

Chapitre 1

Introduction

Ce mémoire se place dans l'étude des *modèles du calcul continu*. Nous y montrons que la géométrie plane permet de calculer. Nous définissons un calcul géométrique et utilisons la continuité de l'espace et du temps pour stocker de l'information au point de provoquer des accumulations.

Dans le monde des automates cellulaires, on parle souvent de *particules* ou de *signaux* (qui forment des lignes discrètes sur les diagrammes espace-temps) tant, pour analyser une dynamique que, pour concevoir des automates cellulaires particuliers. Le point de départ de nos travaux est d'envisager des versions continues de ces signaux. Nous définissons un modèle de calcul continu, les *machines à signaux*, qui engendre des figures géométriques suivant des règles strictes. Ce modèle peut se comprendre comme une extension continue des automates cellulaires. .

Dans cette introduction, nous présentons les automates cellulaires (un modèle du calcul massivement parallèle, profondément discret) et donnons quelques exemples de particules et de signaux discrets à l'origine de notre modèle. Après un rapide état de l'art sur ce que nous connaissons de plus proche de notre approche, nous présentons ce modèle, nos résultats et l'organisation de ce mémoire.

1.1 Automates cellulaires

Les automates cellulaires sont discrets par nature : l'espace est discrétisé en cellules (typiquement sur \mathbb{Z}^d), chaque cellule ne peut être que dans un nombre fini d'états et ne perçoit qu'une partie finie des cellules qui l'entourent. La mise à jour est faite de manière itérative, chaque cellule changeant d'état en fonction de ce qu'elle perçoit autour d'elle. Toutes les cellules sont mises à jour simultanément et en suivant les mêmes règles. Le processus est synchrone, uniforme et local.

Les automates cellulaires sont très étudiés depuis les années 1950 [Ula52]. En plus des études théoriques sur automates cellulaires eux-mêmes, on distingue deux façons de les aborder : pour simuler (un système existant) ou pour concevoir.

Dans le premier cas, le but est de simuler sur ordinateur des systèmes réels (qu'ils soient physiques, biologiques, économiques. . .). Lors de telles simulations, on observe assez souvent

des *signaux* (ou des *particules*) dans les diagrammes espace-temps. Ces signaux peuvent aussi bien être des artefacts de la simulation que correspondre à des phénomènes réels : de leur étude, on déduit des propriétés du système étudié. Par exemple, si l'on prend les diagrammes espace-temps (les itérations successives sont mises les unes sur les autres) de différents automates cellulaires élémentaires (deux états et chaque cellule ne perçoit les états que des deux cellules les plus proches), on peut observer des « signaux » se déplaçant comme sur les différents exemples de la Fig. 1.1 ou de la Figure 1.2 [BNR91, HSC01, JSS02, Siw01] (sous la dénomination de particule ou soliton).

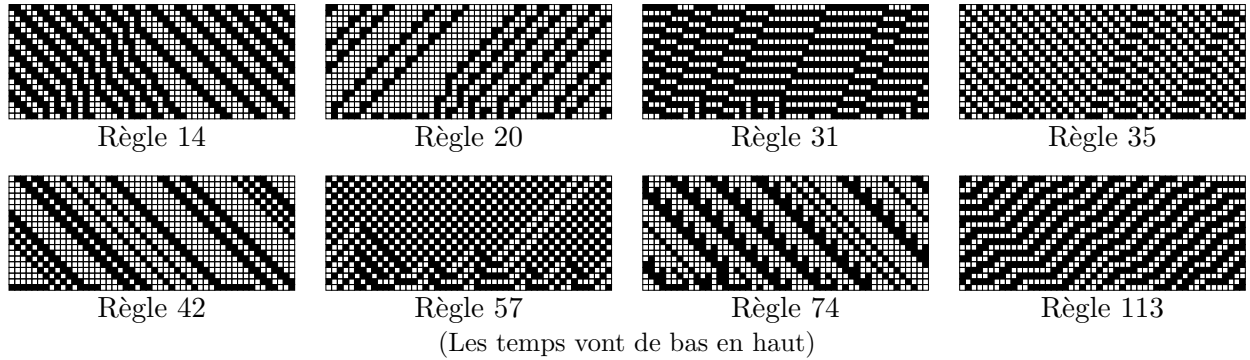


FIG. 1.1 – Différents diagrammes espace-temps d'automates cellulaires élémentaires.

