UNIVERSITÉ DU QUÉBEC À MONTRÉAL

Arithmétique et systèmes de réécriture

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES

PAR

PATRICK ST-AMANT

MAI 2007

REMERCIEMENTS

J'aimerais remercier Luc Bélair pour sa précieuse aide, ses conseils judicieux, son temps et sa patience, en somme pour sa direction exemplaire. Je remercie Srecko Brlek pour son support-conseil et pour m'avoir orienté vers la théorie des systèmes de réécriture. Merci à Gisèle Legault pour son soutien technique et à Manon Gauthier pour son agile travail administratif.

Je remercie mon frère Dominic et mes parents Pierre et Pierrette pour leur soutien infaillible et tous les encouragements. Par-dessus tout, merci d'avoir accepté toutes les décisions, bonnes ou mauvaises, que j'ai prises au cours des années. Merci à Gabriel Chevrefils sur qui je peux toujours compter et à Eric Dubreuil pour m'écouter et faire semblant de comprendre mes monologues mathématiques. Finalement, je remercie ma compagne Johanna Okker pour son soutien inconditionnel et pour toute la confiance qu'elle me porte.

TABLE DES MATIÈRES

REMERCIEMENTS iii

TABLE DES FIGURES vii

RÉSUMÉ ix

INTRODUCTION 1

CHAPITRE I
PRÉLIMINAIRES 7

1.1 Théorie des ensembles 7

1.2 Théorie des graphes 8

1.3 Combinatoire des mots 9

CHAPITRE II
SYSTÈMES DE RÉÉCRITURE 11

2.1 Systèmes abstraits de réécritures 11

2.2 Systèmes de réécriture de termes 12

2.3 Confluence 14

2.4 Terminaison 17

CHAPITRE III
L'ARITHMÉTIQUE PAR DES SYSTÈMES DE RÉÉCRITURE 19

3.1 Propriétés 19

3.1.1 Propriétés générales 19

3.1.2 Bases de représentation 20

3.1.3 Mesure de l'augmentation du nombre de règles 21

3.2 Systèmes connus 22

3.2.1 Peano 22

3.2.2 Cohen et Watson 24

3.2.3 Walters 24

3.2.4 Kennaway 26

3.2.5 Walters et Zantema 27

3.2.6 de Vries et Yamada 29

3.2.7 Contejean, Marché et Rabehasaina 29

CHAPITRE IV
DEUX SYSTÈMES DE RÉÉCRITURE POUR L'ARITHMÉTIQUE 31

4.1 Termes et quelques ensembles utiles 31

4.2 Système de réécriture de termes incluant la division pour les nombres rationnels 32

4.2.1 Principe 32

4.2.2 Termes initiaux 33

4.2.3 Le Système $\mathcal{S}_{ASMD}$ 35

4.2.4 Confluence et Terminaison 45

4.3 Système de réécriture de termes orthogonal sans la division pour les entiers relatifs 46

4.3.1 Principe 46

4.3.2 Termes initiaux 46

4.3.3 Le système confluent $\mathcal{S}_{ASM}$ 47

4.4 Bases et nombre de règles 49

4.5 La division et un théorème de Hardy et Wright 50

CHAPITRE V
VARIATIONS 53

5.1 Système de numération mixte pour la mesure du temps 53

5.2 Système de numération exotique 57

5.3 Systèmes qui encodent des sommes d'unités 59

5.3.1 Encodage apparaissant irrégulier 59

5.3.2 Encodage non-confluent 60

5.4 Commutativité 60

CONCLUSION 63

RÉFÉRENCES 65

INDEX 69

TABLE DES FIGURES

0.1 Algue *Anabaena catenula* 3

RÉSUMÉ

Nous présentons un aperçu de la théorie des systèmes de réécriture et un survol de la littérature concernant la modélisation de l'arithmétique par des systèmes de réécriture de termes. Nous proposons deux nouveaux systèmes de réécriture pour l'arithmétique. Le premier système permet l'addition, la soustraction, la multiplication et la division de nombres rationnels pour une base arbitraire. Le deuxième système est un système orthogonal qui permet l'addition, la soustraction et la multiplication de nombres rationnels pour une base arbitraire. De plus, nous présentons un système de réécriture qui modélise un système de numération mixte.

INTRODUCTION

La théorie de la réécriture est à la frontière entre les mathématiques et l'informatique et s'oriente pour l'instant surtout vers des applications calculatoires. Les systèmes de réécriture remontent à Axel Thue qui publia une série d'articles sur la combinatoire des mots. Dans les articles (Thue, 1914) et (Thue, 1910) que l'on retrouve dans (Thue, 1977) Thue étudie le problème du mot. Prenons l'ensemble Σ^* de tous les mots qui peuvent être construit grâces à l'ensemble de symboles $\Sigma = \{a_1, a_2, ..., a_n\}$. Dans sa forme originale, le problème du mot est le suivant. Soit E un ensemble de relations de la forme $w_1 = w_2$ où $w_1, w_2 \in \Sigma^*$, est-il possible de décider si deux mots arbitraires de Σ^* sont égaux ou pas en tenant compte des relations données ? En d'autres termes, existe-t-il un algorithme de décision qui nous informe en un nombre fini d'étapes (ou en un temps fini) si deux mots sont égaux ou pas. Un problème survient si nous utilisons des équations de la forme $w_1 = w_2$, un algorithme pourrait remplacer w_1 par w_2 dans un mot et ensuite remplacer w_2 par w_1 dans le mot et cela indéfiniment. Thue introduisit une orientation dans les équations afin d'éviter ce problème de boucles infinies. Dans son système, une règle de réécriture $w_1 \rightarrow w_2$ signifie que w_1 peut être remplacé par w_2. En choisissant adéquatement l'orientation des équations, il est possible d'avoir pour certains systèmes que si deux mots se réduisent à un même mot alors les deux mots sont égaux. Ainsi, Thue put résoudre le problème du mot pour certains cas particuliers.

Un système de réécriture de mots consiste à se doter de lois de substitution et de les appliquer à des mots. Par exemple, les deux lois $R_1 : AA \rightarrow B$ et $R_2 : BB \rightarrow A$ qui signifient que dans un mot une occurrence AA est remplacée par B et qu'une occurence BB est remplacée par A. En appliquant successivement les deux lois de substitution à partir du mot $BABAA$ nous obtenons une suite de réductions qui se terminent en A

$$BABAA \rightarrow_{R_1} BABB \rightarrow_{R_2} BAA \rightarrow_{R_1} BB \rightarrow_{R_2} A.$$

Depuis Thue, plusieurs types de systèmes de réécriture ont vu le jour et ont été appliqués dans plusieurs domaines. Un exemple d'application est la hiérarchie de Chomsky qui classifie les grammaires formelles (Chomsky, 1956).

Un autre exemple est les systèmes de Lindenmayer nommé *L-systèmes*. Ce sont des systèmes qui se rapprochent de près des systèmes de réécriture et qui sont utilisés pour modéliser les végétaux (Prusinkiewicz et Lindenmayer, 1991). En particulier Lindemayer (Lindenmayer, 1968) décrit la croissance des algues *Anabaena catenula* (voir la figure 0.1[1]) en utilisant un L-système. Sachant que A et B sont des types de cellules et que les indices g et d indiquent la *polarité*[2] de la cellule, le développement de l'algue *Anabaena catenula* est régi par le système de règles suivant :

$$
\begin{aligned}
A_d &\rightarrow A_g B_d \\
A_g &\rightarrow B_g A_d \\
B_d &\rightarrow A_d \\
B_g &\rightarrow A_g
\end{aligned}
$$

À chaque étape, chaque symbole A_d, A_g, B_d et B_d est remplacé respectivement par $A_g B_d$, $B_g A_d$, A_d et A_g. En partant d'un terme initial A_d, on trouve la suite suivante d'étapes décrivant la croissance de l'algue :

$$
\begin{aligned}
&A_d \\
&A_g B_d \\
&B_g A_d A_d \\
&A_g A_g B_d A_g B_d \\
&B_g A_d B_g A_d A_d B_g A_d A_d \\
&\ldots
\end{aligned}
$$

---

[1]Photographie provenant de *http://cage.rug.ac.be/~bh/L-systemen/sigma_xi_3.html*

[2]La polarité est une caractéristique fondamentale des cellules, qui régit de très nombreuses fonctions biologiques. Par exemple, la mise en place rapide de l'axe antéro-postérieur de l'oeuf fécondé permet l'orientation du futur embryon.

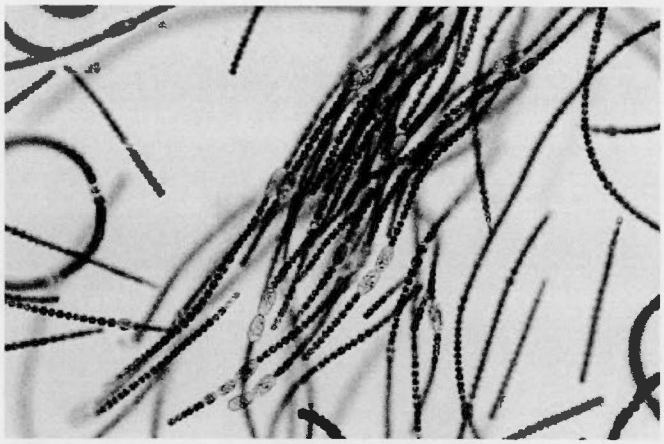

Fig. 0.1 Algue *Anabaena catenula*

La théorie de réécriture de termes possède aussi beaucoup d'applications dans la théorie de la démonstration qui cherche à automatiser la construction des preuves. Un sous-domaine de la théorie de la réécriture qui possède des applications dans la théorie de la démonstration se nomme *réécriture équationnelle*. Ce domaine est un peu l'extension des travaux de Thue sur le problème du mot. La théorie de la démonstration revient souvent à manipuler des équations. Puisque des équations peuvent engendrer des boucles infinies, la théorie de la réécriture équationnelle permet d'éviter ceci en orientant les équations. De plus, en transformant des équations telles que $x + 0 = x$ en règles de réécriture $x + 0 \rightarrow x$, cela permet d'éviter des expression comme $((x + 0) + 0) + 0$ et ainsi d'utiliser moins d'espace mémoire de l'ordinateur. Pour une introduction détaillée au sujet de la réécriture équationnelle se référer à (Plaisted, 1993) et au chapitre 7 de (TeReSe, 2003).

Déjà en 1929, on retrouve dans la thèse de doctorat de Herbrand (Herbrand, 1930) un système de réécriture qui possède la propriété qu'une formule propositionnelle est une tautologie si et seulement si la formule peut être réduite à 1 (*vrai*) en utilisant les règles du système modulo l'associativité et la commutativité. Modulo associativité et commutativité signifie que l'ordre des produits et les parenthèses peut être réarrangé. Voici le système de réécriture de Herbrand :

$$
\begin{aligned}
x \Rightarrow y &\rightarrow x \cdot y + x + 1 \\
x \vee y &\rightarrow x \cdot y + x + y \\
\neg x &\rightarrow x + 1 \\
x + 0 &\rightarrow x \\
x + x &\rightarrow 0 \\
x \cdot 0 &\rightarrow 0 \\
x \cdot 1 &\rightarrow x \\
x \cdot x &\rightarrow x \\
x \cdot (y + z) &\rightarrow x \cdot y + x \cdot z
\end{aligned}
$$

Dans (TeReSe, 2003) il est montré que $p \Rightarrow (q \Rightarrow p)$ se réduit à 1. Ici, nous allons présenter un autre exemple.

Exemple 1. La tautologie $p \vee \neg p$ se réduit à 1 en appliquant les règles du système de Herbrand.

$$
\begin{aligned}
p \vee \neg p &\rightarrow p \cdot (\neg p) + p + \neg p \\
&\rightarrow p \cdot (p + 1) + p + \neg p \\
&\rightarrow p \cdot (p + 1) + p + p + 1 \\
&\rightarrow p \cdot p + p \cdot 1 + p + p + 1 \\
&\rightarrow p \cdot p + p + p + p + 1 \\
&\rightarrow p + p + p + p + 1 = p + p + (p + p) + 1 \\
&\rightarrow p + p + 0 + 1 = p + (p + 0) + 1 \\
&\rightarrow p + p + 1 = (p + p) + 1 \\
&\rightarrow 0 + 1 = 1 + 0 \\
&\rightarrow 1
\end{aligned}
$$

Le but de ce mémoire est de montrer qu'il est possible d'utiliser la théorie des systèmes de réécriture afin de modéliser l'arithmétique et les systèmes de numération. L'arithmétique élémentaire qui inclut les opérations d'addition, de soustraction, de multiplication et de division est normalement étudiée grâce à la théorie des anneaux. Dans ce contexte, l'arithmétique est comprise comme étant un anneau commutatif qui est

défini par quelques axiomes (Dummit et Foote, 1999). Les systèmes de numération sont une partie inhérente aux mathématiques, par exemple la base binaire, la base décimale sont les bases les plus fréquemment utilisées. En plus des systèmes de numération standards, il existe plusieurs systèmes non-standards. Un exemple de base non-standard est la mesure du temps.

Nous effectuons un bref survol de la littérature et nous présentons plusieurs systèmes permettant de modéliser l'arithmétique et les systèmes de numération. Ainsi, nous aurons un nouveau regard sur une théorie qui est souvent traitée avec l'aide de l'algèbre abstraite et nous pourrons définir de nouveaux types de systèmes de numération. Pour l'instant, la théorie des systèmes de réécriture appliquée à l'arithmétique ne semble pas donner des algorithmes plus puissants que les algorithmes utilisant la virgule flottante que nous retrouvons au coeur de l'architecture des ordinateurs modernes. La théorie de la réécriture permet néanmoins de voir l'arithmétique d'une manière différente et probablement plus naturelle.

Le chapitre 2 est un survol des définitions de base de la théorie des systèmes de réécriture. Nous présentons au chapitre 3 les propriétés que nous pouvons attribuer aux systèmes de réécriture modélisant l'arithmétique ainsi qu'un survol des systèmes déjà connus. Au chapitre 4 nous présentons un nouveau système de réécriture qui permet l'addition, la soustraction, la multiplication et la division de nombres rationnels pour une base arbitraire et nous présentons un autre système qui permet l'addition, la soustraction, la multiplication d'entiers pour une base arbitraire. Ensuite, nous étudions les propriétés de ces systèmes. Dans le chapitre 5 nous verrons que les systèmes de réécriture nous permettent aussi de modéliser un système du numération mixte.

CHAPITRE I

PRÉLIMINAIRES

Pour consultation, nous donnons dans ce qui suit une série de définitions qui sont utiles pour la lecture de ce mémoire.

1.1 Théorie des ensembles

Voici quelques définitions concernant les relations. Pour plus de détails se référer par exemple à l'appendice A de (TeReSe, 2003). Le symbole $\mathbb{N}$ désigne l'ensemble des nombres naturels.

Définition 2. *Si $R \subseteq A \times B$ pour A et B des ensembles, alors nous appelons R une* relation *et nous écrivons aRb pour signifier que $(a, b) \in R$.*

Définition 3. *Soit S un ensemble et $R \subseteq S \times S$ une relation sur S.*

1. *Une* suite *dans S est une fonction $s : \mathbb{N} \to S$.*
2. *Une suite s est R-*descendante *si $s(i+1)Rs(i)$ pour tout $i \in \mathbb{N}$.*
3. *Une suite finie de longueur n dans S est une fonction $s : \{0, 1, ..., n-1\} \to S$, où $n \in \mathbb{N}$.*
4. *R est une relation* bien-fondée *s'il n'existe pas de suites R-descendantes infinies dans S.*
5. *R est* irréflexive *si pour tout $x \in S$ ce n'est jamais le cas que xRx.*
6. *R est* transitive *si pour tout $x, y, z \in S$ nous avons que xRy et yRz implique xRz.*
7. *R est une relation d'ordre partiel strict si R est irréflexive et transitive.*

8. *R est une relation d'*ordre bien-fondée *si R est une relation d'ordre partiel strict bien-fondée.*
9. *La* fermeture transitive R^+ *de R est la relation* $\cap\{R' \subseteq S \times S | R \subseteq R'$ *et* R' *est transitive*$\}$. *En d'autres mots, c'est la plus petite relation transitive sur S qui contient R.*
10. *La* fermeture réflexive $R^=$ *de R est la relation* $\cap\{R' \subseteq S \times S | R \subseteq R'$ *et* R' *est réflexive*$\}$.

