre machine et calcul, entre l'acte du calcul et le contenu du

[5] Aphorisme dont les idées remontent, je crois, à Hal Abelson

calcul, un peu à l'instar de la métaphysique qui distingue le corps et l'esprit[6]. Je voudrais défendre ici un point de vue diamétralement opposé : calculs et machines sont structurellement de nature identique. Cette thèse est l'enjeu de ce travail. Mais avant d'entrer dans l'argumentation qui convient, j'ai besoin d'aller plus avant dans la réflexion sur le concept de structure.

Dans les années 30, le concept de structure semble parvenir doublement au statut prestigieux de paradigme. A la fois chez Bourbaki comme concept organisateur de disciplines mathématiques et, à la même époque ou, plutôt, un peu avant, chez les logiciens, sous la forme d'un concept, frère siamois du concept bourbakiste, celui de "modèle" — ce dernier n'ayant pas de parenté sérieuse avec son homonyme en ingénierie. Mais la théorie des modèles s'est montrée plus pérenne et plus en équilibre avec son concept que Bourbaki avec le sien. Peut-être parce que les logiciens ont pensé leur concept avec plus de cohérence ; et la séparation des siamois, si l'on peut dire, s'est faite avec succès pour l'un d'eux.

Le problème pour Bourbaki, c'est qu'un paradigme peut en cacher un autre. Peu de temps après les structures, sont arrivées les catégories. Une catégorie est aussi une structure, mais celles dont il était question, étaient d'emblée, en quelque sorte, des "superstructures". Précisément, c'étaient des catégories dont les éléments (les objets) étaient eux-mêmes des structures. Le calcul n'était plus seulement un calcul des éléments *dans une structure* (comme en arithmétique où l'on calcule des nombres dans la structure de l'anneau des entiers $\mathbb{Z}$), mais un calcul de structures *dans une catégorie* (calcul de produits cartésiens, de quotients, *etc, etc.*). La catégorie qui est à la fois emblématique et, en quelque sorte "génératrice" de ce nouveau monde, est la catégorie des ensembles **Ens**. Ce que l'on appelle théorie des catégories, c'est l'étude de ces "monstres" structurels. Ces "grandes" catégories sont en fait beaucoup plus que des catégories : il y a, dans ces superstructures, sous forme de propriétés (qui sont souvent des données déguisées en propriétés), d'autres ambitions que de généraliser la théorie des monoïdes. Une telle théorie aurait pu tout aussi bien s'appeler théorie des structures ou, montant d'un cran dans l'universalité, théorie des univers. Mais ici nous n'irons pas plus loin dans cette analyse qui met en jeu la réflexivité fondamentale des mathématiques et qui risque de nous entrainer un peu hors de notre sujet.

Donc, le nom de théorie des catégories aurait pu avoir un autre contenu, aller vers d'autres projets, comme celui, par exemple, dont je parlais plus haut : une extension de la théorie des monoïdes, ou mieux, de celle des groupoïdes qui sont présents en topologie algébrique, mais qui le sont aussi en géométrie différentielle, domaine dans lequel Ehresmann a montré leur importance fondamentale.

Mais, maintenant il faut le dire, cette dernière proposition d'une théorie des "petites" catégories est à la fois tenable et intenable. D'abord, elle est trop simpliste : très vite, à cause d'un modeste petit lemme, le lemme de Yoneda, on s'est aperçu qu'elle ne permettait pas sérieusement de séparer "grandes" et "petites" catégories : toute petite catégorie C se plonge dans une grande catégorie $\hat{C} = \mathbf{Ens}^{C^{\mathrm{op}}}$ (celle de ses préfaisceaux), et c'est dans cette grande structure que l'on étudie vraiment au mieux la petite. D'ailleurs ceci n'est pas propre aux catégories, beaucoup d'autres petites structures se traduisent par des moyens similaires en terme de grandes catégories. Mais, d'un autre coté, cela n'efface pas la différence, je veux dire la vraie différence qui, en réalité, n'est pas une question de taille. Je me limiterai à un seul exemple. Si C est une "petite" catégorie, sa connexité, par exemple, se définit ainsi : C n'est pas vide et n'est pas un coproduit de deux catégories non vides. Par contre, la même propriété exprimée sur la "grande" catégorie de ses préfaisceaux $\hat{C}$ s'exprime en de tout

[6]Ceci me sert de métaphore ; ce n'est pas une opinion que j'exprime.

autres termes, bien plus complexes (par ailleurs, $\hat{C}$ est *toujours* connexe selon la première définition), alors que les deux définitions soient équivalentes.

Revenons à la comparaison entre structures classiques (groupes, espaces topologiques, *etc.*) et structures de calcul de l'IT. L'élément clé qui peut nous permettre de comprendre la différence, le "secret " que j'attribuais aux premières et qui explique leurs caractéristiques, est que la notion de structure n'existe qu'à travers un autre concept, celui d'homomorphisme entre structures. Du moins c'est ce qu'apporte de radicalement nouveau, par rapport à Boubarki, la théorie des catégories (expression que nous emploierons désormais au sens habituel). Les structures ne sont que l'élément statique d'un tout où l'élément dynamique est donné par le concept d'homomorphisme entre structures. Les structures les plus anciennes et les plus importantes sont souvent issues secondairement de leurs homomorphismes clairement définis en premiers ! Ainsi, de la continuité est née la structure de topologie, de celle de fonctions différentiables est née celle de variété différentielle, de la notion de représentation est née celle de groupe, *etc.*, mais le plus bel exemple nous est donné par les catégories elles-mêmes, elles n'ont été introduites, au dire de ses concepteurs, que de la nécessité du concept de foncteur et, mieux, ce dernier lui-même, de la nécessité d'introduire la notion de transformation naturelle (homomorphisme d'homomorphismes !). Et quand une "même" structure possède la possibilité de deux notions différentes d'homomorphismes, alors on doit considérer qu'on à affaire à deux structures vraiment distinctes.

Cette dynamique (celle des homomorphismes qui sont des "flèches") se retrouve pleinement dans la notion de calcul. Il y a plus qu'une simple analogie entre calcul et dynamique, entre systèmes formels et systèmes dynamiques. Techniquement, en dehors de finalités différentes qui leur sont demandées, une différence serait que les premiers sont "plutôt" discrets tandis que les seconds sont "plutôt" continus[7]. Et on peut être surpris que cette forte parenté ne se traduise pas pleinement dans les structures de calcul de l'IT, excepté les cas les plus élémentaires, ceux où les monoïdes jouent un rôle important. Or il nous semble que l'IT est la discipline dans laquelle, plus que dans toute autre, une dynamique devrait apparaître dans les structures et cela à deux niveaux qui ne sont pas totalement indépendants. Le premier de ces niveaux est que ces structures devraient être pourvues d'homomorphismes avec toutes les conséquences que cela pourrait avoir (existence de monomorphismes, d'épimorphismes, de produits cartésiens, *etc.*). Le second est interne à ces structures, elles devraient, de par leur dynamique interne, être très proches de la structure des (petites) catégories — et en toutes dimensions.

Il me reste à éclaircir une contradiction apparente, celle où je supposais qu'un lecteur s'étonne de ce que, malgré le fait que la théorie et le langage des catégories soient adoptés largement dans certains secteurs de l'informatique, l'introduction de la *structure* de catégorie, au sens justement de "petites" catégories, ne le soit pas véritablement ; je veux dire, concrètement, autrement que noyée dans de vastes généralités. En fait, les milieux informatiques sont divers. En fréquentant un peu certain d'entre-eux, j'ai eu l'impression d'une dichotomie entre deux tendances. Pour faire image et pour faire bref, disons qu'il y avait ceux qui évoluaient plutôt dans les langages de programmation déclaratifs et ceux qui évoluaient dans les langages impératifs. Les premiers seraient plutôt des "logiciens" (lambda-calcul, logique linéaire, ...) et ceux-la ont adopté la théorie des catégories via de "grandes" catégories comme les catégories cartésiennes fermées. Les seconds, je les appellerais plutôt "algébristes" (étude des monoïdes, des semi-anneaux, ...). Mes propos sur l'intégration des "petites" catégories

[7]On pourrait aussi évoquer des différences telles que, être ou ne pas être un système déterministe ou un système fini.

dans le paysage de l'IT, concerneraient essentiellement ces derniers.

Unifier, généraliser, sont deux activités en relation dialectique. Unifier pour généraliser, généraliser pour unifier. Les idées proposées dans ce papier doivent être pensées sous cet angle. Essayons d'en tirer une sorte de programme :

- Etendre le rôle qu'a joué pendant longtemps la structure de monoïde comme outil de base d'une grande partie de l'IT (études des langages, problème du mot, ...) en lui substituant celle, plus générale, de catégorie n-dimensionnelle (pour $n \in \mathbb{N} \cup \{\omega\}$).
- Mettre en évidence l'unité fondamentale des structures de calcul de l'IT : machines et grammaires (ce sont essentiellement des présentations "par générateurs et relations" de n-catégories).
- Géométriser le concept de calcul qui est un cousin "discret" des systèmes dynamiques.
- Introduire une classification de ces structures de calcul en fonction d'un paramètre dimensionnel (le "n" du préfixe dans "n-catégories"), puis créer de nouvelles structures de calcul (par exemple, des langages d'"images" en dimensions supérieures).
- Donner un statut mathématique clair à ces structures de calcul et les rendre aussi universelles que les structures classiques des mathématiques dites "pures".
- Enfin, une simple remarque : la structure de n-catégorie est essentiellement une structures "algébrico-géométrico-combinatoire", ce qui explique le rôle fondamental qu'elle peut jouer en IT, théorie qui a tant d'affinité avec celle des systèmes dynamiques.