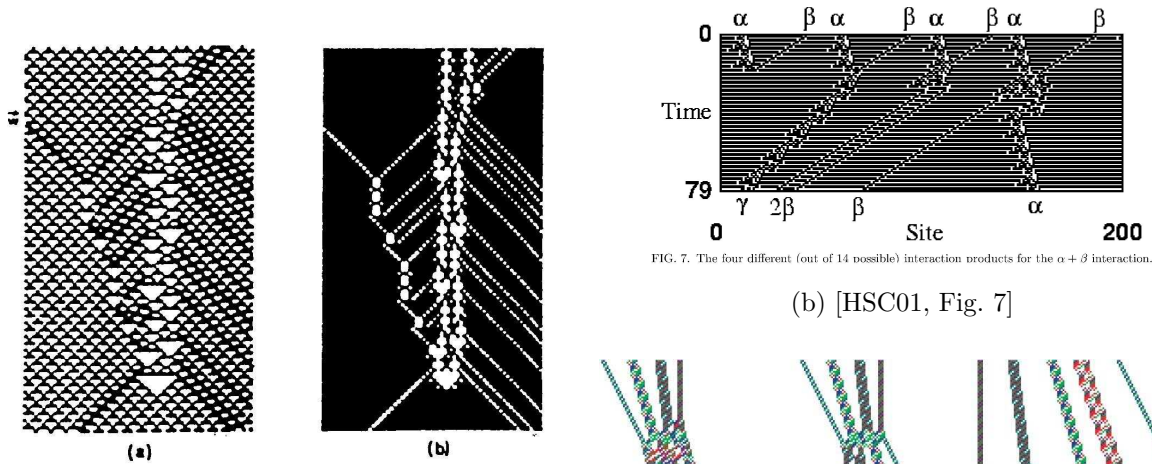


FIG. 7. Rule 54. (a) Annihilation of the radiating particle. (b) The same as (a) with the mapping defined in Fig. 6.

(a) [BNR91, Fig. 7]

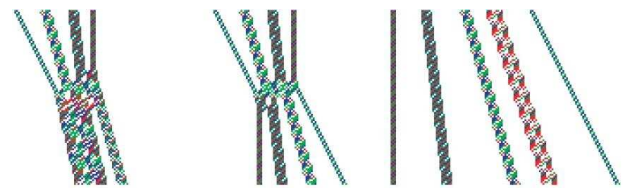


Figure 5. Two collisions of filtrons, and five free filtrons supported by the FPS model; ST diagram applies $q = 1$.

(c) [Siw01, Fig. 5]

(Les temps vont de haut en bas)

FIG. 1.2 – Exemples de particules.

Dans le second cas, on conçoit une machine (ou un circuit) massivement parallèle avant de la réaliser, ou un automate cellulaire particulier ayant certaines propriétés. Le but n'est plus

d'étudier un système mais de concevoir un système menant à bien une tâche prédéterminée. Il est souvent nécessaire de transporter de l'information d'un endroit à un autre. Pour cela, on met généralement en place des signaux véhiculant de l'information. Il arrive même que tout soit défini avec des signaux, ceux-ci définissant la structure comme les données [DL97, Fis65, LN90, Maz96a]. Éventuellement ce sont les signaux par eux-mêmes qui sont étudiés [DM02, MT99]. Les figures 1.3 à 1.6 montrent différentes illustrations de ces cas.

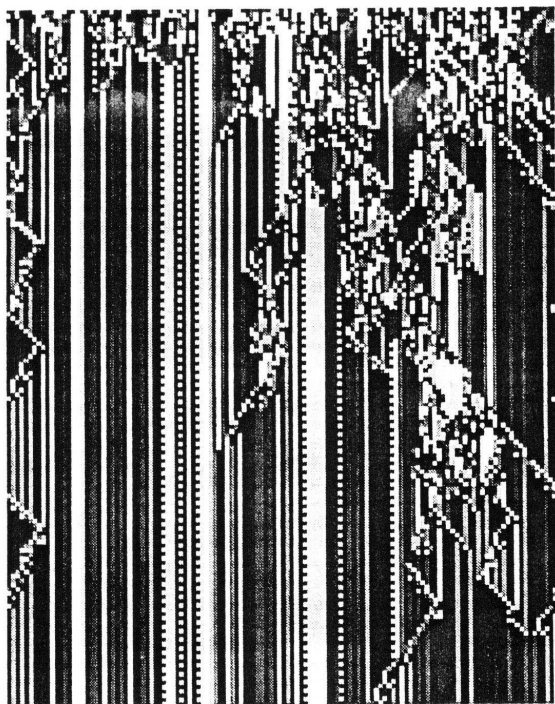



Figure 3: A simulation of the $k = 7, r = 1$ universal CA of table 3 for an uncorrelated initial state (with a density of blanks equal to 0.76). Symbols $y, 0, 1, A, B, \sqcup,$ and T are represented by 