1.2 Théorie des graphes

Voici quelques définitions de la théorie des graphes qui nous seront utiles à la section 3.2.4. Pour plus de détails se référer à (Labelle, 1980).

Définition 4. *Un* graphe simple, *noté* $G(X, A)$, *est formé de deux ensembles : un ensemble fini non-vide* X, *appelé l'ensemble des* sommets *de* G, *et un ensemble* A *de paires de sommets, appelé l'ensemble des* arêtes *de* G, *où* $A \subseteq \{\{u, v\} | u \neq v$ *et* $u, v \in X\}$.

Définition 5. *Dans un graphe simple, une* chaîne *est une séquence finie de sommets,* $x_0, x_1, ..., x_m$, *telle que pour tout* $0 \leq i < m$ *on a que* $\{x_i, x_{i+1}\}$ *est une arête. L'entier* m *s'appelle la* longueur *de la chaîne. Si* $x_0 = x_m$ *alors on appelle cette chaîne un* cycle. *Une chaîne de longueur plus grande que* 1 *dont toutes les arêtes* $\{x_i, x_{i+1}\}$ *sont distinctes est dite* simple.

Définition 6. *Un graphe simple est* connexe *si pour tout* x *et* y, *il existe une chaîne de* x *à* y.

Définition 7. *Une* forêt *est un graphe simple sans cycle simple.*

Définition 8. *Un* arbre *est une forêt connexe.*

Définition 9. *Une* arborescence *est un arbre dont un sommet, appelé* racine, *a été distingué.*

Définition 10. *Soit $x \in X$ un sommet d'un graphe simple. Le* degré *de x est la cardinalité de l'ensemble $\{y \in X | \{x, y\}$ est une arête$\}$.*

Définition 11. *Une sommet de degré* 1 *est appelé une* feuille.

Définition 12. *Le* parent *d'un sommet dans une arborescence est le premier sommet de la chaîne qui va du sommet à la racine.*

Définition 13. *Les* descendants *d'un sommet dans une arborescence sont tous les sommets atteignables par une chaîne qui ne passe pas par le sommet parent.*

Définition 14. *On dit que $H(Y, B)$ est un* sous-graphe *de $G(X, A)$ si $Y \subseteq X$ et $B \subseteq A$.*

1.3 Combinatoire des mots

Voici quelques définitions concernant la combinatoire des mots qui nous seront utiles au chapitre 4.

Définition 15. *Un* alphabet *est un ensemble fini de symboles.*

Définition 16. *Un* mot *sur l'alphabet Σ est une suite finie de symboles de Σ.*

Définition 17. *La* longueur *d'un mot est le nombre de symboles composant le mot.*

Définition 18. *Le* mot vide, *noté ε, est l'unique mot de longueur* 0.

Définition 19. *Soit Σ un alphabet et $k \in \mathbb{N}$, alors Σ^k est l'ensemble de tous les mots de longueur k sur Σ.*

Définition 20. *Soit Σ un alphabet, alors la* fermeture de Kleene *de Σ, notée Σ^*, est l'ensemble de tous les mots sur Σ de toutes les longueurs. En fonction de Σ^k, on écrit $\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$.*

CHAPITRE II

SYSTÈMES DE RÉÉCRITURE

Pour une introduction plus complète aux systèmes de réécriture, on peut se référer à (Klop, 1992), (Dershowitz et Plaisted, 2001) et (Cirstea, 2000).

2.1 Systèmes abstraits de réécritures

Définition 21. *Un* système abstrait de réécriture *est défini comme étant une structure* $\mathcal{A} = \langle A, \{\rightarrow_\alpha \mid \alpha \in I\}\rangle$ *où* A *est un ensemble d'objets,* I *est un ensemble d'indices et* $\rightarrow_\alpha$ *est une relation binaire sur* A.

Si $(a, b) \in \rightarrow_\alpha$ alors on écrit $a \rightarrow_\alpha b$. En identifiant les objets de A par des sommets, il est aussi possible de représenter un système abstrait de réécriture sous forme de graphe orienté. Notons que la définition formelle d'une relation est donnée à la définition 2.

Exemple 22. Si nous prenons l'exemple de l'Introduction, avec les indices R_1 et R_2, nous avons

$$(BABAA, BABB), (BAA, BB) \in \rightarrow_{R_1}$$

$$(BABB, BAA), (BB, A) \in \rightarrow_{R_2} .$$

Nous pouvons aussi définir la relation binaire $\rightarrow := \rightarrow_{R_1} \cup \rightarrow_{R_2}$ et nous aurons que

$$(BABAA, BABB), (BAA, BB), (BABB, BAA), (BB, A) \in \rightarrow .$$

Cette remarque est très utile si nous voulons considérer des systèmes contenant plusieurs règles.

D'autres exemples de système abstrait de réécriture sont le calcul lambda (voir le chapitre 10 de (TeReSe, 2003)) et les systèmes de réécriture de termes que nous allons aborder dans les définitions suivantes.

2.2 Systèmes de réécriture de termes

Définition 23. *$\mathcal{V}$ est un ensemble dénombrable de variables notées $x, y, z, x_1, x_2, ...$*

Définition 24. *Une* signature *$\mathcal{F}$ est un ensemble non-vide de symboles de fonction chacune possédant une certaine arité.*

Notons qu'une autre notation que la notation plaçant le symbole de fonction devant l'expression peut être utilisée, par exemple l'addition peut être dénotée par le symbole $+$ et être positionnée à l'intérieur de l'expression.

Définition 25. *On définit l'ensemble des* termes *$\mathcal{T}(\mathcal{F}, \mathcal{V})$ comme suit :*

1. *$\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$;*
2. *si $t_1, ..., t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ et si $f \in \mathcal{F}$ est d'arité n avec $n \geq 0$ alors $f(t_1, ..., t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. Par cette notation nous considérons que les constantes (cas où $n = 0$) sont dans $\mathcal{T}(\mathcal{F}, \mathcal{V})$.*

Définition 26. *Une* règle de réécriture *est un couple, noté $l \to r$, avec $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ qui respecte les deux conditions suivantes :*

1. *Le membre de gauche l n'est pas une variable ;*
2. *Les variables du membre de droite apparaissent dans le membre de gauche.*

Une règle de réécriture peut être nommée. Par exemple si nous voulons nommer la règle de réécriture ρ nous écrivons $\rho : l \to r$.

Définition 27. *Une* substitution *σ est une fonction de $\mathcal{T}(\mathcal{F}, \mathcal{V})$ dans $\mathcal{T}(\mathcal{F}, \mathcal{V})$ telle que $\sigma(F(t_1, ..., t_n)) = F(\sigma(t_1), ..., \sigma(t_n))$ pour tout symbole de fonction F d'arité $n \geq 0$. Nous dirons que $\sigma(t)$ est une σ-instance de t.*

Ainsi nous voyons qu'une substitution affecte seulement les variables.

Définition 28. *Un* contexte *est un terme sur une signature étendue* $\mathcal{F} \cup \{\Diamond\}$. *En d'autres mot, c'est un terme contenant zéro, un ou plusieurs symboles* $\Diamond$ *qui dénote des* trous.

Si C est un contexte contenant n trous, et $t_1, t_2, ..., t_n$ sont des termes, alors $C[t_1, ..., t_n]$ dénote le résultat du remplacement des trous de C de gauche à droite par $t_1, ..., t_n$.

Définition 29. *Nous appelons* redex *la partie gauche d'une règle de réécriture où l'on a appliqué une substitution. Si nous avons la règle* $\rho : l \to r$ *et une substitution* σ *nous dirons que* $\sigma(l)$ *est un* ρ*-redex.*

Voici un exemple illustrant ce qu'est un redex.

Exemple 30. Soit une règle $\rho : ax \to c$ avec x une variable et $a, b, c \in \mathcal{F}$ des symboles de fonction d'arité 0, soit la substitution $\sigma(x) = b$ alors nous disons que ab est un ρ-redex.

Définition 31. *Un* système de réécriture de termes *est une structure* $\langle \mathcal{F}, \mathcal{R} \rangle$, *où* $\mathcal{F}$ *est une signature et* $\mathcal{R}$ *est un ensemble de règles de réécriture.*

Un système de réécriture de termes sera dans la plupart des cas spécifié par son ensemble de règles en considérant la signature et l'ensemble des variables comme étant uniquement composé des symboles utilisés dans $\mathcal{R}$.

Définition 32. *On dit que* $s \to t$ *est une* étape de réécriture *s'il y a une règle de réécriture* $u : l \to r \in \mathcal{R}$, *une substitution* σ *et un contexte* $C[]$ *tel que* $s = C[\sigma(l)]$ *et* $t = C[\sigma(r)]$. *Si nous voulons être précis, nous écrivons* $s \to_u t$.

Exemple 33. Par exemple, soit σ une substitution définie par $\sigma(x) = b$, $\sigma(a) = a$ et $\sigma(c) = c$, $\mu : ax \to c$ une règle du système et $cc\Diamond b$ un contexte, alors nous aurons que $ccabb \to cccb$ est une étape de réécriture, puisque $s = cc\sigma(ax)b = cc\sigma(a)\sigma(x)b = ccabb$ et $t = cc\sigma(c)b = cccb$.

Définition 34. *Nous dirons que* $a_1 \rightarrow_{\rho_1} a_2 \rightarrow_{\rho_2} a_3 \rightarrow_{\rho_3} \ldots$ *est une* suite d'étapes de réécriture indexées *si pour tout* $i \in \mathbb{N}^+$, *nous avons que* $a_i \rightarrow_{\rho_i} a_{i+1}$ *est une étape de réécriture, avec* $\rho_i \in \mathcal{R}$ *et* a_i *des termes du système. Nous pouvons écrire* $a \rightarrow_{\mathcal{R}} b$ *lorsque pour une règle de réécriture* $\rho \in \mathcal{R}$ *nous avons* $a \rightarrow_{\rho} b$ *. En pratique, lorsque la situation est claire, les indices seront omis.*

Exemple 35. Afin de fixer les idées prenons un autre exemple tiré de (Baader et Nipkow, 1998). Soient les trois règles suivantes :

$$R_1 : f(x, f(y, z)) \rightarrow f(f(x, y), z)$$
$$R_2 : f(e, x) \rightarrow x$$
$$R_3 : f(i(x), x) \rightarrow e$$

Alors, partant du terme $f(i(e), f(e, e))$, voici un exemple d'une suite d'étapes de réécriture indexées :

$$f(i(e), f(e, e)) \rightarrow_{R_1} f(f(i(e), e), e) \rightarrow_{R_3} f(e, e) \rightarrow_{R_2} e.$$

2.3 Confluence

Deux notions importantes des systèmes de réécriture sont celles de forme normale et de confluence. Voici un exemple informel de l'arithmétique usuelle qui illustre ces deux concepts.

Exemple 36. Il y plusieurs manières de "réduire" l'expression $(3 + 4) \cdot (7 + 9)$ à son résultat 112. Par exemple

$$(3 + 4) \cdot (7 + 9) \rightarrow 7 \cdot 7 + 7 \cdot 9 \rightarrow 7 \cdot 7 + 63 \rightarrow 49 + 63 \rightarrow 112$$

$$(3 + 4) \cdot (7 + 9) \rightarrow 7 \cdot (7 + 9) \rightarrow 7 \cdot 16 \rightarrow 112$$

De manière informelle la confluence assure que partant d'une même expression, toutes les manières de calculer arriveront au même résultat final. Ici, puisque 112 ne peut pas être "réduit" davantage, il serait donc considéré comme une forme normale.

Voici les définitions formelles.

Définition 37. *Soit $\mathcal{A} = \langle A, (\rightarrow_\alpha)_{\alpha \in I} \rangle$ un système abstrait de réécriture, $a \in A$ est une* forme normale *si il n'existe aucun $b \in A$ tel que $a \rightarrow b$.*

La propriété qui assure que n'importe laquelle des suites d'étapes de réécriture entraîne la même forme normale est appelée l'unicité des formes normales. La confluence assure l'unicité des formes normales. Ainsi, deux suites de réduction étant appliquées à un même terme initial se termineront à une même forme normale.

Nous écrirons $a \twoheadrightarrow b$ lorsqu'il y a une suite finie telle que $a \rightarrow c_1 \rightarrow c_2 \rightarrow ... \rightarrow b$.

Définition 38. *Soit $\mathcal{A} = \langle A, \rightarrow \rangle$ un système abstrait de réécriture.*

(i) $a \in A$ *est* confluent *si pour tous $b, c \in A$ il existe un $d \in A$ tel que $a \twoheadrightarrow c$ et $a \twoheadrightarrow b$ entraînent $c \twoheadrightarrow d$ et $b \twoheadrightarrow d$.*

(ii) La relation $\rightarrow$ possède la propriété Church-Rosser si tout $a \in A$ est confluent.

(iii) $a \in A$ *est* faiblement confluent *si pour tous $b, c \in A$ il existe un $d \in A$ tel que $a \rightarrow c$ et $a \rightarrow b$ entraînent $c \twoheadrightarrow d$ et $b \twoheadrightarrow d$.*

(iv) La relation $\rightarrow$ possède la propriété faiblement Church-Rosser si tout $a \in A$ est faiblement confluent.

Définition 39. *Si $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ peut être écrit sous la forme $C[s]$ alors nous disons que s est un* sous-terme *de t.*

La définition suivante raffine la notion de sous-terme en permettant sa multiplicité.

Définition 40. *Une* occurrence *d'un sous-terme s dans un terme t se définit comme un couple ordonné $\langle s \mid C[] \rangle$ tel que $C[s] \equiv t$.*

Par exemple, si nous avons un terme $A(M(S(0), 0), S(0))$, nous voyons que le sous-terme $S(0)$ apparaît deux fois. Nous avons donc les occurrences $\langle S(0) \mid A(M(\Diamond, 0), S(0)) \rangle$ et $\langle S(0) \mid A(M(S(0), 0), \Diamond) \rangle$.

Définition 41. *Le* patron *d'une règle de réécriture $\rho : l \to r$ est défini comme un contexte l^{ϵ}, où ϵ est la substitution $\epsilon(x) = \Diamond$ pour toute variable, c'est-à-dire que nous remplaçons les variables par des trous.*

Définition 42. *Deux occurrences de redex dans un terme t se* chevauchent *si leurs patrons partagent au moins une occurrence d'un symbole.*

Définition 43. *Deux règles de réécriture ρ_1, ρ_2 se* chevauchent *si pour un terme t il y a des occurrences de ρ_1-redex s_1 et ρ_2-redex s_2 telles que s_1 et s_2 se chevauchent.*

Exemple 44. Soit le système de réécriture de termes avec les règles

$$F(G(x, S(0)), y, H(z)) \to x$$
$$G(H(x), S(y)) \to y$$

Dans le terme $F(G(H(0), S(0)), y, H(G(x, x)))$ nous avons un ρ_1-redex qui est le terme au complet et un ρ_2-redex qui est le terme $G(H(0), S(0))$. Puisque le patron du ρ_1-redex partage au moins une occurrence de symbole avec le patron du ρ_2-redex, nous avons que les deux règles se chevauchent.

Nous remarquons dans l'exemple précédent que non seulement il y a un chevauchement entre les deux redex, mais aussi que le ρ_1-redex contient le ρ_2-redex. Il est possible de démontrer que si deux redex se chevauchent alors un des redex contient l'autre. La notion précise de "contient" est donnée par la définition suivante.

Définition 45. *Nous disons que $\langle s' \mid C'[]\rangle$ est contenu dans $\langle s \mid C[]\rangle$, noté $\langle s' \mid C'[]\rangle \leq \langle s \mid C[]\rangle$, si pour un contexte $D[]$ nous avons $C'[] \equiv C[D[]]$.*

Définition 46. *Une règle est dite* linéaire-gauche *si aucune variable n'apparaît deux fois dans la partie gauche de la règle de réécriture.*

Maintenant, il est possible de définir ce qu'est un système orthogonal.

