Notations et préliminaires

1.1 Mots et langages et automates

Un *alphabet* A est un ensemble fini d'éléments appelés *lettres*. Un *mot* sur l'alphabet A est un élément du *monoïde libre* A^*, c'est-à-dire une suite finie de lettres de A ou l'identité 1_{A^*}, appelée *mot vide* de A^*.

La *longueur* d'un mot u sur un alphabet A, notée $|u|$, est le nombre de lettres qui composent u. Si a est une lettre de A, alors nous notons $|u|_a$ le nombre d'occurrences de a dans u. Par exemple $|abaa| = 4$, $|abaa|_a = 3$ et $|abaa|_b = 1$.

Un *langage* de mots sur l'alphabet A est un sous-ensemble, fini ou infini, de A^*.

Dans ce mémoire, plusieurs sortes d'automates sont définis : les automates sur un alphabet, les automates dont les sorties peuvent être étiquetées par un mot, les $\mathbb{K}$-automates ainsi que les transducteurs. Nous donnons ici la définition des automates booléens les plus simples.

Définition 1. *Un* automate fini $\mathcal{A}$ *sur un alphabet* A *est un quintuplet* $\langle Q, A, E, I, T\rangle$ *où :*

(i) Q *est un ensemble fini dont les éléments sont appelés* états *;*

(ii) E *est un ensemble d'arcs entre des éléments de* Q *étiquetés par* A*, ce sont les* transitions *de* $\mathcal{A}$ *;*

(iii) I *est un sous-ensemble de* Q*, l'ensemble des* états initiaux *;*

(iv) T *est un sous-ensemble de* Q*, l'ensemble des* états finaux.

Dans ce mémoire, nous ne traitons pas le cas des automates infinis et nous appellons automates les automates finis. La Figure 1.1 montre comment nous représentons les automates. Les états initiaux sont repérés par des flèches entrantes, les états finaux par des flèches sortantes.

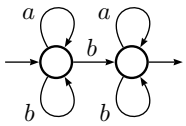

FIGURE 1.1 – Un automate, $\mathcal{B}_1$ sur l'alphabet $\{a, b\}$

Un état p d'un automate $\mathcal{A}$ est dit *acccessible* s'il existe un chemin dans l'automate d'un état initial de I à p. Il est *co-accessible* s'il existe un chemin de p à un élément de T. Un automate $\mathcal{A}$ est *accessible* si tous ses états le sont, *co-accessible* si tous ses états le sont et *émondé* si il est à la fois accessible et co-accessible.

Un *calcul réussi* de $\mathcal{A}$ est un chemin qui commence dans un état initial et qui se termine dans un état final. Le *langage reconnu* par $\mathcal{A}$, noté $|\mathcal{A}|$ est le sous-ensemble de A^* des étiquettes des calculs réussis :

$$|\mathcal{A}| = \left\{ u \in A^* \mid i \xrightarrow{u} t,\ i \in I,\ t \in T \right\} \ .$$

Par exemple le langage reconnu par l'automate $\mathcal{B}_1$ de la Figure 1.1 est $|\mathcal{B}_1| = A^*\{b\}A^*$, c'est-à-dire le langage des mots sur A qui contiennent au moins un b.

Un langage de A^* est dit *reconnaissable* s'il existe un automate fini qui le reconnaît.

Un automate est dit *déterministe* s'il a un unique état initial i et si, pour tout état p, il existe, au plus, une unique transition sortante étiquetée par chaque lettre de A. Un automate est dit *ambigu* si il existe un état q et un mot u tel qu'il y ait plus d'un chemin d'un état initial à q étiqueté u. Un automate déterministe est non-ambigu.

Proposition 1. *Il est possible de transformer tout automate fini $\mathcal{A}$ en un automate fini déterministe $\mathcal{B}$ qui reconnaît le même langage. Si l'automate $\mathcal{A}$ a n états, il est possible de construire $\mathcal{B}$ avec au plus 2^n états.*

Les langages reconnaissables sont donc les langages reconnus par les automates finis déterministes. Il peut y avoir plusieurs automates déterministes qui reconnaissent le même langage.