(a) [LN90, Fig. 3]

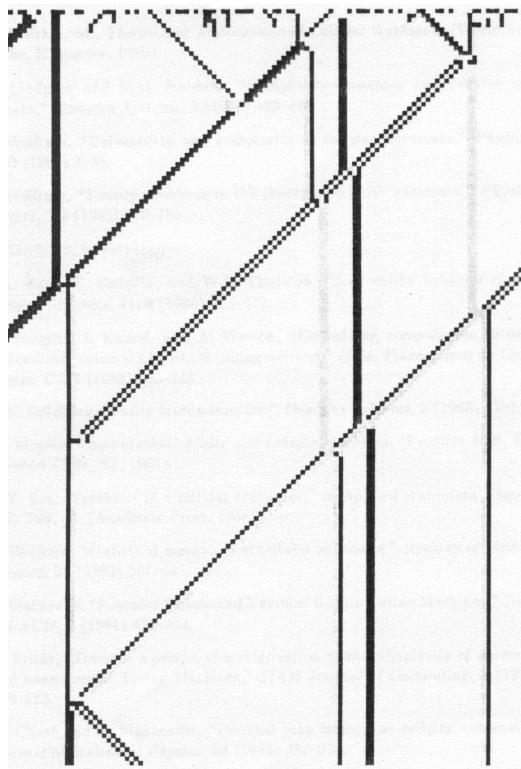



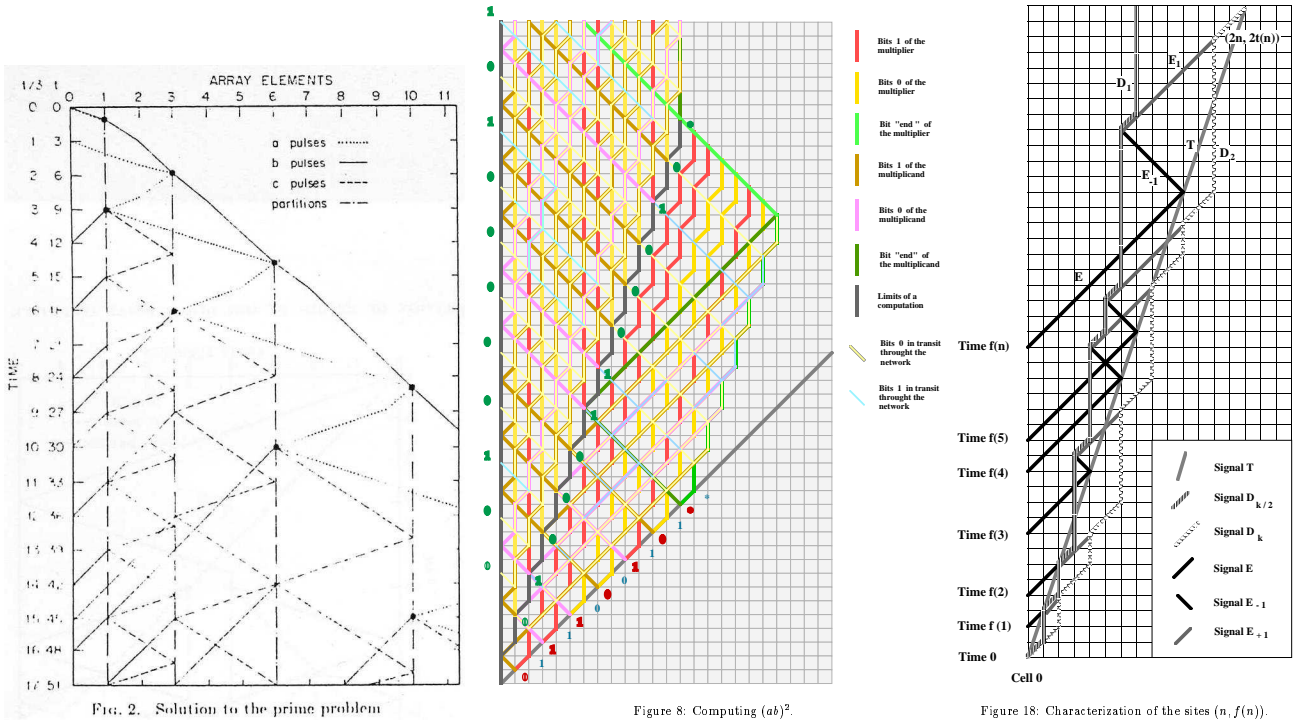
Figure 4: The $k = 4, r = 2$ universal cellular automaton of table 4 simulated starting from a random initial state. The symbols $0, 1, \sqcup,$ and $+$ are represented by 