Définition 47. *Soit un système de réécriture de termes $\mathcal{S}$.*

(i) $\mathcal{S}$ est non-chevauchant s'il n'y a pas de chevauchement entre chaque règle de réduction.

(ii) $\mathcal{S}$ est linéaire-gauche si toutes les règles sont linéaire-gauches.

(iii) $\mathcal{S}$ est orthogonale *si il est non-chevauchant et linéaire gauche.*

Nous trouvons dans (TeReSe, 2003) plusieurs preuves du théorème suivant initialement démontré par Rosen en 1973.

Théorème 48. *Tout système de réécriture orthogonal est confluent.*

Nous nous servirons de ce théorème dans le deuxième système de réécriture de terme présenté au chapitre 4.

2.4 Terminaison

Définition 49. *Soit $\mathcal{A} = \langle A, \rightarrow \rangle$ un système abstrait de réécriture. Nous dirons que $a \in A$ est* fortement normalisant *si toute suite de réduction commençant par a est une suite finie. La relation $\rightarrow$ est fortement normalisante ou possède la* propriété de terminaison *si tout a est fortement normalisant.*

La proposition suivante permet de voir la propriété de terminaison comme une relation d'ordre bien-fondée.

Proposition 50. *Un système de réécriture de termes $\langle \mathcal{F}, \mathcal{R} \rangle$ possède la propriété de terminaison si et seulement si il existe une relation d'ordre bien-fondée sur $\mathcal{T}(\mathcal{F}, \mathcal{V})$, notée $>$, qui est telle que $t > u$ pour tous $t, u \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ satisfaisant $t \rightarrow_R u$.*

Démonstration. Si $\langle \mathcal{F}, \mathcal{R} \rangle$ possède la propriété de terminaison alors nous choisissons comme relation d'ordre bien-fondée $>$ la relation $\leftarrow_{\mathcal{R}}^{+}$ qui est la fermeture transitive de $\leftarrow_{\mathcal{R}}$. Inversement, si $>$ satisfait les conditions alors, puisque la relation est bien-fondée, nous avons que $\langle \mathcal{F}, \mathcal{R} \rangle$ possède la propriété de terminaison. □

Puisque la plupart des systèmes de réécriture de termes possèdent une infinité d'éléments, la proposition ci-dessus est rarement utilisée. La proposition que l'on retrouve ci-dessous est plus utile, car normalement, un système de réécriture de termes possède un nombre fini de règles. Notons que la preuve de cette proposition se trouve à la section 6.1 de (TeReSe, 2003).

Définition 51. *Un* ordre de réduction *sur $\mathcal{T}(\mathcal{F},\mathcal{V})$ est une relation d'ordre $<$ bien-fondé sur $\mathcal{T}(\mathcal{F},\mathcal{V})$ qui satisfait les deux conditions suivantes pour tous $t,u \in \mathcal{T}(\mathcal{F},\mathcal{V})$:*

(i) si $t < u$ et σ est une substitution arbitraire alors $t^\sigma < u^\sigma$.

(ii) si $t < u$ et C est un contexte arbitraire alors $C[t] < C[u]$.

Définition 52. *Un ordre de réduction $<$ sur $\mathcal{T}(\mathcal{F},\mathcal{V})$ est* compatible *avec un système de réécriture de termes $\langle \mathcal{F},\mathcal{R}\rangle$ si $r < l$ pour toute règle de réécriture $l \to r$ dans $\mathcal{R}$*

Proposition 53. *Un système de réécriture de termes $\langle \mathcal{F},\mathcal{R}\rangle$ possède la propriété de terminaison si et seulement si il admet un ordre de réduction compatible sur $\mathcal{T}(\mathcal{F},\mathcal{V})$.*

La difficulté lorsque nous voulons prouver la propriété de terminaison d'un système de réécriture de termes est qu'il n'est pas toujours évident de trouver une relation d'ordre partielle. Nous trouvons dans (TeReSe, 2003) plusieurs techniques.

CHAPITRE III

L'ARITHMÉTIQUE PAR DES SYSTÈMES DE RÉÉCRITURE

Les systèmes de réécriture procurent une autre approche de l'arithmétique, particulièrement une approche différente de l'algèbre abstraite. Dans ce qui suit, nous discutons des propriétés des systèmes de réécriture de termes modélisant l'arithmétique et introduisons à la section 3.1.3 des outils qui nous permettront d'évaluer les systèmes présentés au chapitre 4. De plus, nous faisons un survol des systèmes de la littérature. Il faut noter qu'à part l'article de Contejean, Marché et Rabehasaina (Contejean, Marché et Rabehasaina, 1997), tous les systèmes de la littérature ont pour but principal de représenter l'arithmétique sous la forme de systèmes de réécriture de termes.

3.1 Propriétés

Chaque système de réécriture possède ses avantages et inconvénients. Il est bien-fondé d'évaluer un système par les propriétés qu'il possède.

3.1.1 Propriétés générales

Les propriétés habituellement recherchées pour un système de réécriture modélisant l'arithmétique sont la confluence et la propriété de terminaison. Il y a aussi la propriété d'adéquation, qui signifie que le système de réécriture de termes représente bien l'arithmétique. Une propriété parfois mentionnée est le nombre de règles définissant le système. Une mesure de complexité proposée dans (Walters et Zantema, 1995) est le

nombre d'étapes de réduction entre un terme et sa forme normale. Un système pouvant être lu aisément par une personne est une propriété souvent recherchée, car un système ayant cette propriété représente plus fidèlement l'arithmétique usuelle. Une propriété provenant de l'informatique est la quantité d'espace requis pour effectuer les opérations. L'espace devient un facteur important surtout dans le système que nous construisons au chapitre 4. Une propriété qui serait intéressante à introduire serait la moyenne du nombre de réductions entre un terme initial et un terme final (terme sous forme normale) pour les termes ayant n symboles.

Le nombre d'opérations (addition, soustraction, multiplication et division) qu'un système représente est une propriété importante. Il faut mentionner qu'un système qui inclut l'addition de nombres négatifs et n'inclut pas la soustraction est différent d'un système qui inclut l'addition de nombres négatifs et inclut la soustraction. Il en est de même pour les nombres rationnels et la division, car il peut y avoir des systèmes où l'on peut manipuler des fractions p/q sans pouvoir diviser un entier p par un entier q.

3.1.2 Bases de représentation

Les bases pouvant être utilisées dans le système de réécriture de termes constituent une autre propriété. L'arithmétique de Peano que nous verrons dans ce qui suit sous forme de système de réécriture de termes procure une arithmétique en base 1. Pour d'autres bases, la notation positionnelle permet d'écrire les nombres en base b sous la forme

$$(a_n a_{n-1} \ldots a_1 a_0, c_1 c_2 \ldots)_b = \sum_{k=0}^{n} a_k b^k + \sum_{k=1}^{\infty} c_k b^{-k}$$

Il existe aussi des bases mixtes où la base peut changer à chaque position. Un exemple est la mesure du temps qui est mesuré en semaines, jours, heures, minutes et secondes. Ainsi, 828225 secondes en base dix s'écrit dans la base mixte donnée comme

1 semaine 2 jours 14 heures 3 minutes 45 secondes

3.1.3 Mesure de l'augmentation du nombre de règles

Nous introduisons une nouvelle propriété qui est intéressante à considérer et qui est à la base du système du chapitre 4. Nous définissons ce que nous appellerons *suite r-imbriquée* et *fonction de progression.* Ces deux notions nous aideront à mesurer l'augmentation du nombre de règles. Il ne suffit pas de calculer le nombre de règles entre deux systèmes puisqu'il faut tenir compte de la similitude entre les deux systèmes.

Définition 54. *Soit une suite d'ensembles* $\{A_1, A_2, ..., A_n, A_{n+1}, ...\}$. *Nous dirons que la suite est une* suite r-imbriquée *si* $A_i^r \subset A_{i+1}$ *où l'exposant* r *signifie que* r *éléments ont étés retirés à l'ensemble* A_i, *avec* r *ne pouvant dépasser le cardinal de* A_i.

Le r est une sorte de mesure de proximité pour les ensembles de la suite.

Définition 55. *Soit une suite d'ensembles* $\{B_1, B_2, ..., B_n, B_{n+1}, ...\}$. *La* fonction de progression f *d'une suite est la fonction définie par* $f(i) = |B_{i+1}|$ *pour tout* i, *avec* $|\ |$ *la notation usuelle du cardinal d'un ensemble.*

Soit $A_b = \langle \mathcal{F}_b, \mathcal{R}_b \rangle$ un système de réécriture de termes représentant l'arithmétique pour la base de représentation $b \in \mathbb{N}^+$. Prenons la suite $\{R_1, R_2, ..., R_b, R_{b+1}, ...\}$ des ensembles de règles décrivant les systèmes de réécriture de termes. Une propriété intéressante serait d'avoir une suite r-imbriquée de règles où r est minimal et d'avoir une fonction telle que la dérivée de f par rapport à i est minimale. Ces concepts nous informent des changements apportés afin de passer d'un ensemble de règles à un autre.

Nous verrons au chapitre 4 un système incluant la division où la suite des ensembles de règles ordonnées selon la base est une suite 1-imbriquée et telle que la fonction de progression de la suite est donnée par $f(i) = 2i + k$ pour k un entier positif, ce qui donne $f'(i) = 2$.

3.2 Systèmes connus

3.2.1 Peano

Grâce aux axiomes de Peano il est possible de formaliser l'arithmétique. En identifiant le successeur de zéro par $s(0)$ et le successeur de $s(0)$ par $s(s(0))$ et ainsi de suite nous avons une représentation pour tous les entiers positifs. Pour plus de détails à propos des axiomes de Peano se référer au chapitre 6 de (Cori et Lascar, 2003).

Soit $\mathcal{L}$ le langage du premier ordre comportant le symbole de constante 0, le symbole de fonction unaire S, le symbole de fonction binaire $+$ et le symbole de fonction binaire $\times$. Dans ce qui suit, $F[v_0, v_1, v_2, ..., v_n]$ désigne une formule du langage $\mathcal{L}$ ne comportant aucune variable libre autre que $v_0, v_1, ..., v_n$, c'est-à-dire une expression construite de la façon habituelle à l'aide des connecteurs propositionnels, des quantificateurs "pour tout", "il existe", des symboles $0, S, +, \times$ et des variables $v_0, v_1, ..., v_n, ...$

Définition 56. *Les axiomes de Peano sont les sept axiomes A_1 à A_7 ci-dessous, ainsi qu'une infinité d'axiomes que l'on appellera le schéma d'induction et que l'on notera SI.*

A1. Pour tout v_0, $Sv_0 \neq 0$.

A2. Pour tout v_0 *il existe* v_1 *tel que* $v_0 \neq 0$ *implique* $Sv_1 = v_0$.

A3. Pour tout v_0, v_1, $Sv_0 = Sv_1$ *implique* $v_0 = v_1$.

A4. Pour tout $v_0, v_0 + 0 = v_0$.

A5. Pour tout v_0, v_1, $v_0 + Sv_1 = S(v_0 + v_1)$.

A6. Pour tout $v_0, v_0 \times 0 = 0$.

A7. Pour tout v_0, v_1, $v_0 \times Sv_1 = (v_0 \times v_1) + v_0$.

SI. Pour chaque formule $F[v_0, v_1, ..., v_n]$ *de* $\mathcal{L}$ *on a l'axiome suivant :*

Pour tout $v_1, v_2, ..., v_n$, *si pour tout* v_0, $F[0, v_1, v_2, ..., v_n]$ *et* $(F[v_0, v_1, ..., v_n]$ *implique* $F[Sv_0, v_1, ..., v_n])$, *alors pour tout* v_0, $F[v_0, v_1, ...v_n]$.

L'axiome A1 nous dit que 0 n'est pas un successeur. L'axiome A2 indique que

tout élément différent de 0 est le successeur d'un autre élément. L'axiome A3 dit que la fonction successeur S est injective. Les axiomes A4 et A6 déterminent l'addition et la multiplication par l'élément 0. L'axiome A5 correspond à la définition de l'addition par récurrence. Pour l'arithmétique usuelle cela signifie que $v_0 + (v_1 + 1) = (v_0 + v_1) + 1$. Similairement, l'axiome A7 correspond à la définition de la multiplication par récurrence à l'aide de l'addition. Finalement, le schéma d'induction SI nous assure que l'on peut démontrer à partir des axiomes de Peano des formules qui nécessitent l'induction. Pour illustrer ceci, dans l'arithmétique usuelle l'induction permet de déduire qu'une formule $f(n)$ pour tout $n \in \mathbb{N}$ est vrai si la formule est vraie pour $n = 0$ et si "$f(k)$ implique $f(k+1)$" est vrai pour tout k. Un exemple de formule qui est démontrable par induction est $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

Voici un exemple, provenant de (Walters, 1991), d'un système de réécriture pour l'arithmétique se basant sur les axiomes de Peano.

$$\begin{array}{lrcl} \text{P1.} & 0+x & \to & x \\ \text{P2.} & s(x)+y & \to & s(x+y) \\ \text{P3.} & 0\cdot x & \to & 0 \\ \text{P4.} & s(x)\cdot y & \to & x\cdot y+y \end{array}$$

Notons que ce système de réécriture de termes pour l'arithmétique inclut l'addition et la multiplication d'entiers positifs et est seulement valide pour la base 1.

Exemple 57. Voici comment s'applique les règles au terme $s(s(0)) \cdot s(0)$ qui représente dans ce système l'expression de l'arithmétique usuelle $2 \cdot 1$:

$$\begin{array}{rcl} s(s(0))\cdot s(0) & \to_{P_4} & s(0)\cdot s(0)+s(0) \\ & \to_{P_4} & 0\cdot s(0)+s(0)+s(0) \\ & \to_{P_3} & s(0)+s(0) \\ & \to_{P_2} & s(0+s(0)) \\ & \to_{P_1} & s(s(0)) \end{array}$$

3.2.2 Cohen et Watson

Le système décrit dans (Cohen et Watson, 1991) pour l'addition et la multiplication de l'arithmétique est pour des entiers en base 4. Il est démontré que le système est faiblement confluent. La question de savoir si le système possède la propriété de terminaison est un problème toujours ouvert qui fait partie d'une liste de plusieurs problèmes qui a été présentée à la conférence "Rewriting Techniques and Applications 1993" (Dershowitz, Jouannaud et Klop, 1993).

Le système a été construit de manière à avoir une bonne efficacité en vitesse et en espace. Par un exemple, il est montré que le système peut être considéré comme un module dans un système plus grand. Afin de calculer 10! (dix factorielle) le système nécessite aux alentours de 330 étapes de réécriture, ce qui est beaucoup moins que le système basé sur les axiomes de Peano.

3.2.3 Walters

Dans (Walters, 1994) on nous présente un système de réécriture de termes pour la base 10 avec l'addition et la soustraction de nombres entiers. Le système est confluent et possède la propriété de terminaison.

Le but de l'article est de présenter un système où les entiers sont représentés dans un système de numération décimale qui possède une complexité plus basse que celle du système de réécriture basé sur les axiomes de Peano. Voici le système :

W0.	$0x$	$\rightarrow$	x
W1,1.	$x(yz)$	$\rightarrow$	$(x+y)z$
W1,2.	$x(-(yz))$	$\rightarrow$	$-((y-x)z)$

W2,1,1. $1(-1) \rightarrow 9$

$\vdots$

W2,9,9. $9(-9) \rightarrow 81$

W3,0,1. $x0(-1) \rightarrow x(-1)9$

$\vdots$

W3,9,9. $x9(-9) \rightarrow x81$

W4. $(-x)y \rightarrow -(x(-y))$

W5. $--x \rightarrow x$

W6. $-0 \rightarrow 0$

W7,1. $0+x \rightarrow x$

W7,2. $x+0 \rightarrow x$

W8,1,1. $1+1 \rightarrow 2$

$\vdots$

W8,9,9. $9+9 \rightarrow 18$

W9,1. $x+yz \rightarrow y(x+z)$

W9,2. $xy+z \rightarrow x(y+z)$

W10,1. $x+-y \rightarrow x-y$

W10,2. $-x+y \rightarrow y-x$

W11,1. $0-x \rightarrow -x$

W11,2. $x-0 \rightarrow x$

W12,1,1. $1-1 \rightarrow 0$

$\vdots$

W12,9,9. $9-9 \rightarrow 0$

W13,1. $xy-z \rightarrow x(y-z)$

W13.2. $x-yz \rightarrow -(y(z-x))$

W14,1. $x--y \rightarrow x+y$

W14,2. $-x-y \rightarrow -(x+y)$

Exemple 58. Voici un exemple d'une suite de réductions du terme $21 - 6$:

$$\begin{aligned} 21 - 6 &\rightarrow_{W13,1} 2(1 - 7) \\ &\rightarrow_{W12,1,7} 2(-6) \\ &\rightarrow_{W2,2,6} 14 \end{aligned}$$

3.2.4 Kennaway

Dans (Kennaway, 1995) on présente un système orthogonal pour la base dix contenant l'addition, la soustraction et la multiplication. Les termes qui sont des formes normales ne contenant pas de variables (termes clos) ne sont pas en bijection avec les nombres entiers, c'est-à-dire que des réductions peuvent aboutir à des termes qui ne correspondent pas à des nombres entiers.