1.2 Expressions rationnelles

Afin de définir les expressions rationnelles et les langages rationnels sur un alphabet, nous définissons tout d'abord les opérations rationnelles sur les sous-ensembles d'un monoïde :

Définition 2. *Les opérations rationnelles sur les sous-ensembles d'un monoïde M sont l'union, le produit d'ensembles et l'étoile définie par :*

$$X^* = \{1_M\} \cup \bigcup_{n>0} X^n \ .$$

La famille des sous-ensembles rationnels *de M, notée* $\mathrm{Rat}\, M$*, est la fermeture des ensembles finis par ces opérations.*

En particulier, nous appelons *langages rationnels* sur A^* les sous-ensembles rationnels de A^*. Les expressions rationnelles sont un moyen de décrire ces langages en notant les opérations rationnelles successives pour obtenir le sous-ensemble rationnel. Plus précisément :

Définition 3. *Une* expression rationnelle E *sur A est une formule bien parenthésée obtenue inductivement par la grammaire suivante dans lesquelles les atomes sont les lettres de A et les expressions prédéfinies* 0 *et* 1 *:*

$$\mathsf{E} \to \{a \mid a \in A\} \mid 0 \mid 1 \mid (\mathsf{E} + \mathsf{E}) \mid (\mathsf{E} \cdot \mathsf{E}) \mid (\mathsf{E}^*) \ .$$

L'ensemble des expression rationnelles sur un alphabet A est noté $\mathsf{RatE(A)}$.

Comme toute expression bien parenthésée, une expression rationnelle E peut être représentée par un arbre syntaxique, que nous notons T_{E}. La *profondeur* d'une expression E, notée $\mathsf{d(E)}$, est la profondeur de l'arbre syntaxique, plus précisément :

$$\begin{gathered}
\mathsf{d}(0) = 0 \ , \ \mathsf{d}(1) = 1 \ , \ \mathsf{d}(a) = 1 \ , \\
\mathsf{d}(\mathsf{E} + \mathsf{F}) = max(\mathsf{d}(\mathsf{E}), \mathsf{d}(\mathsf{F})) + 1 \ , \\
\mathsf{d}(\mathsf{E} \cdot \mathsf{F}) = max(\mathsf{d}(\mathsf{E}), \mathsf{d}(\mathsf{F})) + 1 \ , \\
\mathsf{d}(\mathsf{E}^*) = \mathsf{d}(\mathsf{E}) + 1 \ .
\end{gathered}$$

Exemple 1. La formule $((a+b)^* \cdot b) \cdot (a+b)^*$ est une expression rationnelle de profondeur 5.

Par défaut, lorsque les parenthèses seront absentes, c'est le parenthésage à gauche qui sera utilisé : $\mathsf{E} \cdot \mathsf{F} \cdot \mathsf{G} = ((\mathsf{E} \cdot \mathsf{F}) \cdot \mathsf{G})$. Le justification de ce choix sera donnée par l'analyse de l'influence du parenthésage des produits sur les termes dérivés (et les termes dérivés cassés). La *longueur littérale* de l'expression E, notée $\ell(\mathsf{E})$, est le nombre d'occurrences de lettres de l'alphabet A

dans celle-ci. Le *langage dénoté* par une expression rationnelle E, noté $|\mathsf{E}|$, est défini inductivement par :

$$|0| = \emptyset\,, \quad |1| = \{1_{A^*}\}\,, \quad |a| = \{a\}\ ,$$
$$|(\mathsf{E}+\mathsf{F})| = |\mathsf{E}| \cup |\mathsf{F}|\ ,$$
$$|(\mathsf{E}\cdot\mathsf{F})| = |\mathsf{E}||\mathsf{F}|\ ,$$
$$|(\mathsf{E}^*)| = |\mathsf{E}|^*\ .$$

Exemple 2 (*Ex. 1 cont.*)**.** Le langage dénoté par l'expression rationnelle $((a+b)^*\cdot b)\cdot(a+b)^*$ est le langage A^*bA^* des mots contenant au moins une occurrence de b.

Les langages rationnels sur A^* sont exactement les langages dénotés par une expression rationnelle sur A^*. Il peut y avoir plusieurs expressions rationnelles qui dénotent le même langage rationnel. Deux expressions rationnelles sont dites *équivalentes* si elles dénotent le même langage.

Afin de simplifier les manipulations sur les expressions rationnelles, nous définissons des identités triviales, que nous notons **(T)**, et pour lesquelles les expressions rationnelles sont équivalentes de manière évidente et tous les calculs réalisés par la suite seront réalisés modulo **(T)** :

$$\mathsf{E}+0 \equiv 0+\mathsf{E} \equiv \mathsf{E}\,, \quad \mathsf{E}\cdot 0 \equiv 0\cdot\mathsf{E} \equiv 0\,, \quad \mathsf{E}\cdot 1 \equiv 1\cdot\mathsf{E} \equiv \mathsf{E}\,, \quad 0^* \equiv 1\ .(\mathbf{T})$$

Une expression est dite réduite par ces identités triviales si elle ne contient aucune sous-expression de la forme d'un membre gauche d'une de ces identités. En particulier, ces expressions permettent d'assurer qu'une expression différente de 0 n'a pas pour sous-expression 0, ce qui va nous permettre de traiter ce cas différemment lors de nos preuves par induction sur la profondeur des expressions.