(b) [LN90, Fig. 4]

(Les temps vont de haut en bas)

FIG. 1.3 – Utilisation de signaux pour construire une machine universelle [LN90].

De plus, différents algorithmes ont pour base des dessins *à la règle et au compas* qui sont ensuite « discrétisés » pour être exprimés sous forme d'automates cellulaires, de manière à ce que les diagrammes espace-temps de ceux-ci correspondent aux dessins. En particulier, il existe de nombreux algorithmes pour résoudre la *synchronisation d'une ligne de fusiliers (Firing Squad Synchronisation Problem)* [Maz96b]. Ce problème est celui de la synchronisation globale d'automates régulièrement disposés et « cadencés » à la même fréquence. Reformulé dans le contexte des automates cellulaires, cela correspond à avoir deux cellules distinguées (à une extrémité, un général qui démarre la synchronisation et, à l'autre, un délimiteur),



(a) [Fis65, Fig. 2]

(b) [Maz96a, Fig. 8]

(c) [MT99, Fig. 18]

(Les temps vont de haut en bas à gauche et de bas en haut pour les deux autres)

FIG. 1.4 – Algorithmes géométriques.

toutes les autres cellules sont dans un état quiescent (elles y restent tant qu'aucune information ne leur parvient). Le but est que toutes les cellules passent en même temps dans un état qui n'aura pas été atteint précédemment. La plupart des algorithmes sont fondés sur des stratégies de type *diviser pour régner* (ce qui garantit un passage à l'échelle) : on coupe la configuration en parts « égales » et on recommence jusqu'à ne plus pouvoir couper, il y a alors synchronisation.

Un des plus anciens automates cellulaires conçu dans ce but est celui de GOTO [Got66] (c.f. Fig. 1.5). On voit bien, d'un côté, une approche purement géométrique et de l'autre sa discrétisation en automate cellulaire.

Une autre approche est celle de VARSHAVSKY, MARAKHOVSKY et PESCHANSKY. On va chercher la moitié, le quart, le huitième... (tant que l'on n'a pas encore atteint la taille des cellules) depuis le général. Du milieu et de l'autre extrémité, on va aussi chercher au quart, huitième... On recommence à chaque échelle comme on le voit sur la Fig. 1.6.

Nous ne rentrons pas plus avant dans ces algorithmes, ce qui est important pour nous est qu'ils ont des origines géométriques comme en témoignent les figures extraites des articles originels, avant une discrétisation de l'espace et du temps¹. Notre souhait est de développer

¹Il y a parfois utilisation de la discrétisation, comme sur la Fig. 1.6 pour engendrer une famille infinie de signaux de pente 2^{-n} .

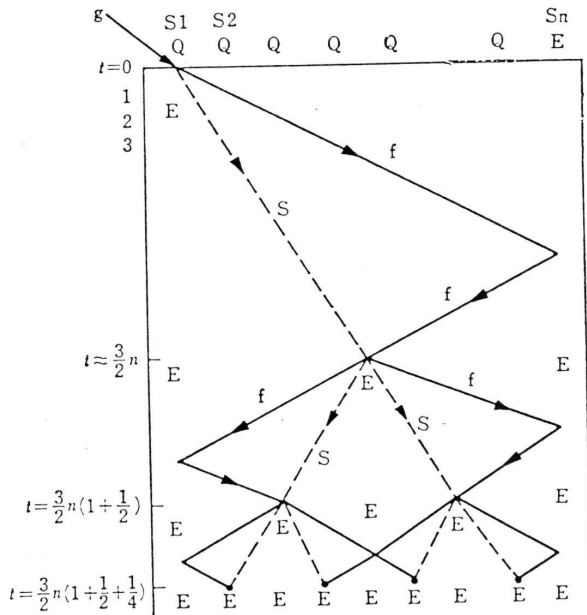


図 3-5 一斉射撃の問題 (連続近似)