Des théorèmes sont présentés qui permettent de vérifier la propriété de terminaison d'un système de réécriture. Ces théorèmes permettent de vérifier que le système de Kennaway possède la propriété de terminaison. Dans la deuxième partie de l'article, il est démontré qu'en utilisant les mêmes méthodes il est possible de construire pour toute fonction totale récursive sur les entiers positifs un système de réécriture de termes représentant cette fonction et possédant la propriété de terminaison. Dans ce qui suit, nous donnons un des théorèmes principaux de l'article utilisé pour montrer la terminaison, et qui concerne les arborescences. Il est connu qu'un terme peut être écrit comme une arborescence étiquetée et vice versa. Pour cette raison le théorème suivant s'applique à des système de réécritures de termes. Pour plus de détails à propos de la correspondance entre termes et arborescences se référer à la page 28 de (TeReSe, 2003).

Voici ce qui est exposé dans l'article de Kennaway.

Définition 59. *Une forêt paramétrée est une forêt où chaque sommet est étiqueté par un symbole d'un alphabet ou par une variable, où seulement les feuilles peuvent être étiquetées par une variable et où chaque arbre de la forêt est une arborescence.*

Définition 60. *Une forêt sur $s_1, s_2..., s_n$ est une forêt paramétrée où chaque variable a été remplacée par une arborescence $s_1, s_2, ...$ ou s_n tel que chaque occurrence d'une*

même variable est toujours remplacée par la même arborescence.

Théorème 61. *Supposons qu'il y a une relation d'ordre bien-fondée sur un ensemble P. Prenons l'ensemble de forêts d'arborescences finies où les sommets sont étiquetés par des éléments de P. Pour deux forêts t_1 et t_2 on définit $t_1 > t_2$ si t_2 peut être obtenu de t_1 par l'opération suivante : prenons un sommet de t_1 étiqueté par un symbole p et désignons par $s_1, s_2, ..., s_n$ les sous-arborescences de t_1 ayant comme sommets seulement des descendants de p ; remplaçons alors p par n'importe quelle forêt sur $s_1, s_2, ..., s_n$ où les nouveaux sommets sont tous plus petits que p de manière à ce que les racines de la forêt soient des descendants du parent de p. Alors la fermeture réflexive transitive de $>$ est bien-fondée.*

3.2.5 Walters et Zantema

Dans (Walters et Zantema, 1995) trois systèmes sont présentés. Le premier est en base 1 et couvre l'addition, la soustraction et la multiplication. Il est confluent et fortement normalisant. Il utilise une fonction prédécesseur qui est inspirée de la fonction successeur des axiomes de Peano. Le deuxième système couvre l'addition et la multiplication pour une base arbitraire et est confluent et fortement normalisant. Le troisième couvre l'addition et la multiplication pour une base arbitraire, n'est pas confluent et il est l'extension pour une base arbitraire de ce qui a été présenté dans (Walters, 1994) vu ci-haut. Notons que le deuxième et troisième système possède la propriété d'être "facilement lisible par une personne", c'est-à-dire que les systèmes se rapprochent de l'arithmétique usuelle.

Les propriétés que les auteurs cherchent à avoir pour un système de réécriture pour l'arithmétique sont la confluence, la propriété de terminaison, une complexité logarithmique de temps et d'espace, une correspondance biunivoque entre les termes qui sont des formes normales et les nombres entiers, une lecture facile du système par une personne, un nombre minimal de règles et s'appliquant à une base arbitraire. La complexité de temps signifie le nombre d'étapes requises pour réduire l'addition, la

soustraction et la multiplication de deux nombres à une forme normale. La complexité d'espace signifie l'espace requis pour emmagasiner un nombre. Étant donné que ces propriétés n'ont pu être regroupées dans un même système, les auteurs présentent les trois systèmes mentionnés ci-dessus. De manière informelle, une complexité logarithmique signifie qu'en fonction de la grandeur de l'expression le nombre d'étapes requises pour réduire l'expression ou bien l'espace requis pour emmagasiner l'expression suit la fonction logarithmique, et ce, lorsque les expressions sont très grandes.

Les auteurs introduisent la notion de *schémata de règles* qui est un ensemble de règles qui doivent être générées pour chacune des bases. Nous avons un exemple de schémata de règles dans la description du système de Walter où les règles W2,1,1 à W2,9,9, les règles W3,1,1 à W3,9,9 et les règles W8,1,1 à W8,9,9 représentent des groupes de règles. Les schématas de règles sont similaires à des tables de multiplication. Ils permettent d'indiquer qu'il y a des règles du type $5 \odot 6 \rightarrow 30$ et $9 \odot 9 \rightarrow 81$. Il n'est pas nécessaire d'inclure des règles telles que $95 \odot 96$ dans les schématas de règles puisque dans ce système on utilise les retenues.

Dans l'article, ces schématas de règles doivent être construits pour ensuite être introduits dans les systèmes. Trois méthodes pour le faire sont brièvement discutées. Premièrement, la construction des schématas de règles pourrait être faite à la main. Cette méthode est utile pour une base n où n est petit. Pour la base dix, le schémata de règles nécessite déjà 414 règles. Deuxièmement, selon les auteurs, il serait possible d'étendre le troisième système en introduisant des fonctions successeur et prédécesseur dans le système. Cela réduirait l'efficacité du système et la facilité de lecture. La troisième méthode serait d'utiliser des fonctions externes telles que celles déjà incorporées dans l'architecture des ordinateurs. Nous verrons que les systèmes présentés au chapitre 4 constituent une autre méthode qui permet d'automatiser la construction des schématas de règles.

3.2.6 de Vries et Yamada

Dans (de Vries et Yamada, 1994) un système pour l'addition, la multiplication des nombres rationnels est présentée en base 3. Afin de généraliser à une base arbitraire il faut aussi avoir recourt à une table de règles. Il est démontré que les termes n'ayant pas de variable sont confluents. Il n'est pas démontré que le système est confluent. La propriété de terminaison du système est prouvée.

Le but de cet article était d'étendre les systèmes de réécriture de l'arithmétique à un système où les nombres rationnels sont représentés avec un développement décimal. Le système a été construit afin d'avoir une lecture facile des termes. Par exemple, $-321,6789$ s'écrit dans ce système comme $-((3:2):1);(6;(7;(8:9)))$. Dans l'appendice de l'article, les auteurs présentent dans leur notation le système de Cohen et Watson qui a été discuté à la section 3.2.2.

3.2.7 Contejean, Marché et Rabehasaina

Le système présenté dans (Contejean, Marché et Rabehasaina, 1997) représente l'arithmétique pour l'addition et la multiplication en base 3 pour les nombres rationnels. Le système est faiblement confluent et qu'il soit fortement normalisant est une conjecture.

L'objectif de cet article est différent des autres. L'algorithme de complétion de Knuth-Bendix permet de transformer un ensemble d'équations en un système de réécriture confluent (TeReSe, 2003). Il a été remarqué que l'algorithme de Buchberger possède de grandes ressemblances avec l'algorithme de complétion de Knuth-Bendix (Buchberger et Loos, 1982). Les auteurs sont intéressés à calculer la base de Gröbner d'idéaux de polynômes sur le corps des nombres rationnels. L'algorithme de complétion S-normalisé présenté dans (Marché, 1996) est une généralisation des deux algorithmes mentionnés ci-dessus. Afin de retrouver l'algorithme de Buchberger à partir de l'algorithme de S-complétion normalisé il est nécessaire d'avoir un système de réécriture S faiblement

confluent pour les anneaux commutatifs. C'est pourquoi l'article présente un système de réécriture efficace et faiblement confluent pour les nombres rationnels (écrit sous la forme de fractions). Un exemple de calcul de bases de Gröbner est présenté à la fin de l'article.

CHAPITRE IV

DEUX SYSTÈMES DE RÉÉCRITURE POUR L'ARITHMÉTIQUE

Dans ce chapitre nous présentons deux systèmes de réécriture pour l'arithmétique. Le premier système de réécriture, $\mathcal{S}_{ASMD}$, permet l'addition, la soustraction, la multiplication et la division de nombres rationnels pour une base arbitraire. En enlevant les règles pour la division au premier système et en ajustant quelques règles, nous avons un deuxième système, $\mathcal{S}_{ASM}$, qui est orthogonal.

Les deux systèmes ont été construits afin d'avoir, en passant d'une base b à une base $b+1$, une faible augmentation du nombre de règles. En contrepartie, ces systèmes nécessiteraient à un programme informatique un grand espace de travail, car ils reviennent à tout calculer avec des unités. Ils réduisent des expressions composées avec des additions, soustractions, multiplications, divisions et des nombres rationnels et produisent des formes normales qui sont des nombres rationnels du type $a_n a_{n-1}...a_1 a_0, c_1 c_2...c_m$. Un des avantages de ces systèmes est que les termes initiaux et les termes sous forme normale sont facilement lisibles, puisqu'ils suivent de près la notation usuelle de l'arithmétique.

4.1 Termes et quelques ensembles utiles

Avant de présenter les systèmes de réécriture suivant décrivant l'arithmétique nous allons définir l'ensemble des termes. Notons que l'ensemble des termes sera plutôt accessoire. L'ensemble de termes T sera l'ensemble ambiant dans lequel les ensembles des

terme T_{ASMD} et T_{ASM}, définis respectivement à la section 4.2.3 et 4.3.3, apparaissent comme des sous-ensembles.

Donnons quelques définitions d'ensembles, incluant les ensembles *CompteurDiv* et *PrecisionInit* qui nous serviront à la section 4.2.3.

Définition 62.

1. *L'*ensemble des variables V *est* $\{x, y, z, t, u, v, w, a, b, c\}$.
2. *L'*ensemble compteur de division *CompteurDiv est* $\{\star\}^*$.
3. *L'*ensemble de précision initial *PrecisionInit est l'ensemble de tous les mots de la forme* $\blacksquare\delta\boxtimes$ *où* $\delta \in \{\varepsilon, \square\}^*$.
4. *L'*ensemble des symboles B pour une base b *est* $\{1, s_2, ..., s_{b-1}, 0\}$.
5. *L'*ensemble R des symboles apparaissant dans les règles *est* $\{+, -, \cdot, \div, [,], \{, \}, (,), \blacktriangle, \oplus, \ominus, \bullet, \infty, 0/0, ;, \langle\} \cup \{, \}$.

Maintenant donnons l'ensemble des termes ambiants T dans lequel les termes des systèmes de réécriture de termes $\mathcal{S}_{ASMD}$ et $\mathcal{S}_{ASM}$ sont inclus.

Définition 63. *Soit* $\Sigma_T = \{\varepsilon\} \cup V \cup B \cup R \cup \{\div^\alpha\} \cup \{\div_\beta\}$
où $\alpha \in CompteurDiv$ *et* $\beta \in \{\square, \blacksquare, \boxtimes\}^*$. *L'*ensemble des termes ambiants T *est* Σ_T^*.

4.2 Système de réécriture de termes incluant la division pour les nombres rationnels

Le système de réécriture $\mathcal{S}_{ASMD}$ permet l'addition, la soustraction, la multiplication et la division de nombres rationnels pour une base arbitraire.

4.2.1 Principe

Lorsque nous avons une expression avec des additions, des soustractions, des multiplications et des divisions de nombres entiers représentés dans une base b alors le système que nous présentons consiste en six étapes principales.

1. Les nombres représentés en base d sont transformés en sommes d'unités.
2. Si ces nombres sont négatifs, alors le signe négatif se propage à toutes les unités de la somme.
3. Nous appliquons la multiplication en distribuant le facteur aux unités de la somme et nous appliquons les règles combinant la division avec les autres opérateurs afin de nous retrouver avec une expression ayant une somme d'unités au numérateur et une somme d'unités au dénominateur.
4. À ce point nous avons une somme d'unités positives ou négatives au numérateur et au dénominateur. Avec l'aide des symboles $\oplus$ et $\ominus$ nous opérons les annulations des unités positives avec les négatives.
5. Nous effectuons la division finale des unités, ce qui donne un terme du type $+1 + ... + 1, +1 + ... + 1; +1 + ... + 1; ....$
6. Nous construisons un nombre en effectuant la collection des unités et en écrivant le nombre dans la base requise.

4.2.2 Termes initiaux

Dans les expressions arithmétiques usuelles qui contiennent des additions, soustractions, multiplications et divisions, il faut prendre soin d'indiquer l'ordre des opérations par des parenthèses pour éviter les ambiguïtés. Pour le système de réécriture de l'arithmétique que nous allons donner, nous n'avons pas à nous soucier de l'ordre des opérations puisqu'il est inclus implicitement dans les règles de réécriture. Afin d'avoir des termes qui ont du sens et qui représentent adéquatement les expressions arithmétiques usuelles nous nous dotons d'un ensemble de termes initiaux. Cet ensemble de termes initiaux est un sous-ensemble de l'ensembles des termes T. Les différences majeures entre les expressions arithmétiques usuelles et les termes initiaux est que dans les termes initiaux nous n'utilisons pas de parenthèses pour encadrer l'addition, tous les termes initiaux possèdent un symbole de division (nous divisons par $+1$ pour représenter les entiers) et tous les termes initiaux sont encadrés par deux triangles noirs qui servent lors de l'application des règles de réécriture.

Nous identifierons toujours l'unité de la base par 1 et le zéro par 0. Notons que notre système est construit afin de modéliser la notation usuelle de l'arithmétique. Il serait éventuellement intéressant de voir s'il est possible de modéliser l'arithmétique écrite sous la forme de notation préfixée sans parenthèses (notation polonaise), où l'on écrit $(x \cdot y)$ comme $\cdot xy$.

Les *termes initiaux de* $\mathcal{S}_{ASMD}$ sont donnés grâce aux deux définitions suivantes.

Définition 64. *Soit* $B = \{1, s_2, ..., s_{b-1}, 0\}$ *un ensemble de symboles pour une base* b*, on définit les ensembles de termes suivants :*

0. T_0 *:*

0.1. $1, s_2, ...s_{b-1}, 0 \in T_0$

0.2. $c \in T_0 \setminus \{0\}$ *et* $d \in T_0$ *alors* $cd \in T_0$

1. T_1 *:*

1.1. si $f \in T_0 \setminus \{0\}$ *alors* $[f] \in T_1$

2. T_2 *:*

2.1. si $r \in T_1$ *alors* $r \in T_2$

2.2. si $s \in T_2$ *et* $t \in T_2$ *alors* $s + t \in T_2$

2.3. si $m \in T_2$ *et* $n \in T_2$ *alors* $(m \cdot n) \in T_2$

2.4. si $l \in T_2$ *alors* $-\{l\} \in T_2$

2.5. si $p \in T_2$ *et* $q \in T_2$ *alors* $(p \div^* q) \in T_2$

3. T_3 *:*

3.1. si $u \in T_2$ *et* u *ne contient pas le symbole* $\div^*$ *alors* $(u \div^* +1) \in T_3$

3.2. si $v \in T_2$ *et* v *contient le symbole* $\div^*$ *alors* $v \in T_3$

4. T_I *:*

4.1. si $w \in T_3$ *alors* $\blacktriangle w \blacktriangle \in T_I$

Définition 65. *Un élément de* T_I *est appelé un terme initial de* $\mathcal{S}_{ASMD}$*.*

En appliquant successivement les règles du système de réécriture à un terme initial, nous obtiendrons un résultat qui sera une forme normale qui prendra la forme d'un nombre représenté dans une base choisie.