1 Catégories globulaires monoïdale

1.1 ω-Graphes catégoriques

Un *n-graphe*, encore appelé *ensemble n-globulaire*, est un diagramme dans les ensembles de la forme :

$$G_0 \underset{\mathrm{t}_0}{\overset{\mathrm{s}_0}{\leftleftarrows}} G_1 \underset{\mathrm{t}_1}{\overset{\mathrm{s}_1}{\leftleftarrows}} G_2 \quad \cdots \quad G_{n-1} \underset{\mathrm{t}_{n-1}}{\overset{\mathrm{s}_{n-1}}{\leftleftarrows}} G_n,$$

et vérifiant, pour tout $2 \leq i \leq n$, les conditions : $\mathrm{s}_{i-2}\mathrm{s}_{i-1} = \mathrm{s}_{i-2}\mathrm{t}_{i-1}$ et $\mathrm{t}_{i-2}\mathrm{s}_{i-1} = \mathrm{t}_{i-2}\mathrm{t}_{i-1}$.

Pour tout $0 \leq j < i \leq n$, on posera $\mathrm{s}^i_j = \mathrm{s}_j\mathrm{s}_{j+1}\cdots\mathrm{s}_{i-1}$ et $\mathrm{t}^i_j = \mathrm{t}_j\mathrm{t}_{j+1}\cdots\mathrm{t}_{i-1}$ (en particulier $\mathrm{s}_{i-1} = \mathrm{s}^i_{i-1}$, $\mathrm{t}_{i-1} = \mathrm{t}^i_{i-1}$).

On appelle *n-graphe catégorique* ou *catégorie n-globulaire* un diagramme similaire, mais réalisé dans les catégories.

On étend immédiatement cette définition à une structure appelée *ω-graphe*, ou ensemble ω-globulaire, lorsque le diagramme précédent est illimité sur sa droite. On posera $\overline{\mathbb{N}} = \mathbb{N} \cup \{\omega\}$ et, si rien n'est spécifié, quand on parlera de n-graphe, il s'agira du cas où $n \in \overline{\mathbb{N}}$. On note $\mathbf{Grph}_n$ la catégorie qui a pour objets ces structures et dont les morphismes sont les morphismes naturels.

Remarque 1.1. La double terminologie employée ci-dessus mérite une explication : comme il est bien connu, les catégories $\mathbf{Grph}_n(\mathbf{Cat})$ et $\mathbf{Cat}(\mathbf{Grph}_n)$ sont équivalentes, cela conduit aux terminologies équivalentes de *n-graphe catégorique* ou *catégorie n-graphique*. Mais celle utilisée ici, *catégorie n-globulaire*, est due à Batanin et procède d'une autre règle syntaxique. Voir une suite de cette remarque au paragraphe suivant (1.2).

1.2 ω-Catégories catégoriques

Pour simplifier les notations dans la définition qui va suivre, nous adopterons, pour certains produits fibrés, les conventions suivantes : dans une catégorie admettant des produits fibrés, si C est un n-graphe, pour tout $0 \leq j < i \leq n$, on notera $C_i \times_j C_i$ le produit fibré qui a pour objets les couples $(y, x) \in C_i \times C_i$ tels que $\mathrm{s}^i_j(y) = \mathrm{t}^i_j(x)$. De même, $C_i \times_j C_i \times_j C_i$ note l'ensemble des (x, y, z) tels que $\mathrm{s}_j x = \mathrm{t}_j y$ et $\mathrm{s}_j y = \mathrm{t}_j z$ et, pour $0 \leq k < i < j \leq n$, et de manière analogue $(C_i \times_k C_i) \times_j (C_i \times_k C_i)$, note l'ensemble des (x, y, x', y') tels que $\mathrm{s}_k x = \mathrm{t}_k y, \mathrm{s}_k x' = \mathrm{t}_k y', \mathrm{s}_j x = \mathrm{t}_j x'$ (d'où il découle que $\mathrm{s}_j y = \mathrm{t}_j y'$).

Une *n-catégorie* C est définie par la liste des données et des conditions suivantes :

- Un n-graphe noté le plus souvent par le même symbole C que la n-catégorie.

Pour tout couple d'entiers tels que $0 \leq j < i \leq n$, on a :

- Une opération $*^i_j : C_i \times_j C_i \longrightarrow C_i$ appelée (j, i)-*composition*. On note $y *^i_j x$ l'image de $(y, x) \in C_i \times_j C_i$, par cette application.
- Une opération $\mathrm{id}^i_j : C_j \longrightarrow C_i$ appelée application (j, i)-identité. On note $\mathrm{id}^i_j(u)$ l'image de $u \in C_j$ par cette application.

Ces données sont soumises d'abord à des conditions de "position" :

- Pour tout $(y, x) \in C_i \times_j C_i$, si $j < k < i$ on a

$$\mathrm{s}_k(y *^i_j x) = \mathrm{s}_k(y) *^i_j \mathrm{s}_k(x) \qquad \text{et} \qquad \mathrm{t}_k(y *^i_j x) = \mathrm{t}_k(y) *^i_j \mathrm{t}_k(y),$$

  et si $k = j$, on a

$$\mathrm{s}_j(y *^i_j x) = \mathrm{s}_j(x) \qquad \text{et} \qquad \mathrm{t}_j(y *^i_j x) = \mathrm{t}_j(y))$$

- Pour $u \in C_j$, si $j < k < i$, on a

$$\mathrm{s}_k(\mathrm{id}^i_j(u)) = \mathrm{id}^i_k(u) = \mathrm{t}_k(\mathrm{id}^i_j(u)),$$

  et si $k = j$, on a

$$\mathrm{s}_j(\mathrm{id}^i_j(u)) = u = \mathrm{t}_j(\mathrm{id}^i_j(u)).$$

Puis aux axiomes "algébriques" suivants, pour tout $j < i$:

- $(z, y, x) \in C_i \times_j C_i \times_j C_i$ on a

$$(z *^i_j y) *^i_j x = z *^i_j (y *^i_j x).$$

- $x \in C_i$ tel que $u = \mathrm{s}^i_j(x)$ et $v = \mathrm{t}^i_j(x)$, on a

$$\mathrm{id}^i_j(v) *^i_j x = x = x *^i_j \mathrm{id}^i_j(u).$$

Et, pour tout $j < k < i$:

- $(y', x', y, x) \in (C_i \times_j C_i) \times_k (C_i \times_j C_i)$, on a

$$(y' *^i_k x') *^i_j (y *^i_k x) = (y' *^i_j y) *^i_k (x' *^i_j x).$$

- $(v, u) \in C_k *_j C_k$, on a

$$\mathrm{id}^i_k(v *^k_j u) = \mathrm{id}^i_k v *^i_j \mathrm{id}^i_k u.$$

- $e \in C_j$, on a

$$\mathrm{id}^i_k \mathrm{id}^k_j e = \mathrm{id}^i_j e.$$

- $e \in C_j$, on a
$$\mathrm{id}^i_j e *_k \mathrm{id}^i_j e = \mathrm{id}^i_j e.$$

Dans la pratique, on simplifie ces notations et, par exemple, on note souvent $y *_j x$ au lieu de $y *^i_j x$, $\mathrm{id}^j u$ au lieu de $\mathrm{id}^j_i u$ et $x *_i y *_i z$ au lieu de $(x *_i y) *_i z$.

Comme pour le cas graphique, on étend cette définition en une notion de ω-catégorie. Pour cela il suffit de l'appliquer à un ω-graphe, nous laissons au lecteur le soin d'en fixer les détails. Et, si rien n'est spécifié, quand on parlera de n-catégrie, il s'agira du cas où $n \in \overline{\mathbb{N}}$.

On laisse au lecteur le rappel de la notion de n-foncteur. Les n-catégories forment les objets d'une catégorie $\mathbf{Cat}_n$ (on sait aussi que l'on a une structure de $(n+1)$-catégorie sur $\mathbf{Cat}_n$). Pour tout $n \in \mathbb{N}$ on a un foncteurs de restriction $S_n : \mathbf{Cat}_{n+1} \longrightarrow \mathbf{Cat}_n$ et un foncteur d'oubli $U_n : \mathbf{Cat}_n \longrightarrow \mathbf{Grph}_n$.