(a) [Got66, Fig. 3]

G	s ₁	s ₂	s ₃	s ₄	s ₅	s ₆
g	Q	Q	Q	Q	Q	E
t=0	f's'Efs	Q	Q	Q	Q	E
1	E	Q2f	Q	Q	Q	E
2	E	Q1	Qf	Q	Q	E
3	E	Q&	Q	Qf	Q	E
4	E	Q	Q2	Q	Qf	E
5	E	Q	Q1	Q	Q	f'Ef
6	E	Q	QS	Q	f'Q	E
7	E	Q	Q	a'Q'	Q	E
8	E	Q	f'S'ESf	f's'Est	Q	E
9	E	f'2Q	E	E	Q2f	E
10	f'Ef	1Q	E	E	Q1	f'Ef
11	E	f'S'ESf	E	E	f's'Est	E
12	a'Ea	E	a'Ea	a'Ea	E	a'Ea
13	F	F	F	F	F	F

図 3-6 一斉射撃解 (n=6)

(b) [Got66, Fig. 6]

(Les temps vont de haut en bas)

FIG. 1.5 – Algorithmes géométriques pour la synchronisation de [Got66].

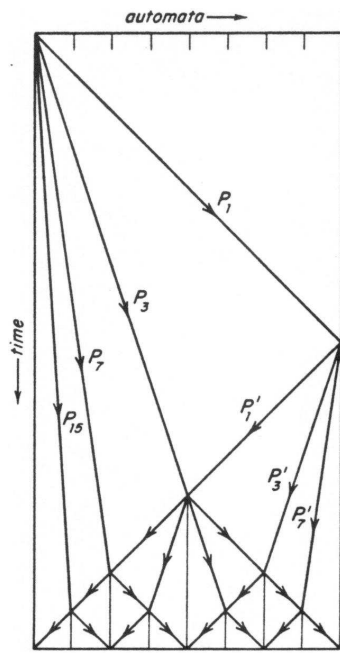
un modèle de calcul correspondant à ces dessins.

1.2 Signaux continus

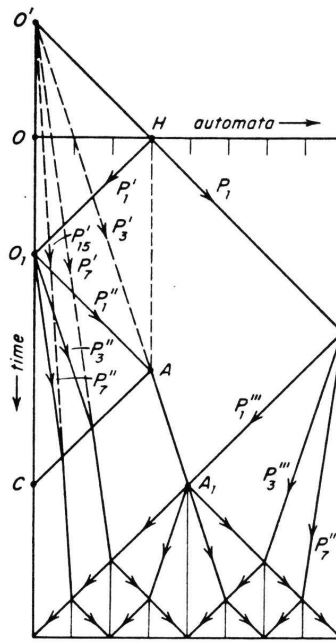
Notre démarche est de prendre ces signaux sur automates cellulaires comme éléments de base et continuer à travailler sans chercher à les discrétiser. C'est une forme d'extension continue des automates cellulaires ou plutôt, la suppression de la discrétisation après une conception continue.

L'espace comme le temps ne sont alors plus discrets mais continus. Une configuration pour nous est un espace affine où sont localisés des signaux. Les signaux sont des instances de méta-signaux. Ces derniers caractérisent complètement les signaux : information véhiculée et vitesse de déplacement. Les signaux se déplacent de manière rectiligne uniforme, de leurs collisions naissent et disparaissent des signaux. Les règles de collisions sont entièrement définies au niveau des méta-signaux. Les diagrammes espace-temps sont des figures géométriques représentant les traces des signaux dans un espace d'une dimension supérieure (pour le temps).