Des exemples de termes initiaux sont :

▲([1011100101] $\div^*$ +1)▲ pour une base binaire donnée par $B = \{1, 0\}$.

▲([34240010] $\div^*$ +1)▲ pour une base décimale donné par $B = \{1, 2, ..., 9, 0\}$.

▲([3AD44F] $\div^*$ +1)▲ pour la base hexadécimale $B = \{1, 2, ..., 9, A, B, ..., F, 0\}$.

▲(([54] · [345]) · [4000])[400]▲ pour la base décimale.

▲(([54] · [345]) · ([4000] $\div^*$ [4]))[400]▲ pour la base décimale.

Nous voyons qu'il y a dans T_I des termes du type

$$▲[100] + ([3] \div^* [10]) + ([2] \div^* [100]) + ([1] \div^* [1000])▲$$

Cela nous indique que, d'une certaine façon, les nombres rationnels font partie de l'ensemble initial de termes, et ce, indépendamment de la base choisie.

4.2.3 Le Système $\mathcal{S}_{ASMD}$

Le système de réécriture $\mathcal{S}_{ASMD}$ est donné par les règles suivantes identifiées par R et un nombre. L'ensemble des termes T_{ASMD} sera spécifié par l'ensemble des règles et par l'ensemble initial T_I. Notons que cet ensemble est un sous-ensemble de l'ensemble des termes T.

Définition 66. *L'ensemble des* termes du système T_{ASMD} *est composé de l'ensemble des termes initiaux T_I et de tous les termes découlant de T_I par l'application successive des règles où les variables qui ne sont pas en indice et qui apparaissent dans les règles de réécriture prennent comme valeurs que des termes de T_0, T_2 ou des termes découlant de T_2 par l'application successive des règles.*

Notons que dans un terme de T_2 nous avons que si la parenthèse "(" apparaît alors il y a une parenthèse ")" qui lui correspond et qui apparaît à un endroit à sa droite. Nous discuterons de ceci immédiatement après avoir présenté les règles R20 à R28.

Le premier sous-ensemble de règles est composé des règles qui permettent de transformer les nombres représentés dans une certaine base en une somme d'unités. Ce

sous-ensemble de règles de déconstruction dépend de la base choisie. Ici nous présentons les règles pour la base décimale. Dans ce qui suit nous avons que la variable x prendra toujours comme valeur un élément de T_0.

$$
\begin{array}{lrcl}
\text{R1.} & 1] & \rightarrow & 0]+1 \\
\text{R2.} & 2] & \rightarrow & 0]+1+1 \\
 & & \vdots & \\
\text{R8.} & 8] & \rightarrow & 0]+1+1+1+1+1+1+1+1 \\
\text{R9.} & 9] & \rightarrow & 0]+1+1+1+1+1+1+1+1+1 \\
\text{R10.} & [x0] & \rightarrow & [x]+[x]+[x]+[x]+[x]+[x]+[x]+[x]+[x]+[x] \\
\text{R11.} & [0] & \rightarrow & \varepsilon \\
\text{R12.} & ++ & \rightarrow & + \\
\text{R13.} & +- & \rightarrow & -
\end{array}
$$

Notons que le mot vide ne peut être valeur d'une variable puisque le mot vide n'est pas dans T_2 ou T_0 . Par exemple, puisque x ne peut pas prendre le mot vide comme valeur la règle $[0] \rightarrow [] + [] + ...[]$ n'est pas valide. Voici un exemple de déconstruction :

Exemple 67.

$$
\begin{array}{lll}
\blacktriangle[14]\blacktriangle & \rightarrow_{R_4} & \blacktriangle[10]+1+1+1+1\blacktriangle \\
 & \rightarrow_{R_{10}} & \blacktriangle[1]+[1]+[1]+[1]+[1]+[1]+[1]+[1]+[1]+[1]+1+1+1+1\blacktriangle \\
 & \rightarrow_{R_1} & \blacktriangle[1]+[0]+1+[1]+[1]+[1]+[1]+[1]+[1]+[1]+[1]+1+1+1+1\blacktriangle \\
 & \rightarrow_{R_1} & \blacktriangle[1]+[0]+1+[1]+[0]+1+[1]+[1]+[1]+[1]+[1]+[1]+1+1+1+1\blacktriangle \\
 & \rightarrow_{R_{12}} & \blacktriangle[1]+[0]+1+[1]++1+[1]+[1]+[1]+[1]+[1]+[1]+1+1+1+1\blacktriangle \\
 & \rightarrow_{R_{13}} & \blacktriangle[1]+[0]+1+[1]+1+[1]+[1]+[1]+[1]+[1]+[1]+1+1+1+1\blacktriangle \\
 & \rightarrow & ... \\
 & \rightarrow & \blacktriangle+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1\blacktriangle
\end{array}
$$

Un nombre négatif est un nombre qui est une série d'unités négatives. Voici les règles qui permettent de changer le signe. Notons que le changement de signe s'exécute seulement aux unités.

R14. $-\{+1 \rightarrow -1-\{$

R15. $-\{-1 \rightarrow +1-\{$

R16. $-\{\} \rightarrow \varepsilon$

Étant donné que nous considérons dans notre système les nombres négatifs, nous devons avoir deux règles pour la multiplication, une s'appliquant à l'unité positive et l'autre s'appliquant à l'unité négative.

R17. $(x \cdot +1 \rightarrow x + (x\cdot$

R18. $(x \cdot -1 \rightarrow -\{x\} + (x\cdot$

R19. $(x\cdot) \rightarrow \varepsilon$

Un exemple avec la multiplication de nombres négatifs est donné dans ce qui suit. Il est important de noter que lors de la construction des termes initiaux, les termes du type $s+t$ ne sont pas encadrées par des parenthèses, cela nous permet d'avoir une notation plus simple. Puisque nous n'utilisons pas de parenthèses pour encadrer l'addition, les termes qui sont multipliés ensemble sont ce qui est inclus entre la parenthèse gauche "(" et le symbole de multiplication "·" et ce qui est inclus entre le symbole de multiplication "·" et la parenthèse droite ")". Le terme $(([3] \cdot [5]) + [7] \cdot [2])$ sera donc réduit au terme 44 et non au terme 29. Pour cette raison, nous avons que l'exemple suivant est interprété dans l'arithmétique usuelle comme 3 multiplié par -1 ce qui donne -3 et non comme 3 multiplié par -2 et ensuite augmenté de 1 ce qui donne -5.

Exemple 68.

$$
\begin{aligned}
([3] \cdot -\{[2]\} + 1) &\rightarrow_{R_2} ([3] \cdot -\{[0] + 1 + 1\} + 1) \\
&\rightarrow_{R_{12}} ([3] \cdot -\{+1 + 1\} + 1) \\
&\rightarrow_{R_{15}} ([3] \cdot -1 - \{+1\} + 1) \\
&\rightarrow_{R_{15}} ([3] \cdot -1 - 1\{\} + 1) \\
&\rightarrow_{R_{18}} ([3] \cdot -1 - 1 + 1) \\
&\rightarrow_{R_{20}} (-\{[3]\} + ([3] \cdot -1 + 1) \\
&\rightarrow_{R_3} (-\{[3]\} + ([0] + 1 + 1 + 1 \cdot -1 + 1) \\
&\rightarrow_{R_{20}} (-\{[3]\} + -\{[0] + 1 + 1 + 1\} + ([0] + 1 + 1 + 1 \cdot +1) \\
&\rightarrow_{R_{12}} (-\{[3]\} + -\{[0] + 1 + 1 + 1\} + (+1 + 1 + 1 \cdot +1)
\end{aligned}
$$

$$
\begin{aligned}
&\rightarrow_{R_{19}} \quad (-\{[3]\} + -\{[0]+1+1+1\} + 1 + 1 + 1 + (+1+1+1\cdot)\\
&\rightarrow \quad \ldots\\
&\rightarrow \quad -1-1-1-1-1-1+1+1+1
\end{aligned}
$$

Maintenant nous donnons les règles impliquant la division permettant de transformer un terme initial en un terme de la forme $\blacktriangle(x \div^m y)\blacktriangle$. Nous devons manuellement compter le nombre de symboles de division apparaissant dans le terme initial. Un système de réécriture parallèle à notre système pour l'arithmétique pourrait obtenir ce nombre. Nous ne construirons pas ce système. Le nombre de symboles de division est représenté par une suite de symboles $\star$. Dans la règle de transition $\blacktriangle(x \div^m y)\blacktriangle \rightarrow (\oplus x \ominus) \div (\oplus y \ominus)$ nous remplaçons m par le nombre adéquat de symboles $\star$. Par exemple si nous avons un terme initial $((t \div^* u)\div^*) + (v \div^* w)$ nous aurons que $m = \star\star\star$. Ainsi, la règle de transition $\blacktriangle(x \div^m y)\blacktriangle$ s'appliquera seulement s'il n'y a plus de symboles de division apparaissant dans x et y du côté gauche de la règle. Cette construction s'apparente à la notion informatique des boucles conditionnelles.

R20.	$(x \div^a y) \div^b z$	$\rightarrow$	$x \div^{ab} (y \cdot z)$
R21.	$z \div^b (x \div^a y)$	$\rightarrow$	$(z \cdot y) \div^{ab} x$
R22.	$x \cdot (y \div^a z)$	$\rightarrow$	$(x \cdot z) \div^a y$
R23.	$(y \div^a z) \cdot x$	$\rightarrow$	$(y \cdot x) \div^a z$
R24.	$(x \div^a y) + z$	$\rightarrow$	$(x + (y \cdot z) \div^a y)$
R25.	$z + (x \div^a y)$	$\rightarrow$	$((z \cdot y) + x \div^a y)$
R26.	$(x \div^a y) \div^b (u \div^c v)$	$\rightarrow$	$(x \cdot v) \div^{abc} (y \cdot u)$
R27.	$\blacktriangle(x \div^m y)\blacktriangle$	$\rightarrow$	$(\oplus x \ominus) \div (\oplus y \ominus)$
R28.	$-\{(x \div^a y)\}$	$\rightarrow$	$(-\{x\} \div^a y)$

Il est important de noter que lorsque nous appliquons les règles R20 à R28 les parenthèses sont bien respectées. Une valeur adéquate de x serait par exemple $([3] \cdot [5]) + [7]$. Puisque l'ensemble des termes du système est composé de l'ensemble initial et de tous les termes découlant de l'application successive des règles où les variables apparaissant dans les règles de réécriture prennent seulement comme valeurs des termes

de T_2 ou des termes découlant de T_2 par l'application successive des règles, on a qu'une valeur de x ne pourrait pas être $4[3]\cdot[5])+[7]$. Cela empêche de mauvaises interprétations des termes.

Puisque dans le terme initial de l'exemple qui suit il y a quatre symboles de division, nous aurons que $\div^m = \div^{****}$.

Exemple 69.

$$
\begin{aligned}
&\blacktriangle(t \div^* ((z \cdot (x \div^* y)) \div^* (v \div^* w))) + u\blacktriangle \\
&\qquad \rightarrow_{R_{25}} \quad \blacktriangle(t \div^* (((z \cdot x) \div^* y) \div^* (v \div^* w))) + u\blacktriangle \\
&\qquad \rightarrow_{R_{29}} \quad \blacktriangle(t \div^* (((z \cdot x) \cdot w) \div^{***} (y \cdot v))) + u\blacktriangle \\
&\qquad \rightarrow_{R_{24}} \quad \blacktriangle((t \cdot (y \cdot v)) \div^{****} ((z \cdot x) \cdot w)) + u\blacktriangle \\
&\qquad \rightarrow_{R_{27}} \quad \blacktriangle((t \cdot (y \cdot v)) + (((z \cdot x) \cdot w) \cdot u) \div^{****} ((z \cdot x) \cdot w))\blacktriangle \\
&\qquad \rightarrow_{R_{30}} \quad (\oplus(t \cdot (y \cdot v)) + (((z \cdot x) \cdot w) \cdot u)\ominus) \div (\oplus((z \cdot x) \cdot w)\ominus)
\end{aligned}
$$

Pour la soustraction, il n'est pas assez d'avoir des règles d'annulation des unités avec leur négatif. Il faut aussi transformer le terme de manière à avoir au numérateur et au dénominateur des sommes d'unités positives et le signe négatif s'il y a lieu à l'extrême gauche du terme. Nous introduisons donc les symboles $\oplus$ et $\ominus$.

R29.	$\oplus + 1$	$\rightarrow$	$+1\oplus$
R30.	$-1\ominus$	$\rightarrow$	$\ominus - 1$
R31.	$+1 - 1$	$\rightarrow$	
R32.	$-1 + 1$	$\rightarrow$	
R33.	$+1 \oplus \ominus - 1$	$\rightarrow$	$\oplus\ominus$

Nous avons aussi besoin des règles suivantes. Pour l'instant nous dirons seulement que $\rho \in \mathit{PrecisionInit}$ où l'ensemble $\mathit{PrecisionInit}$ est défini dans la définition 62. Nous allons donner plus d'explications dans ce qui suit.

R34.	$(\oplus \ominus x) \div (\oplus \ominus y)$	$\rightarrow$	$\bullet 0(-\{x\}) \div_\rho (-\{y\})\bullet$
R35.	$(x \oplus \ominus) \div (\oplus \ominus y)$	$\rightarrow$	$-\bullet\, 0(x) \div_\rho (-\{y\})\bullet$
R36.	$(x \oplus \ominus) \div (y \oplus \ominus)$	$\rightarrow$	$\bullet 0(x) \div_\rho (y)\bullet$
R37.	$(\oplus \ominus x) \div (y \oplus \ominus)$	$\rightarrow$	$-\bullet\, 0(-\{x\}) \div_\rho (y)\bullet$

Dans l'exemple suivant, nous voyons comment s'annulent les unités positives avec les négatives et nous voyons une transition s'opérer à la première et dernière ligne. Dans le cas de ce terme initial il y a une division, alors nous avons que $m = \star$.

Exemple 70.

$$
\begin{aligned}
&\blacktriangle(+1-1+1-1-1-1+1 \div^{\star} +1)\blacktriangle \\
&\qquad \rightarrow \quad (\oplus+1-1+1-1-1-1+1\ominus) \div (\oplus+1\ominus) \\
&\qquad \rightarrow \quad (+1\oplus-1+1-1-1-1+1\ominus) \div (\oplus+1\ominus) \\
&\qquad \rightarrow \quad (+1\oplus-1+1-1-1\ominus) \div (\oplus+1\ominus) \\
&\qquad \rightarrow \quad (+1\oplus-1+1-1\ominus-1) \div (\oplus+1\ominus) \\
&\qquad \rightarrow \quad (+1\oplus-1+1\ominus-1-1) \div (\oplus+1\ominus) \\
&\qquad \rightarrow \quad (+1\oplus\ominus-1-1) \div (\oplus+1\ominus) \\
&\qquad \rightarrow \quad (\oplus\ominus-1) \div (\oplus+1\ominus) \\
&\qquad \rightarrow \quad (\oplus\ominus-1) \div (+1\oplus\ominus) \\
&\qquad \rightarrow \quad \bullet(+1) \div_{\rho} (-\{-1\})\bullet
\end{aligned}
$$

L'action des symboles $\oplus$ et $\ominus$ permet aussi d'avoir des résultats finaux tels que 0, ∞, $-\infty$ et $0/0$.

$$
\begin{aligned}
&\text{R38.} \quad (\oplus\ominus) \div (\oplus\ominus y) \quad \rightarrow \quad 0 \\
&\text{R39.} \quad (\oplus\ominus) \div (y\oplus\ominus) \quad \rightarrow \quad 0 \\
&\text{R40.} \quad (x\oplus\ominus) \div (\oplus\ominus) \quad \rightarrow \quad \infty \\
&\text{R41.} \quad (\oplus\ominus x) \div (\oplus\ominus) \quad \rightarrow \quad -\infty \\
&\text{R42.} \quad (\oplus\ominus) \div (\oplus\ominus) \quad \rightarrow \quad 0/0
\end{aligned}
$$

Le sous-ensemble suivant de règles pour la division permet d'obtenir le résultat sous forme de notation positionnelle. Avant d'appliquer les règles où le symbole ρ apparaît, il faut déterminer la précision, c'est-à-dire identifier ρ à un unique élément de l'ensemble de précision initial *PrecisionInit*. Le nombre de carrés blancs apparaissant dans ρ en indice au signe de division représente le nombre de décimales du résultat. Par exemple, $\div_{\rho} = \div_{\blacksquare\square\square\square\square\square\boxtimes}$ donnera un résultat du type $d_n d_{n-1} \ldots d_1 d_0, d_{-1} d_{-2} d_{-3} d_{-4} d_{-5}$ pour $d_i \in B$.