Si maintenant on réalise cette structure de n-catégorie, non plus dans **Ens**, mais dans **Cat**, alors selon les remarques faites en (1.1) on devrait appeler *n-catégorie catégorique* ou *catégorie n-catégorique*. Cependant, ici encore, nous suivrons Batanin en la nommant *catégorie n-globulaire monoïdale stricte* ou plus simplement *catégorie n-globulaire monoïdale*, omettant l'adjectif "stricte". Voici, en plus du plaisir de simplifier la terminologie, notre justification (qui peut être largement être contestée) : Batanin, dans [Bat 98], sous ce nom de *catégorie n-globulaire monoïdale*, défini une version "faible" de la structure que nous pourrions appeler, comme indiqué ci-dessus, *n-catégorie catégorique*. Il démontre ensuite que toute *catégorie n-globulaire monoïdale*, donc en version "faible", est équivalente à une *catégorie n-globulaire monoïdale stricte*. Cette équivalence est l'analogue de l'équivalence entre les bicatégories de Benabou et les 2-catégories, ce qui veut dire que, jusqu'à un certain point, l'on ne commet pas d'erreur fondamentale en traitant les structures "faibles" de Batanin comme si elles étaient strictes.

Remarque 1.2. Par ailleurs, si la définition de cette structure, donnée par Batanin, est est acceptable d'un point de vue pragmatique, n'est pas idéalement complète : il n'est tenu aucun compte des trois derniers axiomes de la liste qui définit les n-catégories, donnée au début de cette section. Or, dans une définition de catégorie globulaire monoïdale "faible" qui, selon nous, serait complète, ces axiomes ne devraient pas se réduire à des égalité et donner autant de 2-cellules de cohérence (puis d'un nombre imposant d'axiomes — ou 3-cellules — de cohérence). Sans doute en est-il ainsi, parce que dans le cas des n-spans qui est le modèle inspirateur de cette structure, ces axiomes semblent disparaîtres au regard de l'observateur tant ils sont évanescents (sans parler du caractère flou de la définition des compositions des n-spans lequel dépend d'un choix arbitraire de produits fibrés). Ils sont pourtant là, et, s'il est pratique de les négliger, il peut être intéressant également d'en tenir compte, comme nous le montrerons dans un article futur.

Les catégories n-globulaires monoïdales (strictes) forment une catégorie

$$\mathbf{Cgm}_n = \mathbf{Cat}_n(\mathbf{Cat}),$$

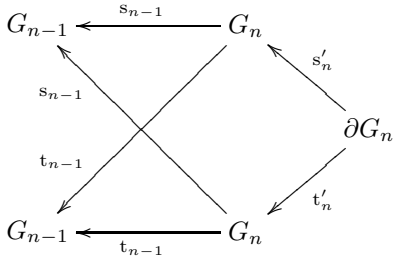
mais aussi une 2-catégorie, car elles héritent de la structure de 2-catégorie de **Cat** (les morphismes sont des transformations naturelles, les 2-cellules correspondent à des "modifications").

Notons qu'une 1-catégorie globulaire monoïdale (stricte) n'est rien d'autre qu'une "catégorie double" d'Ehresmann.

1.3 ∂-Structures

On aura besoin (pour définir la notion de polyplexes plus loin) d'une structure supplémentaire sur un ω-graphe et appelée ∂-structure.

Soit G un ω-graphe. Pour tout $n \geq 1$, on définit un ensemble limite projective ∂G_n :



On complète cette définition en posant $\partial G_1 = G_0 \times G_0$ et $\partial G_0 = 1$ (ensemble final).

Une *∂-structure* sur G est la donnée, pour tout $n \in \mathbb{N}$, d'une application $\partial_n : \partial G_n \longrightarrow G_{n+1}$ telle que $\pi\partial_n = \mathrm{id}_{G_n}$ où $\pi = [s_n, t_n]$.

On définit la même notion en remplaçant les ensembles et les applications par des catégories et des foncteurs. C'est sous cette dernière forme que nous l'utiliserons plus loin, en 4.1 et 4.2.

2 Polygraphes

2.1 Extentions polygraphiques

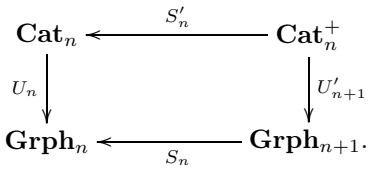
Pour tout entier n, on appellera *extension polygraphique* d'une n-catégorie (ou, *n-catégorie munie d'une extension polygraphique*) la donnée d'un quadruplet $(C, \Sigma, \sigma, \tau)$ où

- C est une n-catégorie,
- Σ est un ensemble,
- $\sigma, \tau : \Sigma \rightrightarrows c_n(C)$ deux applications telles que pour $n \geq 2$ on ait : $s_{n-1}\sigma = s_{n-1}\tau$ et $t_{n-1}\sigma = t_{n-1}\tau$,

Pour $n = 0$, une extension polygraphique d'une 0-catégorie (*i.e. un ensemble*, est la donnée d'un graphe $\sigma, \tau : \Sigma \rightrightarrows C$ puisque $C = c_0(C)$. On pourrait prolonger conventionnellement cette définition pour $n = -1$: la notion se réduirait à la notion d'ensemble.

Souvent, par abus de langage, on notera simplement (C, Σ) au lieu de $(C, \Sigma, \sigma, \tau)$.

Les n-catégories munies d'une extension polygraphique forment les objets d'une catégorie $\mathbf{Cat}_n^+$, qui peut être définie comme le produit fibré de catégories suivant :



En comparant ce carré avec le carré commutatif canonique suivant :

$$\begin{array}{ccc} \mathbf{Cat}_n & \xleftarrow{S_n} & \mathbf{Cat}_{n+1} \\ {\scriptstyle U_n}\downarrow & & \downarrow{\scriptstyle U_{n+1}} \\ \mathbf{Grph}_n & \xleftarrow[S_n]{} & \mathbf{Grph}_{n+1}, \end{array}$$

on en tire un foncteur d'oubli $W_{n+1} : \mathbf{Cat}_{n+1} \longrightarrow \mathbf{Cat}_n^+$, caractérisé par $U'_{n+1}W_{n+1} = U_{n+1}$ et $S'_{n+1}W_{n+1} = S_n$. Donc, pour toute $(n+1)$-catégorie D, on a $W_{n+1}(D) = (S_n(D), \mathrm{c}_{n+1}(D), \mathrm{s}_n, \mathrm{t}_n)$, éventuellement noté $(S_n(D), \mathrm{c}_{n+1}(D))$.

Ce foncteur admet un adjoint à gauche $K_{n+1} : \mathbf{Cat}_n^+ \longrightarrow \mathbf{Cat}_{n+1}$. (L'existence de cet adjoint résulte de l'« esquissabilité projective » de ces structures — voir, par exemple, [Bu 70].) Pour tout objet (C, Σ) de $\mathbf{Cat}_n^+$, on adoptera la notation suivante :

$$C[\Sigma] = K_{n+1}(C, \Sigma).$$

2.2 Définition des n-polygraphes

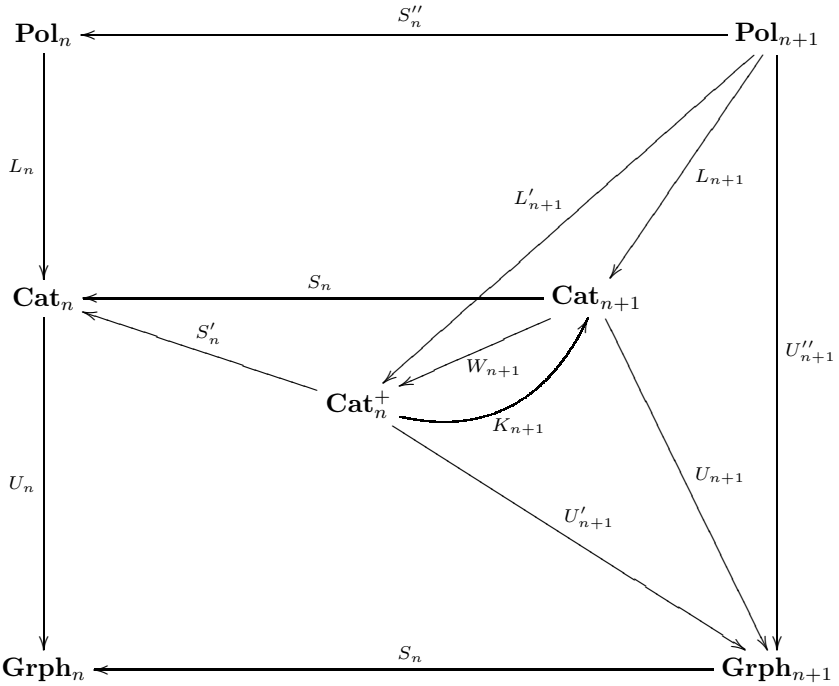
La notion qui suit et qui fait le titre de cette section est définie par induction sur $n \in \mathbb{N}$. Précisément, on va définir, récursivement sur n, à la fois une catégorie $\mathbf{Pol}_n$ et un foncteur $L_n : \mathbf{Pol}_n \longrightarrow \mathbf{Cat}_n$. Les objets de $\mathbf{Pol}_n$ sont appelés *n-polygraphes*.