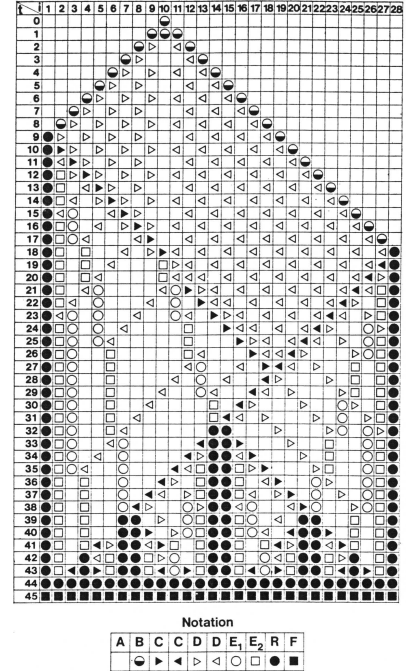
Une autre façon d'envisager le modèle est de considérer des particules en mouvement rectiligne uniforme, dont les collisions sont instantanées et sans dimension.



(a) [VMP70, Fig. 1]



(b) [VMP70, Fig. 2]



(c) [VMP70, Fig. 3]

(Les temps vont de haut en bas)

FIG. 1.6 – Algorithmes géométriques pour la synchronisation de [VMP70].

1.3 État de l’art

À notre connaissance, il n’existe pas de travaux de recherche sur cette approche. Le plus proche que nous connaissons est celui des *automates de Mondrian* de JACOPINI et SONTACCHI [JS90]. Ce modèle considère des polyèdres connexes (avec éventuellement des arêtes / faces internes) de volume fini sans point d’accumulation. Chaque élément du polyèdre (arêtes et faces de toutes dimensions) est coloré. Pour chaque couleur, il y a une boule de référence imposant les couleurs dans le voisinage. Pour être valide, le polyèdre doit, en chaque point, être localement identique à la boule de référence de la couleur du point.

Les auteurs définissent un critère de réversibilité pour les boules qui ne laisse pas de choix pour les autres faces incidentes. Ils montrent que, même en l’imposant, il est possible de définir, pour toute machine de Turing, des boules de référence telles que pour toute entrée de la machine, correspondant à une partie du polyèdre, on ne puisse compléter le polyèdre qu’en réalisant le calcul correspondant. Ce modèle est donc universel pour le calcul au sens de Turing.

Notre modèle est issu de traces de signaux et produit des diagrammes espace-temps qui ne sont pas sans rapport avec ceux de [JS90]. Nous n’utilisons une approche topologique que dans la seconde partie du mémoire et comparons alors les deux modèles (Sous-sect. 8.1.1 du Chap. 8).

1.4 Modèle et résultats

Nous définissons une *machine à signaux* comme un ensemble fini de méta-signaux et de règles correspondant aux collisions de deux ou plus de ces méta-signaux. Chaque signal est caractérisé par le méta-signal dont il est une instance (cela définit sa vitesse). Une configuration est la position de tous les signaux dans un espace affine (\mathbb{R} dans tout le mémoire). Si deux ou plusieurs signaux se rencontrent, ils disparaissent et sont remplacés par de nouveaux signaux suivant la règle correspondante. Par exemple, la rencontre des méta-signaux (bleu, 2) et (rouge, -1) donne (violet, $\frac{3}{2}$).

Nous nommons *diagramme espace-temps* la trace des signaux au cours d'un calcul. Les signaux allant en ligne droite et les mêmes rencontres provoquant les mêmes collisions, on obtient un dessin géométrique régulier. La Figure 1.7 montre un exemple de diagramme avec des instances des trois méta-signaux du paragraphe précédent.

La Figure 1.8 montre des diagrammes beaucoup plus compliqués.

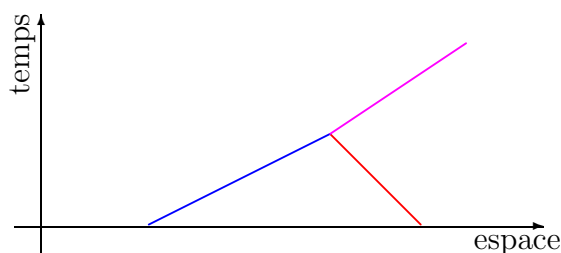


FIG. 1.7 – Exemple de diagramme espace-temps.

Ce modèle est capable de simuler tout automate à deux compteurs pour n'importe quelle entrée. Il est donc universel au sens de Turing.