Dans le cas où la division arrive juste, c'est-à-dire que la période est zéro, les zéros jusqu'à la précision voulue n'apparaîtront pas. Il serait possible d'ajuster les règles afin d'avoir les zéros jusqu'à la décimale voulue. Comparativement au m qui apparaît dans les règles pour la division qui est dépendante du nombre de symboles de division apparaissant dans le terme initial, la précision ρ est choisie arbitrairement.

Le principe sous-jacent aux règles pour la division énoncées ci-dessous est que le système "calcule" combien de fois le dénominateur apparaît dans le numérateur. Si le dénominateur n'apparaît aucune fois, alors nous changeons de position en insérant une virgule ou un point-virgule devant le terme et nous multiplions le numérateur par dix. Lorsque le dénominateur apparaît dans le numérateur alors nous insérons un $+1$ devant le terme.

Notons que les symboles u, w, z apparaissant en indice de $\div$ sont des variables. Dû à la forme des termes initiaux et des règles de réécriture du système nous aurons toujours des indices où $u \in \{\Box\}^*$, $w \in \{\delta \boxtimes | \delta \in \{\Box\}^*\}$ et $z \in \{\alpha \blacksquare \beta \boxtimes | \alpha, \beta \in \{\varepsilon, \Box\}\}$.

R43.	$+1) \div_z (+1$	$\rightarrow$	$) + 1 \div_z +1($
R44.	$+1)x \div_z x(+1$	$\rightarrow$	$) + 1x \div_z x + 1($
R45.	$\bullet v()x \div_z x()$	$\rightarrow$	$\bullet\langle v + 1$
R46.	$()x \div_{\blacksquare\Box w} x(y)$	$\rightarrow$	$,(([10] \cdot x)) \div_{\Box\blacksquare w} (xy)$
R47.	$()x \div_{u\blacksquare\Box w} x(y)$	$\rightarrow$	$;(([10] \cdot x)) \div_{u\Box\blacksquare w} (xy)$
R48.	$(y)x \div_z x()$	$\rightarrow$	$+1(y) \div_z (x)$
R49.	$\bullet v()x \div_{\blacksquare\boxtimes} x(y)\bullet$	$\rightarrow$	$\bullet\langle v\bullet$
R50.	$\bullet v()x \div_{u\blacksquare\boxtimes} x(y)\bullet$	$\rightarrow$	$\bullet\langle v\bullet$

Nous obtiendrons grâce aux règles vues ci-dessus des termes du type

$$+1+1+1+1+1+1+1,+1+1+1+1;+1+1;;+1+1+1$$

Notons que le nombre d'unités encadré par les symboles "," et ";" et les symboles ";" et ";" est inférieur ou égal à 9.

Les cinq règles suivantes sont les règles de collection qui rassemblent les unités.

Pour des bases b standards, ces règles ne dépendent pas de la base choisie. L'exemple 72 illustre le fonctionnement de la réduction des sommes d'unités en un nombre rationnel.

R51. $\langle, \rightarrow ,0\langle$

R52. $\langle; \rightarrow 0\langle$

R53. $\bullet x\langle\bullet \rightarrow x$

R54. $\bullet\langle+1 \rightarrow \bullet 1\langle$

R55. $\langle 0 \rightarrow 0\langle$

Les règles suivantes sont les règles qui donnent les assignations des constantes en fonction de la base choisie.

R56. $0\langle+1 \rightarrow 1\langle$

R57. $1\langle+1 \rightarrow 2\langle$

$\vdots$

R64. $8\langle+1 \rightarrow 9\langle$

R65. $9\langle+1 \rightarrow \langle+10$

Voici un exemple simple illustrant une division d'un entier par l'unité.

Exemple 71.

$\bullet(+1+1) \div_{\blacksquare\square\square\square\boxtimes} (+1)\bullet$

$\rightarrow_{R_{46}} \bullet(+1)+1 \div_{\blacksquare\square\square\square\boxtimes} +1()\bullet$

$\rightarrow_{R_{51}} \bullet+1(+1) \div_{\blacksquare\square\square\square\boxtimes} (+1)\bullet$

$\rightarrow_{R_{46}} \bullet+1()+1 \div_{\blacksquare\square\square\square\boxtimes} +1()\bullet$

$\rightarrow_{R_{48}} \bullet\langle+1+1\bullet$

Voici quelques exemples clés de collection des unités afin de construire le résultat en représentation décimale. Nous avons mis en évidence les transitions ajoutant une nouvelle position au nombre.

Exemple 72.

$\bullet 0\langle+1+1+...+1\bullet \rightarrow \bullet 1\langle+1+1+...+1\bullet$

$\rightarrow \bullet 2\langle+1+1+...+1\bullet$

$\rightarrow ...$

$\rightarrow \quad \bullet 9\langle +1+1+...+1\bullet$

$\rightarrow \quad \bullet\langle +10+1+..+1\bullet$

$\rightarrow \quad \bullet 1\langle 0+1+...+1\bullet$

$\rightarrow \quad \bullet 10\langle +1+...+1\bullet$

$\rightarrow \quad \bullet 11\langle +1+...+1\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet 19\langle +1+1...+1\bullet$

$\rightarrow \quad \bullet 1\langle +10+...+1\bullet$

$\rightarrow \quad \bullet 2\langle 0+...+1\bullet$

$\rightarrow \quad \bullet 20\langle +1+...+1\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet 99\langle +1+...+1\bullet$

$\rightarrow \quad \bullet 9\langle +10+...+1\bullet$

$\rightarrow \quad \bullet\langle +100+...+1\bullet$

$\rightarrow \quad \bullet 1\langle 00+...+1\bullet$

$\rightarrow \quad \bullet 10\langle 0+...+1\bullet$

$\rightarrow \quad \bullet 100\langle 1+...+1\bullet$

L'exemple suivant montre une réduction où la division arrive juste. Afin de minimiser l'espace utilisé par notre texte, nous utilisons une notation où 1_i signifie que c'est la i-ème unité. Cette notation est seulement utilisée pour aider la présentation sur papier des réductions, elle ne fait donc pas partie des termes du système. Voici donc l'exemple pour $3 \div 5$.

Exemple 73.

$\bullet(1+1+1) \div \blacksquare\square\square\boxtimes\ (+1+1+1+1+1)\bullet$

$\rightarrow \quad \bullet(+1+1)+1 \div \blacksquare\square\square\boxtimes +1(+1+1+1+1)\bullet$

$\rightarrow \quad \bullet(+1)+1+1 \div \blacksquare\square\square\boxtimes +1+1(+1+1+1)\bullet$

$\rightarrow \quad \bullet()+1+1+1 \div \blacksquare\square\square\boxtimes +1+1+1(+1+1)\bullet$

$\rightarrow \quad \bullet, (([10]\cdot +1+1+1)) \div \square\blacksquare\square\boxtimes\ (+1+1+1+1+1)\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet, (1_1 + 1_2 + 1_3 + ... + 1_{30}) \div \square\blacksquare\square\boxtimes\, (+1 + 1 + 1 + 1 + 1)\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet, (1_1 + ... + 1_{25}) + 1_{26} + ... + 1_{30} \div \square\blacksquare\square\boxtimes\, +1_1 + ... + 1_5()\bullet$

$\rightarrow \quad \bullet, +1(1_1 + ... + 1_{25}) \div \square\blacksquare\square\boxtimes\, (+1 + 1 + 1 + 1 + 1)\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet, +1 + 1 + 1 + 1 + 1() + 1_1 + ... + 1_5 \div \square\blacksquare\square\boxtimes\, +1_1 + ... + 1_5()\bullet$

$\rightarrow \quad \bullet 0\langle, +1 + 1 + 1 + 1 + 1 + 1\bullet$

$\rightarrow \quad \bullet 0, 0\langle +1 + 1 + 1 + 1 + 1 + 1\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet 0, 6\langle\bullet$

$\rightarrow \quad 0, 6$

L'exemple qui suit est la division de 20 par 7, ce qui est un cas où il y aura une période. Il devient donc important d'avoir un compteur $\blacksquare\square\square\boxtimes$ afin de limiter la précision.

Exemple 74.

$\bullet(1_1 + 1_2 + ... + 1_{20}) \div \blacksquare\square\square\boxtimes\, (+1_1 + ... + 1_7)\bullet$

$\rightarrow \quad \bullet(1_1 + 1_2 + ...1_{19}) + 1_{20} \div \blacksquare\square\square\boxtimes\, +1_1(+1_2 + ... + 1_7)\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet(1_1 + 1_2 + ...1_{13}) + 1_{14} + ... + 1_{20} \div \blacksquare\square\square\boxtimes\, +1_1 + 1_2 + ... + 1_7()\bullet$

$\rightarrow \quad \bullet + 1(1_1 + 1_2 + ...1_{13}) \div \blacksquare\square\square\boxtimes\, (+1_1 + 1_2 + ... + 1_7)\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet + 1 + 1()1_1 + 1_2 + ... + 1_6 \div \blacksquare\square\square\boxtimes\, +1_1 + 1_2 + ...(+1_7)\bullet$

$\rightarrow \quad \bullet + 1 + 1, (([10] \cdot 1_1 + 1_2 + ...1_6)) \div \square\blacksquare\square\boxtimes\, (+1_1 + 1_2 + ... + 1_7)\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet + 1 + 1, +1_1 + ... + 1_8(+1 + 1 + 1 + 1) \div \square\blacksquare\square\boxtimes\, (+1_1 + 1_2 + ... + 1_7)\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet + 1 + 1, +1_1 + ... + 1_8() + 1 + 1 + 1 + 1 \div \square\blacksquare\square\boxtimes + 1_1 + 1_2 + 1_3 + 1_4(+1_5 + ... + 1_7)\bullet$

$\rightarrow \quad \bullet + 1 + 1, +1_1 + ... + 1_8; (([10] \cdot 1 + 1 + 1 + 1)) \div \square\square\blacksquare\boxtimes\, (+1_1 + ... + 1_7)\bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet + 1 + 1, + 1_1 + ... + 1_8; + 1_1 + ... + 1_5 (+ 1_1 + ... + 1_5) \div \square\square\blacksquare\boxtimes \, (+ 1_1 + ... + 1_7) \bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet + 1 + 1, + 1_1 + ... + 1_8; + 1_1 + ... + 1_5() + 1_1 + ... + 1_5 \div \square\square\blacksquare\boxtimes$
$+ 1_1 + ... + 1_5 (+ 1_6 + 1_7) \bullet$

$\rightarrow \quad \bullet 0 \langle + 1 + 1, + 1_1 + ... + 1_8; + 1_1 + ... + 1_5 \bullet$

$\rightarrow \quad \bullet 1 \langle + 1, + 1_1 + ... + 1_8; + 1_1 + ... + 1_5 \bullet$

$\rightarrow \quad \bullet 2 \langle , + 1_1 + ... + 1_8; + 1_1 + ... + 1_5 \bullet$

$\rightarrow \quad \bullet 2, 0 \langle + 1_1 + ... + 1_8; + 1_1 + ... + 1_5 \bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet 2, 8 \langle ; + 1_1 + ... + 1_5 \bullet$

$\rightarrow \quad \bullet 2, 80 \langle + 1_1 + ... + 1_5 \bullet$

$\rightarrow \quad ...$

$\rightarrow \quad \bullet 2, 85 \langle \bullet$

$\rightarrow \quad 2, 85$

4.2.4 Confluence et Terminaison

Une différence entre ce système et les systèmes généralement utilisés dans la théorie des systèmes de réécriture est que dans ce cas-ci nous construisons un système pour représenter une structure déjà connue.

Si nous voulions montrer la confluence de ce système par la propriété d'orthogonalité, nous rencontrons un problème avec la division, puisqu'il apparaît des règles non linéaire-gauches soit les règles R40 à R47 où la variable x apparaît deux fois dans la partie gauche de chaque règle. Pour la preuve de confluence et terminaison, il faudrait donc utiliser d'autres techniques. Mentionons qu'il existe des logiciels, tels que *AProVE* (Giesl, Thiemann, Schneider-Kamp et Falke, 2004) et *CoLoR* (Blanqui, Delobel, Coupet-Grimal, Hinderer et Koprowski, 2006), qui permettent de montrer de tels résultats.

4.3 Système de réécriture de termes orthogonal sans la division pour les entiers relatifs

Basé sur le système $\mathcal{S}_{ASMD}$ nous donnons le système $\mathcal{S}_{ASM}$ incluant l'addition, la soustraction et la multiplication et pour lequel il est aisé de prouver la propriété de confluence puisque le système est orthogonal.

4.3.1 Principe

Lorsque nous avons une expression avec des additions, des soustractions et des multiplications de nombres entiers représentés dans une base d alors le système que nous présentons exécute cinq principales étapes.

1. Les nombres représentés en base d sont transformés en sommes d'unités.
2. Si ces nombres sont négatifs, alors le signe négatif se propage à toutes les unités de la somme.
3. Nous opérons la multiplication en distribuant les facteurs aux unités de la somme.
4. À ce point nous devrions avoir une somme d'unités positives et négatives et nous opérons donc les annulations.
5. Nous reconstruisons un nombre en effectuant la collection des unités.

4.3.2 Termes initiaux

Les *termes initiaux de* $\mathcal{S}_{ASM}$ sont donnés grâce aux deux définitions suivantes.

Définition 75. *Soit* $B = \{1, s_2, ..., s_{b-1}, 0\}$ *un ensemble de symboles pour une base* b*, on définit les ensembles de termes suivant :*

0. T_0 *:*

0.1. $1, s_2, ..., s_{b-1}, 0 \in T_0$

0.2. $c \in T_0 \setminus \{0\}$ *et* $d \in T_0$ *alors* $cd \in T_0$

1. T_1 *:*

1.1. si $f \in T_0 \setminus \{0\}$ *alors* $[f] \in T_1$

2. T_2 *:*

2.1. si $r \in T_1$ *alors* $r \in T_2$

2.2. si $s \in T_2$ *et* $t \in T_2$ *alors* $s + t \in T_2$

2.3. si $m \in T_2$ *et* $n \in T_2$ *alors* $(m \cdot n) \in T_2$

2.4. si $l \in T_2$ *alors* $-\{l\} \in T_2$

3. T_i *:*

3.1. si $w \in T_2$ *alors* $\bullet \oplus w \ominus \bullet \in T_i$

Définition 76. *Un élément de* T_i *est appelé un terme initial de* $\mathcal{S}_{ASM}$.

4.3.3 Le système confluent $\mathcal{S}_{ASM}$

Le système de réécriture $\mathcal{S}_{ASM}$ est donné par les règles suivantes identifiées par C et un nombre. Suite à l'énoncé des règles de réécriture, nous démontrons que $\mathcal{S}_{ASM}$ est confluent.

Définition 77. *L'ensemble des* termes du système T_{ASM} *est composé de l'ensemble des termes initiaux* T_i *et de tous les termes découlant de* T_i *par l'application successive des règles où les variables apparaissant dans les règles de réécriture prennent comme valeurs que des termes de* T_0, T_2 *ou des termes découlant de* T_2 *par l'application successive des règles.*

Voici les règles du système de réécriture $\mathcal{S}_{ASM}$.