Au début, on pose $\mathbf{Pol}_0 = \mathbf{Ens}$, on a alors également $\mathbf{Pol}_0 = \mathbf{Grph}_0 = \mathbf{Cat}_0$, et on peut poser $L_0 = \mathrm{id}_{\mathbf{Pol}_0} : \mathbf{Pol}_0 \longrightarrow \mathbf{Cat}_0$. Supposons définie, pour $n \in \mathbb{N}$, la catégorie $\mathbf{Pol}_n$ et le foncteur $L_n : \mathbf{Pol}_n \longrightarrow \mathbf{Cat}_n$, on forme alors le produit fibré :

$$\begin{array}{ccc} \mathbf{Pol}_n & \xleftarrow{S''_n} & \mathbf{Pol}_{n+1} \\ {\scriptstyle U_nL_n}\downarrow & & \downarrow{\scriptstyle U''_{n+1}} \\ \mathbf{Grph}_n & \xleftarrow[S_n]{} & \mathbf{Grph}_{n+1} \end{array} .$$

Ce qui définit $\mathbf{Pol}_{n+1}$. Puis, si on compare ce produit fibré au précédent qui définit $\mathbf{Cat}_n^+$, on en tire un foncteur $L'_{n+1} : \mathbf{Pol}_{n+1} \longrightarrow \mathbf{Cat}_n^+$ déterminé par les relations $U'_{n+1}L'_{n+1} = U''_{n+1}$ et $S'_{n+1}L'_{n+1} = S''_n$. On pose alors $L_{n+1} = K_{n+1}L''_{n+1}$.

L'ensemble de ces données se retrouve sur le diagramme suivant :



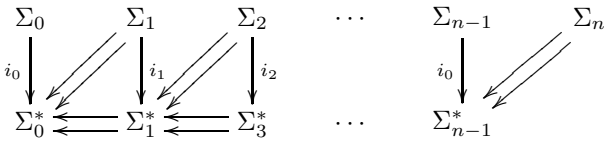
Il est clair d'après cette définition qu'on peut identifier un n-polygraphe, pour $n \geq 1$, à un quadruplet $(Q, \Sigma, \sigma, \tau)$ où Q est un $(n-1)$-polygraphe, $\sigma, \tau : \Sigma \rightrightarrows \mathrm{c}_{n-1}(L_{n-1}(Q))$ un graphe et, si $n \geq 2$, avec les conditions : $\mathrm{s}_{n-1}\sigma = \mathrm{s}_{n-1}\tau$ et $\mathrm{t}_{n-1}\sigma = \mathrm{t}_{n-1}\tau$ (et donc, $(L_n(Q), \Sigma)$ est une n-catégorie munie d'une extension polygraphique).

Par la suite, pour simplifier le formalisme, on notera $C_{-1} = \Sigma_{-1} = 1$, par convention identifiés à l'ensemble final $1 = \{0\}$, et $\sigma_{-1} = \tau_{-1} : \Sigma_0 \longrightarrow \Sigma_{-1}$ l'application canonique vers l'ensemble final. Ainsi, on peut définir : un 0-polygraphe comme un quadruplet $(C_{-1}, \Sigma_0, \sigma_{-1}, \tau_{-1})$ ou, en notation simplifiée, un couple (C_{-1}, Σ_0). Ce même couple (C_{-1}, Σ_0) désigne une (-1)-catégorie avec une extension polygraphique.

On peut également considérer un n-polygraphe comme une suite

$$((C_{-1}, \Sigma_0), (C_0, \Sigma_1), \dots (C_{n-1}, \Sigma_n))$$

où, pour tout $0 \leq i < n-1$, (C_i, Σ_{i+1}) est une i-catégorie avec une extension polygraphique, si $C_0 = \Sigma_0$, et si, pour $1 \leq i \leq n-1$, on a $C_i = C_{i-1}[\Sigma_i]$. Dans ce cas, il convient d'introduire les notations $\sigma_{i-1}, \tau_{i-1} : \Sigma_i \rightrightarrows \Sigma_{i-1}^*$, à placer sur le diagramme ci-dessous (qu'on trouve aussi dans [Bu93] et [Met03]) :



où, pour tout $0 \leq i < n$, $\Sigma_i^* = \mathrm{c}_i(C_{i-1}(\Sigma_i))$ et où les morphismes $\mathrm{s}_{i-1}, \mathrm{t}_{i-1} : \Sigma_{i-1}^* \rightrightarrows \Sigma_i^*$ prolongent les morphismes précédents, notés $\overline{\sigma}_{i-1}$ et $\overline{\tau}_{i-1}$ (ou encore, par abus de langage

σ_{i-1} et τ_{i-1}) et défini par la propriété universelle des Σ_i^*. Une façon plus linéaire et plus brève de présenter ce diagramme est la suivante :

$$\Sigma_0 \xrightarrow{i_0} \Sigma_0^* \underset{\tau_0}{\overset{\sigma_0}{\leftleftarrows}} \Sigma_1 \qquad \cdots \qquad \Sigma_{n-1} \xrightarrow{i_{n-1}} \Sigma_{n-1}^* \underset{\tau_{n-1}}{\overset{\sigma_{n-1}}{\leftleftarrows}} \Sigma_n$$

Pour un n-polygraphe P, avec ces notations, on posera $\mathrm{c}_k(P) = \Sigma_k$ et $\mathrm{c}_k^*(P) = \Sigma_k^*$.

Pour tout $n \in \mathbb{N}$, on a un foncteur pleinement fidèle et bijectifs sur les objets $I_{n+1} : \mathbf{Pol}_n \longrightarrow \mathbf{Pol}_{n+1}$ qui associe au n-polygraphe P le polygraphe $I_{n+1}(P)$ tel que $S''_n(I_{n+1}(P)) = P$ et $\mathrm{c}_{n+1}(I_{n+1}(P)) = \emptyset$. Plus généralement, pour $n \leq n'$, on a un foncteur composé $I_{n,n'} : \mathbf{Pol}_n \longrightarrow \mathbf{Pol}_{n'}$. La limite inductive de ces données forme une catégorie $\mathbf{Pol}_{(\omega)}$ (en fait, les foncteurs $I_{n,n'}$ permettent d'identifier la catégorie $\mathbf{Pol}_n$ à une sous-catégorie de $\mathbf{Pol}_{n+1}$, donc de voir $\mathbf{Pol}_{(\omega)}$ comme une réunion de ces catégories). On a également des foncteurs de restrictions $S''_{n,n'} : \mathbf{Pol}_{n+1} \longrightarrow \mathbf{Pol}_{n'}$ et on notera $\mathbf{Pol}_\omega$, la limite projective de ces foncteurs. On a également un foncteur $I : \mathbf{Pol}_{(\omega)} \longrightarrow \mathbf{Pol}_\omega$ qui permet d'identifier $\mathbf{Pol}_{(\omega)}$ à une sous-catégorie de $\mathbf{Pol}_\omega$.

2.3 Colimites de polygraphes

On aura juste besoin de sommes amalgamées dans **Pol**, mais les constructions de colimite dans **Pol** sont toutes obtenues avec le même procédé :

Lemme 2.1. *Chaque catégorie* $\mathbf{Pol}_n$ *est cocomplète. Les colimites se calculent de manière "pointwise" (la signification est dans la preuve). Les catégories* $\mathbf{Pol}_{(\omega)}$ *et* $\mathbf{Pol}_{(\omega)}$ *sont cocomplètes.*

Preuve : La preuve étant similaire pour toutes les colimites, on se limitera au cas des conoyaux.

Un diagramme dans $\mathbf{Pol}_n$ de la forme

$$P' \rightrightarrows P \longrightarrow P''$$

est un conoyau ssi, pour tout $0 \leq k \leq n$, le diagramme suivant dans **Ens** est un conoyau :

$$\Sigma''_k \rightrightarrows \Sigma_k \longrightarrow \Sigma''_k \ ,$$

où $\Sigma'_k = \mathrm{c}_k(P')$, $\Sigma_k = \mathrm{c}_k(P)$, $\Sigma''_k = \mathrm{c}_k(P'')$. On peut en suivre la preuve sur le diagramme :