Les diagrammes étant des dessins, nous avons cherché des opérateurs ayant une signification géométrique. Nous avons montré que l'on pouvait faire, au fil du calcul, des translations et des homothéties avec ou sans gel du calcul. De plus, nous avons pu mettre en place des signaux de manière à réitérer ces opérations, ce qui permet : de restreindre un calcul à une bande d'espace borné et, de contraindre une telle bande à une partie finie de l'espace-temps. Ceci fonctionne uniquement parce que l'espace comme le temps sont continus ; on retrouve l'idempotence de l'espace affine réel et de toute partie bornée.

Cette contraction de n'importe quel diagramme espace-temps à un triangle provoque l'apparition d'un point d'accumulation. Les machines ne sont pas prévues pour gérer les accumulations ; leur cadre n'est pas le plus approprié pour les étudier.

Pour gérer les accumulations, nous considérons les diagrammes espace-temps en tant que tels et les redéfinissons de manière topologique. Nous montrons qu'ils correspondent exactement à ceux définis par les traces des signaux. Équipé de cette caractérisation topologique, nous proposons une typologie pour les singularités isolées ainsi qu'un critère de validité pour les diagrammes en contenant. Le déterminisme peut être perdu en sortie d'accumulation (problème de *l'œuf et la poule*). Nous montrons que la prévision de l'apparition d'un point d'accumulation est Σ_2^0 dans le hiérarchie arithmétique (*i.e.* fortement indécidable).

Il existe de nombreuses directions de recherches pour prolonger ce travail, nous essayons de les présenter en fin de mémoire.