C1. $1] \rightarrow 0] + 1$

C2. $2] \rightarrow 0] + 1 + 1$

$\vdots$

C8. $8] \rightarrow 0] + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$

C9. $9] \rightarrow 0] + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$

C10. $[x0] \rightarrow [x] + [x] + [x] + [x] + [x] + [x] + [x] + [x] + [x] + [x]$

C11. $[0] \rightarrow \varepsilon$

C12. $++ \rightarrow +$

C13. $+- \rightarrow -$

C14. $-\{+1 \rightarrow -1 - \{$

C15. $-\{-1 \rightarrow +1 - \{$

C16. $-\{\} \rightarrow \varepsilon$

C17. $(x) \cdot (+1 \rightarrow x + (x) \cdot ($

C18. $(x) \cdot (-1 \rightarrow -\{x\} + (x) \cdot ($

C19. $(x\cdot) \rightarrow \varepsilon$

C20. $\oplus + 1 \rightarrow +1 \oplus$

C21. $-1 \ominus \rightarrow \ominus - 1$

C22. $+1 - 1 \rightarrow$

C23. $-1 + 1 \rightarrow$

C24. $+1 \oplus \ominus - 1 \rightarrow$

C25. $\bullet x \oplus \ominus \bullet \rightarrow \bullet\langle x \bullet$

C26. $\bullet \oplus \ominus x \bullet \rightarrow - \bullet \langle -\{x\} \bullet$

C27. $\bullet \oplus \ominus \bullet \rightarrow 0$

C28. $\bullet x \langle \bullet \rightarrow x$

C29. $\bullet\langle +1 \rightarrow \bullet 1\langle$

C30. $\langle 0 \rightarrow 0\langle$

C31. $0\langle+1 \rightarrow 1\langle$

C32. $1\langle+1 \rightarrow 2\langle$

$\vdots$

C39. $8\langle+1 \rightarrow 9\langle$

C40. $9\langle+1 \rightarrow \langle+10$

Nous démontrons par le théorème suivant que $\mathcal{S}_{ASM}$ est confluent.

Théorème 78. *Le système $\mathcal{S}_{ASM}$ est confluent.*

Démonstration. Le système $\mathcal{S}_{ASM}$ est linéaire-gauche puisque aucune variable n'apparaît deux fois dans la partie gauche de chaque règle. Par inspection le système est non-chevauchant puisqu'il n'y a pas de chevauchement entre les règles. Le système $\mathcal{S}_{ASM}$ est donc orthogonal. Nous avons donc que le système $\mathcal{S}_{ASM}$ est confluent puisque par le théorème 29 tout système de réécriture orthogonal est confluent. □

4.4 Bases et nombre de règles

Les systèmes rencontrés au chapitre 3 qui permettent de représenter une base arbitraire nécessitent l'introduction de plusieurs règles qui prennent la forme de schémata de règles. L'avantage avec les deux systèmes que nous venons de présenter est qu'ils ne nécessitent pas l'introduction de schématas de règles. Si nous voulions passer de la base b à la base $b+1$, nous n'avons qu'à modifier une règle et à rajouter deux règles qui sont en fait les définitions associées au nouveau symbole. En pratique, nous retirerons une règle et en ajouterons trois.

Exemple 79. Pour passer de la base 10 à la base 11 dans l'un des deux systèmes précédents nous devons ajouter un symbole A à l'ensemble des symboles de la base. Nous ajoutons au sous-ensemble de règles de déconstruction la règle

$$A] \rightarrow 0] + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$$

Nous retirons au sous-ensemble de règles de construction la règle

$$9\langle +1 \rightarrow \langle +10$$

Nous ajoutons les deux règles suivantes au sous-ensemble de règles de construction

$$9\langle +1 \rightarrow A\langle$$

$$A\langle +1 \rightarrow \langle +10$$

Ainsi pour $\mathcal{S}_{ASMD}$ ou $\mathcal{S}_{ASM}$, nous avons une suite 1-imbriquée sur l'ensemble des règles ordonnées selon la base.

Pour $i \in \mathbb{N}^+$ la base, nous avons pour $\mathcal{S}_{ASMD}$ la fonction de progression f_{ASMD} qui nous donne le nombre de règles :

$$f_{ASMD}(i) = 2i + 45$$

Pour $i \in \mathbb{N}^+$ la base, nous avons pour $\mathcal{S}_{ASM}$ la fonction de progression f_{ASM} qui nous donne le nombre de règles :

$$f_{ASM}(i) = 2i + 20$$

Nous trouvons donc dans les deux cas $f'(i) = 2$. Ce qui est fort probablement la valeur la plus petite qu'il est possible d'avoir puisque nous avons besoin d'une définition du symbole pour la déconstruction et une pour la construction.

Il serait intéressant de trouver un système de réécriture où les entiers positifs apparaissant à la place de 45 et 20 soient des nombres minimaux.

4.5 La division et un théorème de Hardy et Wright

En utilisant le théorème suivant, il serait possible d'éviter d'avoir à choisir une précision p pour un système de réécriture incluant la division. Ce théorème apparaît dans (Hardy et Wright, 1979) à la section 9.2 page 111.

Théorème 80. *Le développement décimal d'un nombre rationnel p/q entre 0 et 1 se termine ou est périodique, et tout développement décimal qui se termine ou est périodique est égal à un nombre rationnel. Si $pgcd(p,q) = 1$, $q = 2^\alpha 5^\beta$ et $max(\alpha, \beta) = \mu$, alors le développement décimal se termine après μ chiffres. Si $pgcd(p,q) = 1$, $q = 2^\alpha 5^\beta Q$ où $Q > 1$, $pgcd(Q, 10) = 1$ et ν est l'ordre multiplicatif de 10 modulo Q, alors le développement décimal contient μ chiffres non-périodiques suivis de ν chiffres périodiques.*

Pour $pgcd(a,n) = 1$, l'ordre multiplicatif de a modulo n est le plus petit entier positif k tel que $a^k \equiv 1 \, mod \, n$.

Afin d'appliquer ce théorème, il faudrait s'assurer que nous arrivons à un terme ayant la forme p/q où $(p,q) = 1$. Trouver par un système de réécriture l'ordre multiplicatif est aussi un problème intéressant.

Le théorème ci-dessus est donné pour la base décimale. Un théorème qui s'applique pour toute autre base est aussi donné à la section 9.3 page 112 de (Hardy et Wright, 1979).

CHAPITRE V

VARIATIONS

En nous basant sur les systèmes développés au chapitre 4 nous présentons des systèmes de réécriture qui représentent différents types de systèmes de numération et d'arithmétiques. Nous allons présenter un système de numération mixte, un système de numération que l'on qualifie d'*exotique*, deux systèmes qui encodent des sommes d'unités et pour terminer nous présentons d'autres types de "multiplication" et évaluons la propriété de commutativité.

5.1 Système de numération mixte pour la mesure du temps

En nous inspirant des règles des deux systèmes du chapitre 4, nous pouvons représenter une somme d'unités dans une base positionnelle mixte, où à chaque position la base peut changer. Un exemple courant est la mesure temps qui peut s'exprimer en semaines (s), jours (j), heures (h), minutes (m) et secondes (s). Nous donnons le système de règles de construction pour cet exemple dans ce qui suit. Nous allons seulement donner des règles pour la collection des unités. Nous qualifierons cette base de *base temporelle.*

Un terme initial sera un terme ayant la forme suivante :

$$\bullet d_1 s d_2 j d_3 h d_4 m d_5 \langle S \bullet s$$

où S est une somme d'unité, par exemple $+1+1+1+1+1$ et où les d_i sont des suites de chiffres incluant le zéro.

Puisque les bases sont différentes à différentes positions et puisque les nombres qui apparaissent avant un changement de position sont 59 secondes ou minutes, 23 heures et 6 jours, il faut avoir des règles détaillées lorsqu'il y a un 9, un 3 et un 6 en dernière position. Voici les règles que nous qualifions d'*universelles* qui sont indépendantes de la position :

T1. $0\langle+1 \rightarrow 1\langle$

T2. $1\langle+1 \rightarrow 2\langle$

T3. $2\langle+1 \rightarrow 3\langle$

T4. $4\langle+1 \rightarrow 5\langle$

T5. $5\langle+1 \rightarrow 6\langle$

T6. $7\langle+1 \rightarrow 8\langle$

T7. $8\langle+1 \rightarrow 9\langle$

Voici les règles générales de collection :

T8. $\langle 0 \rightarrow 0\langle$

T9. $\langle m \rightarrow m\langle$

T10. $\langle h \rightarrow h\langle$

T11. $\langle j \rightarrow j\langle$

T12. $\langle s \rightarrow s\langle$

T13. $\langle 0 \rightarrow 0\langle$

T14. $\bullet x\langle\bullet \rightarrow x$

Sachant que y est une variable qui peut être remplacée par des nombres, nous avons les règles pour les termes où apparaît le sous-terme "$3\langle$". :

T15. $m3\langle+1 \rightarrow m4\langle$

T16. $my3\langle+1 \rightarrow my4\langle$

T17. $h3\langle+1 \rightarrow h4\langle$

T18. $hy3\langle+1 \rightarrow hy4\langle$

T19. $\bullet 3\langle+1 \rightarrow \bullet 4\langle$

T20. $\bullet y3\langle+1 \rightarrow \bullet y4\langle$

T21. $s3\langle+1 \rightarrow s4\langle$

T22. $j3\langle+1 \rightarrow j4\langle$

T23. $j13\langle+1 \rightarrow j14\langle$

T24. $j23\langle+1 \rightarrow \langle+1j0$

Nous avons les règles pour les termes où apparaît le sous-terme "6⟨". Notons que y est une variable qui peut être remplacée par des chiffres :

T25. $m6\langle+1 \rightarrow m7\langle$

T26. $my6\langle+1 \rightarrow my7\langle$

T27. $h6\langle+1 \rightarrow h7\langle$

T28. $hy6\langle+1 \rightarrow hy7\langle$

T29. $\bullet 6\langle+1 \rightarrow \bullet 7\langle$

T30. $\bullet y6\langle+1 \rightarrow \bullet y7\langle$

T31. $j6\langle+1 \rightarrow j7\langle$

T32. $j16\langle+1 \rightarrow j17\langle$

T33. $s6\langle+1 \rightarrow \langle+1s0$

Nous avons les règles pour les termes où apparaît le sous-terme "9⟨".

T34. $m9\langle+1 \rightarrow m10\langle$

T35. $my9\langle+1 \rightarrow my\langle+10$

T36. $m59\langle+1 \rightarrow \langle+1m0$

T37. $h9\langle+1 \rightarrow h10\langle$

T38. $hy9\langle+1 \rightarrow hy\langle+10$

T39. $h59\langle+1 \rightarrow \langle+1h0$

T40. $j9\langle+1 \rightarrow j10\langle$

T41. $j19\langle+1 \rightarrow j20\langle$

T42. $\bullet 9\langle+1 \rightarrow \bullet 10\langle$

T43. $\bullet y9\langle+1 \rightarrow \bullet y\langle+10$

L'exemple suivant nous montre comment se propagent les unités de position à position.

Exemple 81. $\bullet 39s6j23h59m59\langle+1 \bullet s$

$\rightarrow \bullet 39s6j23h59\langle+1m0 \bullet s$

$\rightarrow \bullet 39s6j23\langle+1h0m0 \bullet s$

$\rightarrow \bullet 39s6\langle+1j0h0m0 \bullet s$

$\rightarrow \bullet 39\langle+1s0j0h0m0 \bullet s$

$\rightarrow \bullet 3\langle+10s0j0h0m0 \bullet s$

$\rightarrow \bullet 4\langle 0sj0h0m0 \bullet s$

$\rightarrow \bullet 40\langle s0j0h0m0 \bullet s$

$\rightarrow \bullet 40s0\langle j0h0m0 \bullet s$

$\rightarrow \ldots$

$\rightarrow \bullet 40s0j0h0m0\langle \bullet s$

$\rightarrow 40s0j0h0m0s$

Ainsi, il est possible en utilisant les règles présentées pour l'arithmétique de déconstruire un nombre donné sous n'importe quelle base et obtenir une somme d'unités et ensuite de reconstruire un nombre dans une base standard ou mixte de notre choix,

en s'assurant de commencer la collection des unités avec un terme de la forme

$$\bullet 0s0j0h0m0\langle +1+1+...\bullet s$$

Nous pouvons aussi faire de l'arithmétique sur ces bases non-standards en utilisant nos règles décrivant les opérations d'addition, de soustraction, de multiplication et de division.

5.2 Système de numération exotique

Il est possible de se doter de systèmes de numération totalement différents des systèmes normalement utilisés. Grâce à un ensemble de règles il est possible d'avoir des systèmes de numérations irréguliers. Normalement dans un système de numération il est possible de construire des nombres dans une certaine base et pour chacun de ces nombres il est possible de le déconstruire afin de l'exprimer en somme d'unités. Le premier des deux systèmes suivants possède des règles de construction et de déconstruction et le deuxième système est composé de seulement de règles de construction. Une question intéressante à considérer dans le futur sera de déterminer pour quels systèmes de règles de construction il existe un ensemble de règles de déconstruction qui fait que le système est un système de numération.

Prenons un ensemble de règles de reconstruction donnée par :

E1. $0\langle +1 \rightarrow 1\langle$

E2. $1\langle +1 \rightarrow 2\langle$

E3. $2\langle +1 \rightarrow 23\langle$

E4. $3\langle +1 \rightarrow 4\langle$

E5. $4\langle +1 \rightarrow \langle +10$

E6. $\langle 0 \rightarrow 0\langle$

Ces règles définissent un nouveau type de système de numération. La suite en base décimale $\{1, 2, 3, 4, ...\}$ qui peut être écrite comme $\{+1, +1+1, +1+1+1, +1+1+1+1, ...\}$

sera représenté dans cette nouvelle base par :

$$\{1, 2, 23, 24, 230, 231, 232, 2323, 2324, 23230, 23231, 23232, 232323, ...\}$$

Afin que le système soit un système de numération, pour chaque nombre de la nouvelle base il faut pouvoir retrouver le nombre représenté par une somme d'unités à partir duquel il a été construit. Voici les règles de déconstruction.

E7. $[0] \rightarrow$

E8. $1] \rightarrow 0] + 1$

E9. $2] \rightarrow 0] + 1 + 1$

E10. $23] \rightarrow 0] + 1 + 1 + 1$

E11. $4] \rightarrow 3] + 1$

E12. $30] \rightarrow 4] + 1$

Voici un exemple de construction d'un nombre. Notons que nous introduisons devant la somme d'unités les symboles "0⟨".

Exemple 82.

$$\begin{aligned}
0\langle +1+1+1+1+1+1+1+1+1+1 &\rightarrow 1\langle +1+1+1+1+1+1+1+1+1+1 \\
&\rightarrow 2\langle +1+1+1+1+1+1+1+1+1 \\
&\rightarrow 23\langle +1+1+1+1+1+1+1 \\
&\rightarrow 24\langle +1+1+1+1+1+1 \\
&\rightarrow 2\langle +10+1+1+1+1+1 \\
&\rightarrow 23\langle 0+1+1+1+1+1 \\
&\rightarrow 230\langle +1+1+1+1+1 \\
&\rightarrow 231\langle +1+1+1+1 \\
&\rightarrow 232\langle +1+1+1 \\
&\rightarrow 2323\langle +1+1 \\
&\rightarrow 2324\langle +1 \\
&\rightarrow 232\langle +10 \\
&\rightarrow 2323\langle 0 \\
&\rightarrow 23230\langle
\end{aligned}$$

Voici la déconstruction du nombre obtenu à la dernière ligne de l'exemple ci-dessus où l'on retire le symbole "⟨".

Exemple 83.

$$
\begin{aligned}
[23230] &\rightarrow [2324]+1\\
&\rightarrow [2323]+1+1\\
&\rightarrow [230]+1+1+1+1+1\\
&\rightarrow [24]+1+1+1+1+1+1\\
&\rightarrow [23]+1+1+1+1+1+1+1\\
&\rightarrow [0]+1+1+1+1+1+1+1+1+1+1\\
&\rightarrow +1+1+1+1+1+1+1+1+1+1
\end{aligned}
$$

5.3 Systèmes qui encodent des sommes d'unités

Voici deux systèmes pour lesquels nous ne savons pas s'il y a un ensemble de règles de déconstruction qui ajouté aux règles de construction donnerait un système de numération.