$$\begin{array}{ccccccc}
\Sigma'_0 & \xrightarrow{i_0} & \Sigma'^*_0 \underset{\tau_0}{\overset{\sigma_0}{\leftleftarrows}} \Sigma'_1 & \cdots & \Sigma'_{n-1} & \xrightarrow{i_{n-1}} & \Sigma'^*_{n-1} \underset{\tau_{n-1}}{\overset{\sigma_{n-1}}{\leftleftarrows}} \Sigma'_n \\
{\scriptstyle u_0}\downarrow\downarrow{\scriptstyle v_0} & & {\scriptstyle u_0^*}\downarrow\downarrow{\scriptstyle v_0^*} \quad {\scriptstyle u_1}\downarrow\downarrow{\scriptstyle v_1} & & {\scriptstyle u_{n-1}}\downarrow\downarrow{\scriptstyle v_{n-1}} & & {\scriptstyle u_{n-1}^*}\downarrow\downarrow{\scriptstyle v_{n-1}^*} \quad {\scriptstyle u_n}\downarrow\downarrow{\scriptstyle v_n} \\
\Sigma_0 & \xrightarrow{i_0} & \Sigma_0^* \underset{\tau_0}{\overset{\sigma_0}{\leftleftarrows}} \Sigma_1 & \cdots & \Sigma_{n-1} & \xrightarrow{i_{n-1}} & \Sigma_{n-1}^* \underset{\tau_{n-1}}{\overset{\sigma_{n-1}}{\leftleftarrows}} \Sigma_n \\
{\scriptstyle w_0}\downarrow & & {\scriptstyle w_0^*}\downarrow \quad {\scriptstyle w_1}\downarrow & & {\scriptstyle w_{n-1}}\downarrow & & {\scriptstyle w_{n-1}^*}\downarrow \quad {\scriptstyle w_n}\downarrow \\
\Sigma''_0 & \xrightarrow{i_0} & \Sigma''^*_0 \underset{\tau_0}{\overset{\sigma''_0}{\leftleftarrows}} \Sigma''_1 & \cdots & \Sigma''_{n-1} & \xrightarrow{i_{n-1}} & \Sigma''^*_{n-1} \underset{\tau_{n-1}}{\overset{\sigma_{n-1}}{\leftleftarrows}} \Sigma''_n
\end{array} \ ,$$

où les colonnes de la forme :

$$\Sigma'^*_k \rightrightarrows \Sigma^*_k \longrightarrow \Sigma''^*_k \ ,$$

avec ${\Sigma'}_k^* = \mathrm{c}_k^*(P')$, $\Sigma_k^* = \mathrm{c}_k^*(P)$, ${\Sigma''}_k^* = \mathrm{c}_k^*(P'')$, sont données fonctoriellement (mais ce ne sont pas des conoyaux en général). L'inclusion (par l'identification triviale évoquée plus haut) de $\mathbf{Pol}_n$ dans $\mathbf{Pol}$ commute avec toutes les colimites (parce que les plongements I_{n+1} ont des adjoints à droite, les S_n''), le calcul des colimites est le même dans les deux catégories.

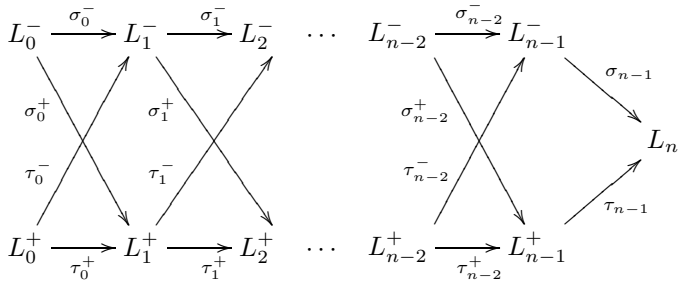
La même preuve s'adapte évidemment aux deux dernières catégories puisque les foncteurs I_n et Sn commutent aux limites inductives. □

3 transiteurs

La structure de transiteur définie ci-dessous est l'outil de base pour l'étude polygraphique des automates et des grammaires. Informellement, on peut dire que les transiteurs sont aux polygraphes ce que, en géométrie, les variétés à bord sont aux variétés.

3.1 Définition des n-transiteurs

Soit n un entier naturel, un n-transiteur L est la donnée dans la catégorie $\mathbf{Pol}$ d'un diagramme commutatif de la forme suivante :



où, L_n^- est un n-polygraphe et, pour tout $0 \leq i \leq n-1$, L_i^- et L_i^+ sont des i-polygraphes (en bref : un n-transiteur est un "n-cospan filtré par les dimensions" de $\mathbf{Pol}$). La commutativité s'exprime par des relations de la forme : $\sigma\sigma = \sigma\tau$, $\tau\sigma = \tau\tau$ avec les paires d'indices qui conviennent.

Un homomorphisme entre n-transiteurs $f : L \longrightarrow L'$ est une transformation naturelle entre diagrammes. La catégorie ainsi obtenue se note $\mathbf{Trs}_n$. On note $T_n : \mathbf{Trs}_n \longrightarrow \mathbf{Pol}_n$ le foncteur défini, pour tout transiteur L, par $T_n(L) = L_n$ et, pour tout $0 \leq i < n$ et $\varepsilon \in \{\pm\}$, $T_i^\varepsilon : \mathbf{Trs}_n \longrightarrow \mathbf{Pol}_n$, le foncteur défini par $T_i^\varepsilon(L) = L_i^\varepsilon$. Pour $1 \leq n$, on a des foncteurs de restriction évidents $\mathbf{s}_{n-1}^\circ, \mathbf{t}_{n-1}^\circ : \mathbf{Trs}_n \rightrightarrows \mathbf{Trs}_{n-1}$ tels que $T_{n-1}(\mathbf{s}_{n-1}^\circ(L)) = L_{n-1}^-$ et $T_{n-1}(\mathbf{t}_{n-1}^\circ(L)) = L_{n-1}^+$.

On définit ainsi une (grande) catégorie globulaire qu'on notera $\mathbf{Trs}$:

$$\mathbf{Trs}_0 \underset{\mathbf{t}_0^\circ}{\overset{\mathbf{s}_0^\circ}{\leftleftarrows}} \mathbf{Trs}_1 \underset{\mathbf{t}_1^\circ}{\overset{\mathbf{s}_1^\circ}{\leftleftarrows}} \mathbf{Trs}_2 \quad \cdots \quad \mathbf{Trs}_{n-1} \underset{\mathbf{t}_{n-1}^\circ}{\overset{\mathbf{s}_{n-1}^\circ}{\leftleftarrows}} \mathbf{Trs}_n \quad \cdots$$

Dans [Bat 98], Batanin montre que les n-spans d'une catégorie munie de produits fibrés forment une catégorie globulaire monoïdale. Ce dont on a besoin ici c'est du résultat dual sur les n-cospans, plus précisément, sur les n-transiteurs puisque ce sont des n-cospans dans

la catégorie $\mathbf{Pol}_{(\omega)}$ qui admet des sommes amalgamées. La catégorie globulaire **Trs** peut ainsi être munie d'une structure qui en fait une catégorie globulaire monoïdale. La notation des lois monoïdales est notée par le symbole "$\otimes$".

3.2 transiteurs marqués

On appelle *n-polygraphe marqué* un couple $P = (P, \theta)$ où P est un n-polygraphe et où θ est un élément de $\mathrm{c}^*_n(P)$ appelé *marque* de P (avec un abus de langage, on emploiera encore la notation $P = (P, \theta_P)$). Un homomorphisme $f : (P, \theta_P) \longrightarrow (P', \theta_{P'})$ entre deux n-polygraphes marqués est un homomorphisme $f : P \longrightarrow P'$ entre les n-polygraphes sous-jacents tels que $\theta_{P'} = f_n(\theta_P)$. Les n-polygraphes marqués forment avec leurs homomorphismes une catégorie $\mathbf{Pol}^\bullet_n$.

Un *n-transiteur marqué* est la donnée d'un couple (L, θ) où L est un n-transiteur et θ une famille de marques des i-polygraphes composants de L telle que (L_n, θ_n) soit un n-polygraphe marqué et, pour chaque couple d'indices (i, ε) tel que $0 \leq i < n$ et $\varepsilon \in \{\pm\}$, $(L_i^\varepsilon, \theta_i^\varepsilon)$ soit un i-polygraphe marqué. De plus, ces données doivent vérifier les relations $\sigma_{n-1}(\theta_{n-1}) = \overline{\mathrm{s}}_{n-1}(\theta_n)$, $\tau_{n-1}(\theta_{n-1}) = \overline{\mathrm{t}}_{n-1}(\theta_n)$ et, pour tout $0 \leq i < n$, $\sigma_i^\varepsilon(\theta_i^\varepsilon) = \overline{\mathrm{s}}_i(\theta_{i+1}^\varepsilon)$, $\tau_i^\varepsilon(\theta_i^\varepsilon) = \overline{\mathrm{t}}_i(\theta_{i+1}^\varepsilon)$ et $\sigma_{n-1}(\theta_{n-1}) = \overline{\mathrm{s}}_{n-1}(\theta_n)$.

Les n-transiteurs marqués forment les objets d'une catégorie évidente notée $\mathbf{Trs}^\bullet_n$ anaTrsue à la catégorie globulaire $\mathbf{Trs}_n$ et, pour tout $n \geq 1$, on a deux foncteurs : $\mathbf{s}^\bullet_{n-1}, \mathbf{t}^\bullet_{n-1} : \mathbf{Trs}^\bullet_n \rightrightarrows \mathbf{Trs}^\bullet_{n-1}$ définis, pour tout objet (L, θ) de $\mathbf{Trs}^\bullet_n$, par $\mathbf{s}^\bullet_{n-1}(L, \theta) = (\mathbf{s}^\circ_{n-1}(L), \theta')$, $\mathbf{t}^\bullet_{n-1}(L, \theta) = (\mathbf{t}^\circ_{n-1}L, \theta'')$, où θ' et θ'' coïncident avec les données de θ, respectivement, sur $\mathbf{s}^\circ_{n-1}(L)$ et $\mathbf{t}^\circ_{n-1}(L)$.