Il apparaît que toute expression rationnelle E peut être réécrite dans une expression réduite équivalente E' et que cette expression réduite est unique indépendamment de l'ordre dans lequel sont appliquées les différentes identités.

Finalement, il est possible de définir inductivement un booléen, appelé le *terme constant* d'une expression rationnelle E, noté $\mathsf{c}(\mathsf{E})$, par :

$$\mathsf{c}(0) = 0\,, \quad \mathsf{c}(1) = 1\,, \quad \forall a \in A \quad \mathsf{c}(a) = 0\ ,$$
$$\mathsf{c}(\mathsf{F}+\mathsf{G}) = \mathsf{c}(\mathsf{F}) \vee \mathsf{c}(\mathsf{G})\,, \quad \mathsf{c}(\mathsf{F}\cdot\mathsf{G}) = \mathsf{c}(\mathsf{F}) \wedge \mathsf{c}(\mathsf{G})\,, \quad \mathsf{c}(\mathsf{F}^*) = 1\ .$$

Proposition 2. *Le terme constant d'une expression* E *est égal à* 1 *si, et seulement si, le mot vide* 1_{A^*} *est dans le langage* $|\mathsf{E}|$.

Nous avons pris soin de donner des définitions sur les expressions rationnelles de manière inductive car nous allons pouvoir ainsi utiliser les objets définis dans des preuves par induction. Ces définitions inductives permettent également de faire les calculs sur l'arbre syntaxique de bas en haut, ce qui sera utilisé lorsque nous donnerons une méthode pour calculer l'automate des termes dérivés.

Finalement nous rappelons le théorème de Kleene qui est le théorème fondamental à la base de la théorie des automates :

Théorème 3 (Kleene)**.** *Pour tout alphabet fini, l'ensemble des langages rationnels est égal à l'ensemble des langages reconnaissables.*

Il est donc possible de représenter un même langage rationnel par un automate fini qui le reconnaît ou par une expression rationnelle qui le dénote. C'est ce théorème qui nous amène à envisager la possibilité de construire des automates à partir d'expressions rationnelles, ou de construire des expressions rationnelles à partir d'automates en gardant le langage représenté.

1.3 Automate des positions

Dans cette section nous allons donner l'exemple le plus classique d'automate construit à partir d'une expression rationnelle. Cet automate, que nous appellerons *l'automate des positions* d'une expressions rationnelle, a été défini simultanément et indépendamment par Glushkov [26] – il est, pour cette raison, souvent appelé *automate de Glushkov* de l'expression – et MacNaughton et Yamada [34]. Cet automate est intrinsèquement très lié à l'expression rationnelle et en particulier il existe plusieurs algorithmes très différents pour le construire à partir de l'expression. Nous décrirons ici la méthode originale pour le construire.

L'automate des positions est *standard*, c'est-à-dire qu'il a un unique état initial i et qu'aucune transition ne pointe vers i. Dans [43], l'automate des positions est également appelé *automate standard* de l'expression et, il est possible de le calculer par une méthode inductive à l'aide de constructions de bases sur les automates standard pour les opérations rationnelles.

Pour construire l'automate, nous devons tout d'abord définir les positions d'une expression rationnelle. Une expression rationnelle E peut être linéarisée - c'est-à-dire que chaque lettre n'apparaît qu'au plus une fois dans E – sur un alphabet plus grand (de la taille de l'expression) en une expression rationnelle $\overline{\mathsf{E}}$ en indiçant les occurrences de chaque lettre. Le nouvel alphabet ainsi créé est appelé *l'ensemble des positions* de l'expression E. Une autre

manière de voir ces positions est sur l'arbre syntaxique : les positions sont les feuilles qui sont étiquetées par une lettre de l'alphabet. L'ensemble des positions est noté $\mathrm{fp}(\mathsf{E})$ et appelé l'ensemble des *feuilles propres* de E.