5.3.1 Encodage apparaissant irrégulier

Prenons l'ensemble de règles suivantes :

$$
\begin{aligned}
&\text{F1.} && 0\langle+1 && \rightarrow && 1\langle\\
&\text{F2.} && 1\langle+1 && \rightarrow && 2\langle\\
&\text{F3.} && 2\langle+1 && \rightarrow && \langle+13\\
&\text{F4.} && 3\langle+1 && \rightarrow && 4\langle\\
&\text{F5.} && 4\langle+1 && \rightarrow && 5\langle\\
&\text{F6.} && 5\langle+1 && \rightarrow && \langle+10\\
&\text{F7.} && \langle 0 && \rightarrow && 0\langle\\
&\text{F8.} && \langle 3 && \rightarrow && 3\langle\\
&\text{F9.} && \bullet\langle+1 && \rightarrow && \bullet 0\langle+1
\end{aligned}
$$

En partant avec un terme $\bullet 0\langle+1+1+\ldots+1$, il est possible de trouver la suite en base

décimale $\{1, 2, 3, 4, ...\}$ qui sera représentée dans cette nouvelle base par :

$$\{1, 2, 13, 14, 15, 20, 21, 22, 133, 134, 135, 140, 141, 142, 153, 154, 155, 200, ...\}$$

5.3.2 Encodage non-confluent

Il est possible d'avoir un système qui encode les sommes d'unités de manière non-confluente. Voici un exemple :

N1. $0\langle+1 \rightarrow 1\langle$

N2. $1\langle+1 \rightarrow 2\langle$

N3. $2\langle+1 \rightarrow 3\langle$

N4. $3\langle+1 \rightarrow 6\langle$

N5. $3\langle+1 \rightarrow 4\langle$

N6. $4\langle+1 \rightarrow 5\langle$

N7. $5\langle+1 \rightarrow \langle+10$

N8. $6\langle+1 \rightarrow 7\langle$

N9. $7\langle+1 \rightarrow 8\langle$

N10. $8\langle+1 \rightarrow \langle+10$

N11. $\langle 0 \rightarrow 0\langle$

N12. $\bullet\langle+1 \rightarrow \bullet 0\langle+1$

Les règles qui entraînent la non-confluence sont les règles $3\langle+1 \rightarrow 34\langle$ et $3\langle+1 \rightarrow 6\langle$.

5.4 Commutativité

Dans le système de réécriture pour l'arithmétique nous avons défini la multiplication par la règle $(x \cdot +1 \rightarrow x + (x\cdot$, ce qui est communément appelé distributivité. Pour des sommes d'unités nous remarquons qu'il y a la propriété de commutativité qui apparaît naturellement, par exemple $+1 + 1 \cdot +1 + 1 + 1 = +1 + 1 + 1 \cdot +1 + 1$. Nous pourrions généraliser l'opération binaire de multiplication par la règle suivante avec k

une somme finie d'unités :

$$(x \cdot +1 \rightarrow x + k + (x\cdot$$

Par exemple si $k = +1 + 1$ nous avons :

$$(x \cdot +1 \rightarrow x + 1 + 1 + (x\cdot$$

Cette multiplication perd la propriété de commutativité pour tous les nombres sauf lorsque nous avons le même nombre d'unités de chaque côtés de l'opération.

Il est possible de définir une opération définie par deux règles où nous avons la propriété de commutativité si de chaque côté de l'opération nous avons la même parité, c'est-à-dire pair-pair ou impair-impair :

$$(x \cdot_1 +1 \rightarrow x + 1 + (x\cdot_2$$

$$(x \cdot_2 +1 \rightarrow x - 1 + (x\cdot_1$$

Exemple 84. Dans le cas où nous n'avons pas la même parité de chaque côté nous voyons qu'il n'y a pas la commutativité. Notons que nous utiliserons en plus les deux règles de disparition $(x\cdot_1) \rightarrow$ et $(x\cdot_2) \rightarrow$:

$$\begin{aligned}
1+1+1\cdot_1+1+1 &\rightarrow 1+1+1+1+(1+1+1\cdot_2+1 \\
&\rightarrow +1+1+1+1+1+1+1-1+(1+1+1\cdot_1) \\
&\rightarrow +1+1+1+1+1+1+1-1
\end{aligned}$$

$$\begin{aligned}
1+1\cdot_1+1+1+1 &\rightarrow 1+1+1+(+1+1\cdot_2+1+1 \\
&\rightarrow +1+1+1+1+1-1+(1+1\cdot_1+1) \\
&\rightarrow +1+1+1+1+1-1+1+1+1(\cdot_2) \\
&\rightarrow +1+1+1+1+1-1+1+1+1
\end{aligned}$$

CONCLUSION

Pour conclure, nous allons énoncer dans ce qui suit quelques problèmes à étudier, possibles extensions et possibles applications.

Les deux systèmes présentés au chapitre 4 transforment les nombres en unités. Afin de limiter l'espace utilisé, il est possible d'avoir un système modifié qui reconstruit les nombres immédiatement après une opération, cela limiterait l'espace requis, mais augmenterait le nombre d'exécutions, ce qui nous donnera un système de calcul plus lent.

Nous avons vu au chapitre 4 que l'introduction de la division dans notre système enlevait la propriété d'orthogonalité du système. Une question serait de savoir s'il est possible de définir une division dont toutes les règles sont linéaires-gauches. Il faudrait dans de futurs travaux fournir des preuves que les systèmes présentés au chapitre 4 sont confluents et possèdent la propriété de terminaison.

Dans (Walters et Zantema, 1995) le troisième système qui n'est pas confluent est comparable aux algorithmes communément utilisés. Pour cela il faut générer le schémata de règles. Trois possibilités sont discutées afin de construire le schémata de règles. Le système que nous avons construit est idéal pour la construction de schémata de règles puisqu'il est capable de réduire des termes du type $3 \cdot 5$ et $9 \cdot 9$ à leur forme normale respective. Bien que notre système n'est pas économique en espace et possède un grand nombre de réductions il reste utile à la production de schémata de règles et est intéressant du point de vu théorique en tant que système extrême. Dans ce système extrême, il serait intéressant de voir quelles règles entraîneraient une plus grande vitesse de calcul.

Un problème qui reste ouvert est de trouver un système de réécriture modélisant

l'arithmétique des nombres rationnels dans une base arbitraire qui inclut la division et qui fournit un algorithme de calcul efficace. Nous espérons avoir fait quelques pas dans cette direction.

Nous avons vu qu'en plus de modéliser l'arithmétique, certains systèmes de réécriture permettent de modéliser des concepts tels les systèmes de numération. La théorie de la réécriture pourrait donc fournir une approche alternative aux méthodes normalement utilisées et pourrait fournir des moyens pour généraliser l'arithmétique et les systèmes de numération. Ces considérations ouvrent la porte à plusieurs nouvelles sortes de bases et d'arithmétiques. La modélisation grâce aux systèmes de réécriture pourrait aussi s'étendre à d'autres objets et domaines mathématiques. Cela permettrait de généraliser d'autres structures mathématiques. Il sera intéressant d'étudier ces concepts dans de futurs travaux.

RÉFÉRENCES

Baader F. et T. Nipkow. 1998. *Term Rewriting and All That.* Cambridge : Cambridge University Press. 316 p.

Blanqui F., W. Delobel, S. Coupet-Grimal, S. Hinderer et A. Koprowski. 2006. « CoLoR, a Coq Library on Rewriting and termination ». In *Eighth International Workshop on Termination, WST-2006* (Seattle, État-Unis, 15-16 août 2006). http ://www.easychair.org/FLoC-06/WST.html.

Buchberger, B et R. Loos. 1982. « Algebraic Simplification ». In *Computer Algebra - Symbolic and Algebraic Computation*, Éd. B. Buchberger et al. New York : Springer, p. 11-44.

Chomsky N. 1956. « Three models for the description of language ». *IEEE Transactions on Information Theory*, vol. 2, no 3, p. 113-396.

Cirstea H. 2000. « Calcul de réécriture : fondement et applications ». Thèse de doctorat, Nancy, Université Henri Poincaré-Nancy 1, 211 p.

Cohen D. et P. Watson. 1991. « An efficient representation of arithmetic for term rewriting ». In *Rewriting Techniques and Applications : Actes du 4^e colloque Rewriting Techniques and Applications, RTA-91* (Como, Italie, 10-12 avril 1991), Éd. R. B. Book, Coll. « Lecture Notes in Computer Science », vol. 488, p. 240-251. New York : Springer.

Contejean E., C. Marché et L. Rabehasaina. 1997. « Rewrite Systems for Natural, Integral, and Rational Arithmetic ». In *Rewriting Techniques and Applications : Actes du 8^e colloque Rewriting Techniques and Applications, RTA-97* (Sitges, Espagne, 2-5 juin 1997), Éd. H. Comon, Coll. « Lecture Notes in Computer Science », vol. 488, p. 98-112. New York : Springer.

Cori R. et D. Lascar. 2003. *Fonctions récursives, théorème de Gödel, théorie des ensembles, théorie des modèles.* T.2 de *Logique mathématique.* Paris : Dunod. 347 p.

de Vries F. J. et J. Yamada. 1994. « On termination of rewriting with real numbers ». In *Functional Programming 2 : Actes de l'atelier Japan Society for Software Science and Technology, JSSST-94*, Éd. M. Takeichi, Coll. « Lecture Notes on Software Gaku 10 », vol. 10, p. 233-247. Tokyo : Kindai-kagaku-sya.

Dershowitz N., J. Jouannaud et J. W. Klop. 1993. « More Problems in Rewriting ». In *Rewriting Techniques and Applications : Actes du 5e colloque Rewriting Techniques and Applications, RTA-93* (Montréal, Canada, 16-18 juin 1993), Éd. C. Kirchner, Coll. « Lecture Notes in Computer Science », vol. 690, p. 468-487. New York : Springer.

Dershowitz N. et D. A. Plaisted. 2001. « Rewriting ». In *Handbook of Automated Reasoning*, Éd. J. A. Robinson et A. Vorondov, Coll. « Lecture Notes in Computer Science », vol. 1, p. 535-610. New York : Elsevier et MIT Press.

Dummit D. S. et R. M. Foote. 1999. *Abstract Algebra.* 2e édition. New York : John Wiley and Sons. 898 p.

Giesl J., R. Thiemann, P. Scheider-Kamp et S. Falke. 2004. « Automated termination proofs with AProVE ». In *Rewriting Techniques and Applications : Actes du 15e colloque Rewriting Techniques and Applications, RTA-04* (Aachen, Allemagne, 3-5 juin 2004), Éd. V. Oostrom, Coll. « Lecture Notes in Computer Science », vol. 3091, p. 210-220. New York : Springer.

Hardy G. H. et E. M. Wright. 1979. *An Introduction to the Theory of Numbers.* 5e édition. Oxford : Oxford University Press, 435 p.

Herbrand J. 1930. « Recherches sur la théorie de la démonstration ». Thèse de doctorat, Paris, Université de Paris.

Kennaway R. 1995. *Complete term rewrite systems for decimal arithmetic and other total recursive functions*, Présenté lors de l'atelier *Second International Workshop on Termination* (La Bresse, France, 29-31 Mai 1995). http ://citeseer.ist.psu.edu/332816.html.

Klop J. W. 1992. « Term rewriting systems ». In *Handbook of Logic in Computer Science*, Éd. S. Abramsky et al., vol. 2, p. 1-116. Oxford : Oxford University Press.

Labelle J. 1980. *Théorie des graphes.* Montréal : Modulo. 183 p.

Lindenmayer A. 1968. « Mathematical models for cellular interaction in development, Parts I and II ». *Journal of Theoretical Biology*, vol. 18, p. 280-315.

Marché C. 1996. « Normalized Rewriting An Alternative to Rewriting Modulo a Set of Equations ». *Journal of Symbolic Computation*, vol. 21, no 3, p. 253-288.

Plaisted D. A. 1993. « Equational Reasoning and Term Rewriting Systems ». In *The Handbook of Logic in Artificial Intelligence and Logic Programming*, Éd. D. Gabbay et al., vol. 1, p. 274-367. Oxford : Oxford University Press.

Prusinkiewicz P. et A. Lindenmayer. 1991. *The Algorithmic Beauty of Plants.* New York : Springer. 228 p.

TeReSe. 2003. Term Rewriting Systems. Éd. M. Bezem, et al., Coll. « Cambridge Tracts

in Theoretical Computer Science », vol. 55. Cambridge : Cambridge University Press.

Thue A. 1910. « Die Lösung eines Spezialfalles eines genrellen logischen Problems ». *Kra. Vidensk. Selsk. Skrifter. I. Mat.-Nat. Kl.*, no 8.

———. 1914. « Probleme uber Veranderungen von Zeichenreihen nach gegbenen Regeln ». *Kra. Vidensk. Selsk. Skrifter. I. Mat.-Nat. Kl.*, no 10.

———. 1977. *Selected Mathematical Papers.* Préf. de C. L. Siegel, Éd. T. Nagell et al. Oslo : Universitetsforlaget.

Walters H. R. 1991. « On Equal Terms : Implementing Algebraic Specifications ». Thèse de doctorat, Amsterdam, Universiteit van Amsterdam. 193 p.

———. 1994. A complete term rewriting system for decimal integer arithmetic. Rapport no. CS-R9435. Amsterdam : Centrum voor Wiskunde en Informatica. http ://ftp.cwi.nl/CWIreports/1994/CS-R9435.pdf.

Walters H. R. et H. Zantema « Rewrite systems for integer arithmetic ». In *Rewriting Techniques and Applications : Actes du 6^e colloque Rewriting Techniques and Applications, RTA-95* (Kaiserslautern, Germany, 5-7 Avril), Éd. J. Hsiang, Coll. « Lecture Notes in Computer Science », vol. 914, p. 324-338. New York : Springer.

INDEX

algorithme de Buchberger, 29
alphabet, 9
arête, 8
arborescence, 8
arbre, 8
axiomes de Peano, 22

chaîne, 8
chaîne simple, 8
chevauchement
    d'occurences, 16
    de règles, 16
complétion de Knuth-Bendix, 29
compteur de division, 32
confluent, 15
    faiblement, 15
contexte, 13
cycle, 8

de précision initial, 32
degré, 9
descendants, 9

encodage
    irrégulier, 59
    non-confluent, 60
étape de réécriture, 13

fermeture de Kleene, 9
fermeture réflexive, 8
fermeture transitive, 8
feuille, 9
fonction de progression, 21
forêt, 8
forme normale, 15

graphe connexe, 8
graphe simple, 8

L-systèmes, 2
linéaire-gauche, 16
longueur d'un mot, 9
longueur d'une chaîne, 8

mot, 9
mot vide, 9
mots de longueur k, 9

occurrence, 15
ordre de réduction, 18
    compatible, 18
orthogonale, 17

parent, 9
patron, 16

règle de réécriture, 12
racine, 8
redex, 13

relation, 7
    bien-fondée, 7
    irréflexive, 7
    ordre bien-fondée, 8
    ordre partiel strict, 7
    transitive, 7

schémata de règles, 28
signature, 12
sommet, 8
sous-terme, 15
substitution, 12
suite, 7
    R-descendante, 7
    finie, 7
suite r-imbriquée, 21
suite d'étapes de réécriture indexées, 14
symboles, 32
symboles apparaissant dans les règles, 32
système abstrait de réécriture, 11
système de numération
    exotique, 57
    mixte, 20
    mixte pour le temps, 53
    positionnelle, 20
système de réécriture de termes, 13
système de réécriture pour l'arithmétique
    Cohen et Watson, 24
    Contejean, Marché et Rabehasaina, 29
    de Vrie et Yamada, 29
    Kennaway, 26
    Peano, 22
    Walters, 24
    Walters et Zantema, 27

termes, 12
termes ambiants, 32
termes du système T_{ASMD}, 35
termes du système T_{ASM}, 47
termes initial
    de $\mathcal{S}_{ASMD}$, 34
    de $\mathcal{S}_{ASM}$, 47
terminaison, 17

variables, 12, 32