On obtient ainsi une catégorie globulaire notée $\mathbf{Trs}^\bullet$ et anaTrsue à **Trs** :

$$\mathbf{Trs}^\bullet_0 \underset{\mathbf{t}^\bullet_0}{\overset{\mathbf{s}^\bullet_0}{\leftleftarrows}} \mathbf{Trs}^\bullet_1 \underset{\mathbf{t}^\bullet_1}{\overset{\mathbf{s}^\bullet_1}{\leftleftarrows}} \mathbf{Trs}^\bullet_2 \qquad \cdots \qquad \mathbf{Trs}^\bullet_{n-1} \underset{\mathbf{t}^\bullet_{n-1}}{\overset{\mathbf{s}^\bullet_{n-1}}{\leftleftarrows}} \mathbf{Trs}^\bullet_n \qquad \cdots$$

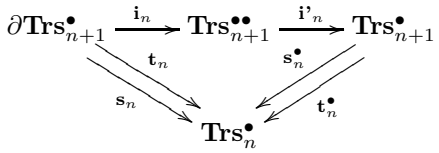
Nous voulons maintenant munir $\mathbf{Trs}^\bullet$ d'une structure de catégorie globulaire monoïdale, c'est-à-dire d'y définir des "lois monoïdales" compatibles avec celles de **Trs**. Soient (L', θ') et (L, θ) deux n-transiteurs marqués tels que, pour un $i \in \mathbb{N}$ tel que $0 \leq i \leq n-1$, on ait $\mathbf{s}^\bullet_i(L') = \mathbf{t}^\bullet_i(L)$. On définit un n-transiteur marqué composé $L \otimes^\bullet_i L' = (L \otimes_i L', \theta'')$ où θ'' est déterminé par le composé de n-cellules $(\theta'')_n = (\theta')_n *_i (\theta)_n$, ce qui est rendu possible par la relation anaTrsue sur les marques $\mathrm{s}_i({\theta'}_i^-) = \mathrm{t}_i(\theta_i^+)$ déduite de la relation anaTrsue sur les n-transiteurs marqués qui leur correspondent.

4 Plexes et polyplexes

Les plexes sont aux polygraphes un peu ce que les simplexes sont aux ensembles simpliciaux, des briques élémentaires de leur construction. Mais la situation est plus complexe à décrire ici parce qu'on est invité à les introduire par une définition inductive (on pourrait aussi rechercher une caractérisation des polyplexes parmi les polygraphes).

4.1 Une construction essentielle

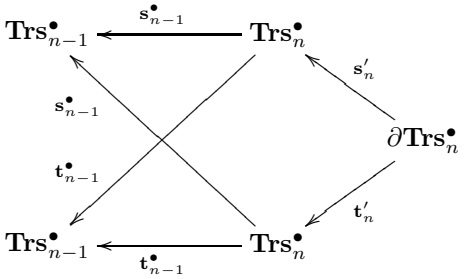
Nous allons expliciter le diagramme de catégories suivant :



où

$$\mathbf{s}^\bullet_n\,\mathbf{i}'_{n+1}\mathbf{i}_n = \mathbf{s}_n, \quad \text{et} \quad \mathbf{t}^\bullet_n\,\mathbf{i}'_{n+1}\mathbf{i}_n = \mathbf{s}_n.$$

Pour tout $n \in \mathbb{N}$, on définit la catégorie $\partial\mathbf{Trs}^\bullet_n$ comme limite projective de catégories :

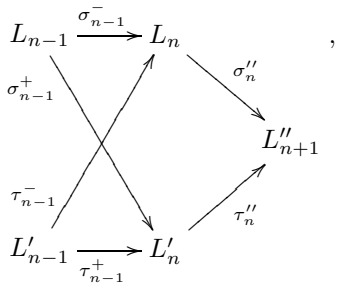


Par définition des projections $\mathbf{s}'_n$ et $\mathbf{t}'_n$ de cette limite projective, on a les relations $\mathbf{s}^\bullet_{n-1}\mathbf{s}'_n = \mathbf{s}^\bullet_{n-1}\mathbf{t}'_n$ et $\mathbf{t}^\bullet_{n-1}\mathbf{s}'_n = \mathbf{t}^\bullet_{n-1}\mathbf{t}'_n$. On appelle *n-paire parallèle* (de transiteurs marqués) un objet de cette catégorie. C'est la donnée d'un couple d'objets $((L,\theta),(L',\theta'))$ de $\mathbf{Trs}^\bullet_n$ tels que $\mathbf{s}^\bullet_{n-1}(L) = \mathbf{s}^\bullet_{n-1}(L')$ et $\mathbf{t}^\bullet_{n-1}(L) = \mathbf{t}^\bullet_{n-1}(L')$.

La construction que l'on vient de faire n'est autre que celle décrite en 1.3. dans sa version catégorique.

On définit une seconde catégorie auxiliaire $\mathbf{Trs}^{\bullet\bullet}_{n+1}$. Elle est construite de manière analogue à $\mathbf{Trs}^\bullet_n$. Ses objets sont formés de couples (L,Θ) tels que L est un n-transiteur et où Θ fournit le même type de donnée que le θ utilisé pour la définition d'un objet de $\mathbf{Trs}^\bullet_n$, mais, à la dimension n, Θ fournit non pas un élément de $\mathrm{c}^*_n(L_n)$, mais un couple de tels éléments, donc une double marque $\Theta_n = (\theta^-_n, \theta^+_n)$, avec les conditions suivantes : $\mathrm{s}_n(\theta^-_n) = \mathrm{s}_n(\theta^+_n) = \sigma_{n-1}(\theta^-_{n-1})$ et $\mathrm{t}_n(\theta^-_n) = \mathrm{t}_n(\theta^+_n) = \tau_{n-1}(\theta^+_{n-1})$.

On définit le foncteur $\mathbf{i}_n : \partial\mathbf{Trs}^\bullet_n \longrightarrow \mathbf{Trs}^{\bullet\bullet}_{n+1}$ de la manière suivante : soit $((L,\theta),(L'\theta'))$ un objet de $\partial\mathbf{Trs}^\bullet_n$, le foncteur $\mathbf{i}_n$ lui associe l'objet $\mathbf{i}_n((L,\theta,L'\theta')) = (L'',\Theta)$ où, d'abord L'' est le n+1-transiteur défini à l'aide de la somme amalgamée :



,

puis θ'' est le prolongement de θ et θ', et où θ''_{n+1} est le couple de n-cellules $(\sigma''_n(\theta_n), \tau''_n(\theta'_n))$.

Enfin, on définit le foncteur $\mathbf{i'}_{n+1} : \mathbf{Trs}^{\bullet\bullet}_{n+1} \longrightarrow \mathbf{Trs}^{\bullet}_{n+1}$. Ce foncteur associe à un objet (L, Θ) (avec $\Theta_n = (\theta_n^-, \theta_n^+)$) de **Trs** un $(n+1)$-transiteur $\mathbf{i}'_{n+1}((L, \Theta)) = (\overline{L}, \overline{\theta})$ où $\overline{L}$ est le $(n+1)$-transiteur L'' (pour lequel, il faut le noter, l'ensemble $\mathrm{c}_{n+1}T(L)$ est vide !) construit précédement et auquel on a ajouté formellement (polygraphiquement peut-on dire) une $(n+1)$-cellule qu'on notera $\varpi(L, \Theta)$ ou encore, par abus de langage, $\varpi(\theta_n^-, \theta_n^+)$ et caractérisée par $\mathrm{s}_n(\varpi) = \sigma''_n(\theta_n)$ et $\mathrm{t}_n(\varpi) = \tau_n(\theta_n)$. Le $(n+1)$-transiteur marqué $(\overline{L}, \overline{\theta})$ est donc tel que $\overline{L}_{n+1} = \{\varpi\}$ et $\overline{\theta}_{n+1} = \varpi$. Tout n-polyplexe est, par construction engendré par des n-plexes dans la n-catégorie (catégorique) $\mathbb{M}_n$.

Nous venons de définir ainsi, une ∂-structure sur $\mathbf{Trs}^{\bullet}$ avec la suite de foncteurs $\partial_n : \partial\mathbf{Trs}^{\bullet}_n \longrightarrow \mathbf{Trs}^{\bullet}_{n+1}$ en posant $\partial_n = \mathbf{i'}_{n+1}\mathbf{i}_n$.

4.2 Les polyplexes

On va construire une sous-catégorie monoïdale de $\mathbf{Trs}^{\bullet}$ dont les éléments, appelés polyplexes jouent un rôle analogue par rapport aux polygraphes que, par exemple, les simplexes par rapport aux ensembles simpliciaux. Leur rôle principal est de représenter les foncteurs c_n et c_n^* dans **Pol** (voir ci-dessous).