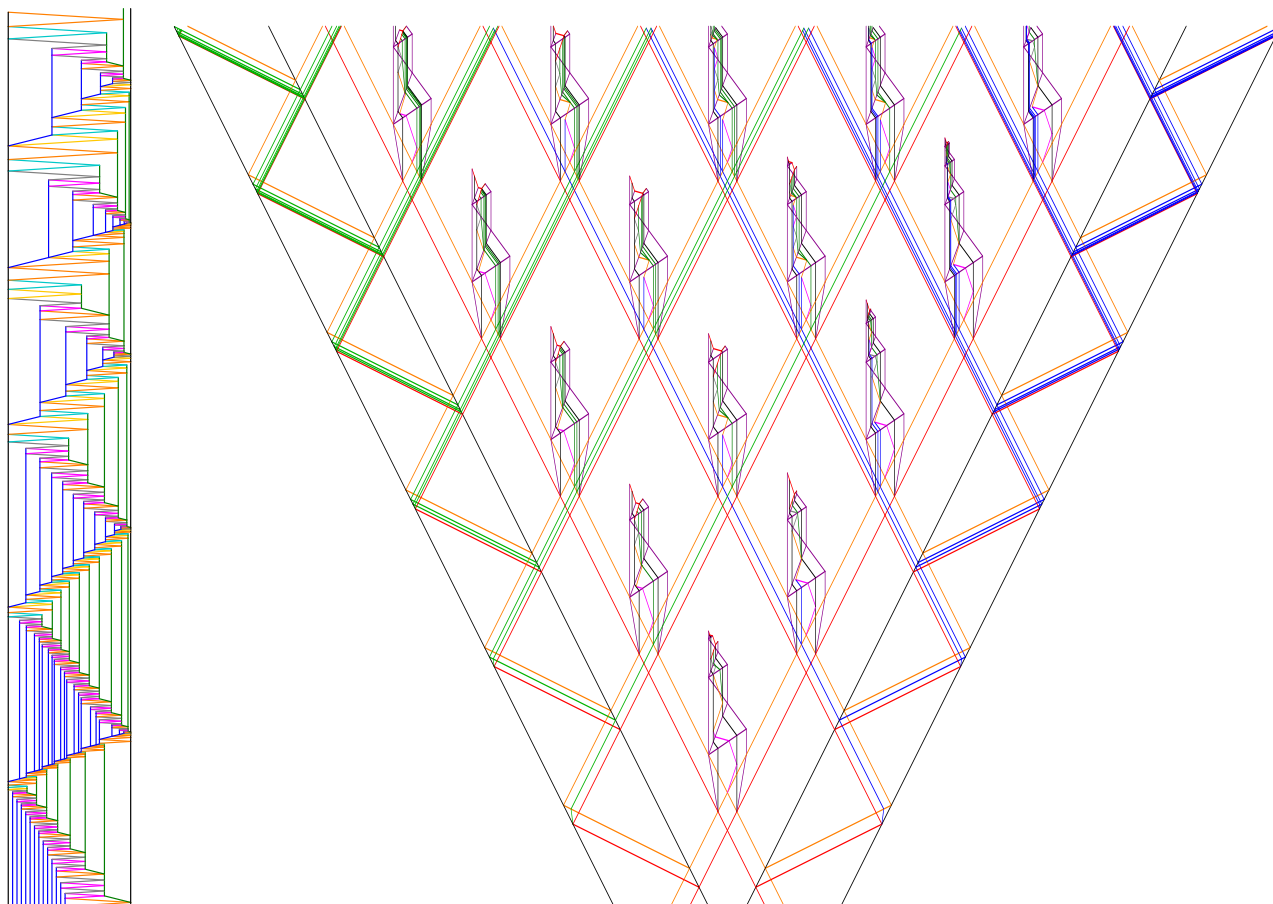


FIG. 1.8 – Diagrammes plus complexes.

1.5 Plan du mémoire

Le mémoire comporte deux parties. La première se concentre sur l'approche machine à signaux, capacité à calculer, modifications géométriques sans points d'accumulation. La seconde porte sur la reformulation topologique, la gestion des singularités et la difficulté de prévision d'une accumulation.

Dans un chapitre préliminaire (numéro 2), nous tentons de définir les concepts de calcul, de continu et de discret. Nous y faisons une classification des modèles existants ce qui nous permet de situer notre modèle. Les machines à signaux sont des modèles à dynamique locale et, à notre connaissance, elles forment le seul modèle à temps et support continus mais à valeurs et mises à jour discrètes.

La Partie I est composée comme suit :

dans le Chapitre 3, sont regroupées toutes les définitions nécessaires : méta-signaux, règles de collision, diagramme espace-temps. . .

dans le Chapitre 4, nous montrons comment simuler tout automate à deux compteurs par une machine à signaux, et prouvons ainsi les capacités à calculer au sens de Turing du modèle ;

dans le Chapitre 5, nous donnons une définition de simulation entre diagrammes espace-temps et montrons quelques simulations simples par des modifications statiques, *i.e.* de la machine ou de la configuration initiale ;

dans le Chapitre 6, nous montrons comment geler un calcul et le mettre en translation. En utilisant le théorème de Thalès, il est possible de contracter un calcul en temps comme en espace. En itérant à l'infini cette contraction, n'importe quel calcul peut être restreint à un espace borné ;

dans le Chapitre 7, nous montrons comment réaliser une homothétie sur une partie du diagramme espace-temps sans pour autant geler le calcul. Nous faisons une construction en abîme qui contracte tout calcul spatialement borné à un triangle borné. Tout calcul peut donc être restreint à un triangle ; nous retrouvons l'idempotence du plan et du triangle, mais nous créons volontairement un point d'accumulation.

Les machines à signaux ne gèrent pas les points d'accumulation. La Partie II propose une approche pour cela. Elle est composée comme suit :

dans le Chapitre 8, nous proposons une définition topologique des diagrammes espace-temps et prouvons qu'elle engendre les mêmes diagrammes que précédemment. Des singularités sont introduites pour définir les points où le diagramme ne peut être défini normalement ;

dans le Chapitre 9, nous établissons une typologie des singularités isolées et proposons un critère de validité pour prolonger les diagrammes espace-temps. Nous montrons avec des exemples que l'on peut perdre le déterminisme ;

dans le Chapitre 10, nous montrons que l'on ne peut prédire l'apparition de points d'accumulation. Ceci est fait en montrant, dans un premier temps, que, pour certaines constructions, leurs apparitions correspondent exactement à l'absence d'arrêt de machines à deux compteurs. Dans un second temps, nous faisons une construction qui a un point d'accumulation si, et seulement si, une machine à deux compteurs ne s'arrête pas pour au moins une entrée. La prédiction est donc Σ_2^0 -difficile. Nous montrons que ce problème est dans Σ_2^0 et est donc Σ_2^0 -complet.

Le dernier Chapitre (numéro 11) conclut le mémoire et propose diverses perspectives de recherche.

L'Annexe A présente le logiciel écrit pour réaliser toutes les figures. Toutes les constructions du mémoire y sont implantées.

L'Annexe B montre des cas où les singularités ne sont pas isolées et présente l'état de notre réflexion sur ce sujet. Elle ne fait pas partie du corps du mémoire car nous ne la considérons pas suffisamment aboutie.