Nous avons décrit la catégorie globulaire monoïdale $\mathbf{Trs}^{\bullet}$, puis nous lui avons adjoint une ∂-structure. On a donc une structure formée d'un couple $(\mathbf{Trs}^{\bullet}, \partial)$, une catégorie globulaire monoïdale et une ∂-structure sur l'ensemble globulaire sous-jacent.

Pour cette structure, la notion, par exemple, de sous-structure engendrée par certaines données a un sens. Ainsi il existe une sous-structure de cette structure $(\mathbf{Trs}^{\bullet}, \partial)$ engendrée par $\emptyset$. On notera $\mathbb{M}$ la catégorie globulaire monoïdale sous-jacente à cette sous-structure. On appelle *n-polyplexes* les éléments de $\mathbb{M}_n$ et *n-plexes* les n-polyplexes de la forme $\partial_n(N, N')$ où N et N' sont des $(n-1)$-polyplexes. On notera $\mathbb{P}_n$, l'ensemble des n-plexes. On identifiera souvent un n-polyplexe M au n-polygraphe $T(P)$.

Remarque 4.1. L'introduction des polygraphes et transiteurs marqués, n'avait pas d'autre but, ici, que de décrire les plexes et polyplexes. On peut espérer trouver un moyen plus direct de les construire.

Nous introduirons sans preuve le résultat suivant (qui reste donc une conjecture) : soit P un n-polygraphe, on a des bijections canoniques :

$$\mathrm{c}_i(P) \simeq \coprod_{E \in \tilde{\mathbb{P}}_i} \mathrm{Hom}(E, P), \qquad \mathrm{c}_i^*(P) \simeq \coprod_{M \in \tilde{\mathbb{M}}_i} \mathrm{Hom}(M, P)$$

où $\tilde{\mathbb{M}}_i$ (resp. $\tilde{\mathbb{P}}_i$) est une partie de $\mathbb{M}_i$ (resp. $\mathbb{P}_i$) représentative des classes d'isomorphie de cet ensemble.

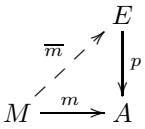
5 Automates en dimensions supérieures

5.1 Une définition formelle des automates

Le schéma de définition d'un automate est réduit ici à un minimum conceptuel. Il ne prend sa valeur que par l'implémentation des données abstraites qui figurent dans sa définition.

Soit $\mathcal{E}$ une catégorie et $\mathbb{M} \subset \mathrm{Ob}(\mathcal{E})$. Les éléments de $\mathbb{M}$ seront appelés *formes verbales*. Un $(\mathcal{E}, \mathbb{M})$-*automate* est un morphisme $p : E \longrightarrow A$ de $\mathcal{E}$. Relativement à ces données, l'objet

A s'appelle l'*alphabet*, l'objet E, l'espace des *états*. Un *mot* est la donnée d'un morphisme $m : M \longrightarrow A$ de $\mathcal{E}$ tel que $M \in \mathbb{M}$. On dit que le mot m est *reconnu* par le $(\mathcal{E}, \mathbb{M})$-automate s'il existe un morphisme $\overline{m} : M \longrightarrow E$ qui relève m le long de p *i.e.* $m = p\,\overline{m}$. L'ensemble des mots (à isomorphisme près) est désigné sous le nom de *langage* reconnu par cet $(\mathcal{E}, \mathbb{M})$-automate p.



L'expression "à isomorphisme près" signifie que, s'il existe un isomorphisme $i : M \longrightarrow M'$ et deux mots $m : M \longrightarrow A$, $m' : M' \longrightarrow A$ tels que $m' = m\,i$, on identifie ces deux mots.

5.2 n-Automates

Définition 5.1. *Pour tout $n \in \mathbb{N}$, on appelle* n-automate *(ou* automate n-D*) un* $(\mathbf{Trs}_n, \mathbb{M}_n)$-*automate.*

Remarque 5.2. A ce niveau de généralités, nous n'introduisons *a priori* aucunes conditions de finitude, de non vacuité, ni même de "déterminisme" (lequel demande à être défini ici). C'est au contexte de préciser ces conditions.

L'exemple le plus simple, et parfaitement trivial, est celui des 0-automates, c'est-à-dire la donnée d'une application $p : E \longrightarrow A$. Un mot revient à la donnée d'un élément de A, il est accepté, s'il est dans l'image de p.

5.3 automates finis

Nous allons montrer comment interpréter les automates finis classiques comme des automates 1-D.

Nous supposons le lecteur familier avec le concept d'*automate fini*. Nous nous contentons de fixer la terminologie et de rappeler brièvement la définition. Un automate fini (déterministe) est, essentiellement, la donnée de deux ensembles finis Σ et E, respectivement appelés alphabet et ensemble des états de l'automate, et d'une action associative et unitaire (qu'on notera multiplicativement)

$$\Sigma^* \times E \longrightarrow E.$$

du monoïde libre Σ^* engendré par Σ. Cette donnée se complète par celle d'un état particulier q_0 appelé *état initial* et d'une partie F de E dont les éléments sont appelés *états finaux*.

Montrons comment cette structure apparaît comme un cas particulier de la notion de 1-*automate* en traduisant les données précédentes de Σ, E, ... en celle de $p : E \longrightarrow A$ qui est un homomorphisme de graphes.

On interprète d'abord l'ensemble Σ comme un graphe A. Il suffit de poser $\mathrm{c}_0(A) = 1$ (où 1 note un singleton $\{0\}$ ou $\{*\}$) et $\mathrm{c}_1(A) = \Sigma$. On construit un graphe B en posant $\mathrm{c}_0(B) = E$, $\mathrm{c}_1(B) = \Sigma \times E$ et en définissant les sources et buts de B de la manière suivante : pour tout $(a, q) \in \Sigma \times E$, $\mathrm{s}_0(a, q) = q$ et $\mathrm{t}_0(a, q) = aq$. Enfin p est défini par $p(a, q) = a$.

5.4 Réseaux de Petri

Nous avons indiqué dans l'introduction que nous ne donnerons pas d'exemples, mais comme cela peut se rattacher à une référence précise (voir plus bas), nous amorcerons cet exemple :

Les réseaux de Petri peuvent s'interpréter comme un cas particulier de 2-automates. Nous commencerons par donner une version de ces structures qui est à la fois plus simple (du point de vue du formalisme des polygraphes) et plus générale :

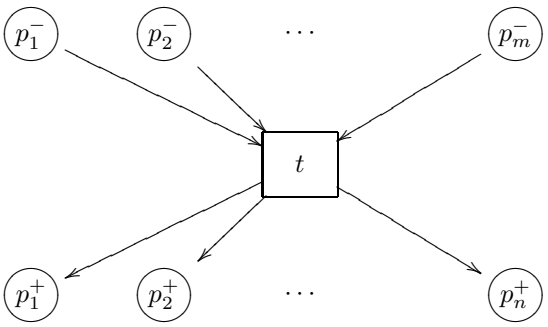
–

La donnée d'un *réseaux de Petri non commutatif* R est constitué par les données suivantes :

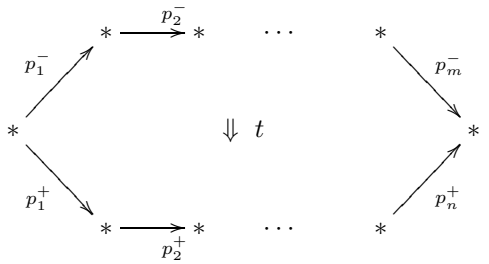
- Un ensemble fini d'éléments appelés *places*.
- Un ensemble fini d'éléments appelés *transitions*.
- Un ensemble fini de flèches appelée *pré* et de la forme $i^- : p \longrightarrow t$ où p est une place et t une transition.
- Un ensemble fini de flèches appelée *post* et de la forme $i^+ : t \longrightarrow p$ où p est une place et t une transition.

On traduit ces données par celle d'un polygraphe $P(R)$ ainsi défini :

- Une seule 0-cellule notée "$*$".
- Pour chaque place p, une 1-cellule, notée par le même symbole, $p : * \longrightarrow *$.
- Pour chaque transition t une 2-cellule t, une 2-cellule, notée par le même symbole, mais avec les
-



se traduit par



On les appellera réseaux de Petri *non commutatifs* (Les transitions tiennent compte de l'ordre des places). Pour avoir les réseaux de Petri habituel, donc commutatifs, où l'ordre des jetons n'a pas d'importance, il faut introduire des 2-cellules de transposition entre les places deux à deux. Soit p et q deux places, on introduit dans le polygraphe qui représente le réseau de Petri une 2-cellule de la forme : $pq \Rightarrow qp$.

Cette traduction est à la base de la mise sous forme polygraphique des réseaux de Petri. Ceux-ci peuvent servir d'automates (également de grammaires), mais nous n'avons pas la place ici de développer plus longuement cette question. Pour un exposé plus complet voir [Bu 02] et [Guir 06]. Dans ce dernier article, Yves Guiraud montre comment on peut éliminer, en passant à une dimension supérieure, les 2-cellules de transpositions.

6 Grammaires en dimensions supérieures

6.1 Une définition formelle des grammaires

Même remarque que pour la définition générale des automates, on commence par donner le minimum conceptuel de la description des grammaires.

Soit $U : \overline{\mathcal{E}} \longrightarrow \mathcal{E}$ un foncteur et $\mathbb{M} \subset \mathrm{Ob}(\mathcal{E})$, $\overline{\mathbb{M}} \subset \mathrm{Ob}(\overline{\mathcal{E}})$ deux ensembles, les éléments de $\mathbb{M}$ et $\overline{\mathbb{M}}$ seront appelés, respectivement, *formes verbales* et *formes grammaticales*. On appelle $(U, \overline{\mathbb{M}}, \mathbb{M})$-*grammaire*, un objet G de $\overline{\mathcal{E}}$. Relativement à ces données $(U, \overline{\mathbb{M}}, \mathbb{M}, G)$, l'objet $U(G)$ s'appelle *alphabet initial* de la grammaire. Un *mot*, relatif à ces données, est la donnée d'un morphisme $m : M \longrightarrow U(G)$ de $\mathcal{E}$ où $M \in \mathbb{M}$. On dit que le mot m est *accepté* par cette grammaire, s'il existe un morphisme $\overline{m} : \overline{M} \longrightarrow \overline{G}$ de $\overline{\mathcal{E}}$ tel que $\overline{m} \in \overline{\mathbb{M}}$ et $m = U(\overline{m})$. On dit que $\overline{m}$ est une *analyse grammaticale* du mot m.

Remarque 6.1. Notre définition des grammaires diffère de la définition classique en de nombreux points et, en tout premier lieu, par une inversion de la "flèche du temps", c'est-à-dire un renversement de sens du calcul qui fait que $U(G)$, dans la terminologie classique, serait appelé "alphabet terminal". (Nous choisissons ce sens pour être en accord avec le travail des automates, mais aussi parce qu'il nous semble que c'est le "sens" du mot "analyse" : on commence par prendre l'objet que l'on cherche à analyser.)

6.2 Les automates sont des grammaires

Nous montrerons dans une version plus étendue de ce travail comment, précisément, un n-automate peut être transformé en n-grammaire — voir définition ci-dessous — équivalente (*i.e.* de sorte que les mots reconnus par l'automate soient les mots acceptés par la grammaire). En revanche la traduction dans l'autre sens n'est généralement pas possible : les n-grammaires, pour une dimension donnée, produisent en général des objets plus complexes que ceux produits par les n-automates. Ceci est vrai en dimension 1, et sans doute, en toutes dimensions.

La traduction des automates en grammaires proposée ici est purement formelle, mais elle est tout-à-fait conforme à ce qui peut être fait dans des cas concrets. Elle montre bien la liaison entre les deux notions.

On reprend certaines données et notations de la section précédente. Soit $p : E \longrightarrow A$ un $(\mathcal{E}, \mathbb{M})$-*automate* où $\mathcal{E}$ est une catégorie et $\mathbb{M} \subset \mathrm{Ob}(\mathcal{E})$ un ensemble de formes de mots. On va lui associer une grammaire formelle équivalente :

On considère la catégorie $\overline{\mathcal{E}} = \mathcal{E}^{\mathbf{2}}$ où $\mathbf{2}$ est la catégorie engendrée par le graphe réduit à deux objets et une flèche $(\cdot \longrightarrow \cdot)$. La catégorie $\mathcal{E}^{\mathbf{2}}$ est la *catégorie des morphismes* de $\mathcal{E}$. Ses objets sont les morphismes de $\mathcal{E}$ et ses morphismes sont les carrés commutatifs de $\mathcal{E}$. Précisément, un morphisme $(\overline{f}, f) : g \longrightarrow h$ de cette catégorie est formés de morphismes tels que l'égalité suivante $gf = h\overline{f}$ ait un sens et soit satisfaite. On a un foncteur $U : \mathcal{E}^{\mathbf{2}} \longrightarrow \mathcal{E}$

défini, sur un objet $g : X \longrightarrow Y$, par $U(g) = Y$ et, pour un morphisme, avec les notations ci-dessus, par $U(\overline{f}, f) = f$. On prend pour ensemble de formes de mots, l'ensemble $\mathbb{M} \subset \mathrm{Ob}(\mathcal{E})$ des formes de mots de l'automate et, pour ensemble de formes grammaticales, l'ensemble des carrés de la forme $(\overline{m}, m) : \mathrm{id}_M \longrightarrow p$ où $M \in \mathcal{M}$ est la source de m.

On considère alors la $(U, \overline{\mathbb{M}}, \mathbb{M})$-grammaire $p : E \longrightarrow A$ comme objet de $\mathcal{E}^{\mathbf{2}}$. Cette grammaire accepte les même mots que l'automate associé.

6.3 n-Grammaires

Définition 6.2. *Pour tout $n \in \mathbb{N}$, on apelle* n-grammaire *(ou* grammaire n-D*) une $(\mathrm{s}_n, \mathbb{M}_{n+1}, \mathbb{M}_n)$-grammaire, où $\mathrm{s}_n : \mathbf{Trs}_{n+1} \longrightarrow \mathbf{Trs}_n$ est le foncteur source sur les transiteurs.*

Remarque 6.3. Mêmes précisions que dans le cas des automates sur l'absence, *a priori*, de conditions de finitude , de non vacuité etc . . . . Ici encore nos définitions sont énoncées sans *a priori* déterministe.

Le cas le plus simple, pas nécessairement trivial, est celui des 0-grammaires : il concerne ce que l'on peut appeler le "problème du labyrinthe" et consiste à trouver une sortie dans un graphe.

Je renvoie à un prochain travail la donnée d'exemples. Toutefois, on trouvera un premier aperçu de tels exemples dans un manuscrit "Automata and Grammars of Higher Dimensions" qui sont des notes personnelles pour un exposé fait à Marseille en 2002 [Bu 02]. On y trouvera, entre autre, la traduction en termes polygraphiques des automates à piles et des machines de Turing (avec un complément qui est une version corrigée de ce dernier cas).

Remerciements

Ce travail trouve ses orignes en partie dans une série d'exposés (lesquels étaient surtout consacrés à illustrer par des exemples — commme par exemple les langages d'images — la conception de structures de calculs telle que développée ici) qui ont été donnés dans le cadre d'un "groupe de travail" qui s'est tenu durant l'année universitaire 1995-96 à l'Université Denis-Diderot (Paris 7). Je remercie les collègues qui ont suivi mes exposés et qui ont partagé avec moi, dans un climat d'amitié, un certain nombre de ces idées, parfois même en participant à leur élaboration en cours d'exposés. Un grand merci d'abord à mon ami François Métayer qui a été l'organisateur du groupe comprenant Elisabeth Burroni, Gianluigi Bellin, Arnaud Fleury, Mark Junker, Sylvain Kahane, Kazem Lelahi, Jacques Penon, Jacques Van de Wiele.

Je ne saurais clore cette liste sans y ajouter le nom de mon ami Yves Lafont qui n'était pas à Paris à cette époque et à qui les polygraphes doivent beaucoup.

Références

- [Bat 98] M. A. Batanin *Monoidal Globular Categories As a Natural Environment for the Theory of Weak n-Categories.* Advances in Mathematics 136, 39-103 (1998)
- [Bu 71] A. Burroni : Thèse de 3$^{\mathrm{e}}$cycle. Paris 1971.
- [Bu 87] A. Burroni : Conférence (non-publiée) aux Journées d'Etudes E.L.I.T. 27 juin - 2 juillet (1988) Paris.
- [Bu 89] A. Burroni : Congrès international Bangor. juillet 1989.

- [Bu 89'] A. Burroni : Congrès Saint Jacques de Compostelle. Septembre 1989
- [Bu 91] A. Burroni : *Higher dimensional word problem.* LNCS 530 (septembre 1991) 94-105
- [Bu 93] A. Burroni : Higher-dimensional word problems with applications to equational logic. Theoretical Computer Science, 115 :43-62, 1993.
- [Bu 02] A. Burroni : *Automatas and Grammars of Higher Dimensions.* (http ://www.math.jussieu.fr/ burroni/). Marseille Luminy (http ://iml.univ-mrs.fr/ lafont/li2002/semaine2.html)
- [Bu 02'] Conférence à un Séminaire (SIC) du 16 mars 2002 à Tours.
- [Ehr 63] Charles Ehresmann : catégories structurées. *Annales scientifique de l'E.N.S. 3^esérie, tome 80, n^0 4 (1963).* 349-426.
- [Guir 06] Yves Guiraud. *Two polygraphics présentations of Petri net.* Annals of Pure and Applied Logic 141, n°1-2, 266-295, 2006.
- [Met 03] François Métayer. *Résolutions by polygraphs.* Theory and Applications of Categories,
- [Pow 91] John Power. An n-categorical pasting theorem. In Category Theory, Proc. Int. Conf., Como/Italy 1990, number 1488 in Lect. Notes Math., pages 326-358, 1991.