Définition 4. *Soit* E *une expression rationnelle. L'ensemble* $\mathsf{First}(\mathsf{E})$ *des premières positions de* E*, l'ensemble* $\mathsf{Last}(\mathsf{E})$ *des dernières positions de* E *et l'ensemble* $\mathsf{Follow}(l,\mathsf{E})$ *des positions qui suivent* l *dans* E*, où* l *est une position, sont définis inductivement par :*

$$\begin{aligned}
\mathsf{First}(0) = \mathsf{First}(1) &= \emptyset \ , \\
\mathsf{First}(a) &= \{a\},\ a \in \mathrm{fp}(\mathsf{E}) \ , \\
\mathsf{First}(\mathsf{F}+\mathsf{G}) &= \mathsf{First}(\mathsf{F}) \cup \mathsf{First}(\mathsf{G}) \ , \\
\mathsf{First}(\mathsf{F}\cdot\mathsf{G}) &= \mathsf{First}(\mathsf{F}) \cup \mathsf{c}(\mathsf{F})\mathsf{First}(\mathsf{G}) \ , \\
\mathsf{First}(\mathsf{F}^*) &= \mathsf{First}(\mathsf{F}) \ ,
\end{aligned}$$

$$\begin{aligned}
\mathsf{Last}(0) = \mathsf{Last}(1) &= \emptyset \ , \\
\mathsf{Last}(a) &= \{a\},\ a \in \mathrm{fp}(\mathsf{E}) \ , \\
\mathsf{Last}(\mathsf{F}+\mathsf{G}) &= \mathsf{Last}(\mathsf{F}) \cup \mathsf{Last}(\mathsf{G}) \ , \\
\mathsf{Last}(\mathsf{F}\cdot\mathsf{G}) &= \mathsf{Last}(\mathsf{G}) \cup \mathsf{c}(\mathsf{G})\mathsf{Last}(\mathsf{F}) \ , \\
\mathsf{Last}(\mathsf{F}^*) &= \mathsf{Last}(\mathsf{F}) \ ,
\end{aligned}$$

$$\begin{aligned}
\mathsf{Follow}(l,0) = \mathsf{Follow}(l,1) &= \emptyset \ , \\
\mathsf{Follow}(l,a) &= \emptyset,\ a \in \mathrm{fp}(\mathsf{E}) \ , \\
\mathsf{Follow}(l,\mathsf{F}+\mathsf{G}) &= \mathsf{Follow}(l,\mathsf{F}) \cup \mathsf{Follow}(l,\mathsf{G}) \ , \\
\mathsf{Follow}(l,\mathsf{F}\cdot\mathsf{G}) &= \begin{cases} \mathsf{Follow}(l,\mathsf{F}) \cup \mathsf{c}(\mathsf{F})\mathsf{Follow}(l,\mathsf{G}) \cup \mathsf{First}(\mathsf{G}) & \textit{si } l \in \mathsf{Last}(\mathsf{F}) \\ \mathsf{Follow}(l,\mathsf{F}) \cup \mathsf{c}(\mathsf{F})\mathsf{Follow}(l,\mathsf{G}) & \textit{sinon} \end{cases} \ , \\
\mathsf{Follow}(l,\mathsf{F}^*) &= \begin{cases} \mathsf{Follow}(l,\mathsf{F}) \cup \mathsf{First}(\mathsf{F}) & \textit{si } l \in \mathsf{Last}(\mathsf{F}) \\ \mathsf{Follow}(l,\mathsf{F}) & \textit{sinon} \end{cases} \ .
\end{aligned}$$

C'est-à-dire que $\mathsf{First}(\mathsf{E})$ est l'ensemble des positions qui sont les premières positions d'un mot dans le langage dénoté par l'expression linéarisée $\overline{\mathsf{E}}$. L'ensemble $\mathsf{Last}(\mathsf{E})$ est l'ensemble des positions qui sont les dernières positions d'un mot dans $|\overline{\mathsf{E}}|$. Si l est une position alors $\mathsf{Follow}(l,\mathsf{E})$ est l'ensemble

des positions telles qu'il existe un mot dans $|\overline{\mathsf{E}}|$ pour lequel elles suivent la position l.

Pour la suite, et lorsqu'il n'y aura pas de confusion possible, nous nous autoriserons à parler de positions dans les mots du langage $|\mathsf{E}|$ pour parler des positions correspondantes dans les mots de $|\overline{\mathsf{E}}|$.

Exemple 3 (*Ex. 1 cont.*)**.** Considérons l'expression rationnelle $((a+b)^* \cdot b) \cdot (a+b)^*$. Afin de différencier les feuilles qui représentent une même lettre, nous donnons un indice à chaque occurrence de lettres dans cette expression rationnelle – linéarisation – : $\mathsf{E}_1 = ((a_1 + b_1)^* \cdot b_2) \cdot (a_2 + b_3)^*$.

Nous avons alors :

$$\begin{aligned} \mathsf{First}(\mathsf{E}_1) &= \{a_1, b_1, b_2\} \ , \\ \mathsf{Last}(\mathsf{E}_1) &= \{b_2, a_2, b_3\} \ , \\ \mathsf{Follow}(a_1, \mathsf{E}_1) = \mathsf{Follow}(b_1, \mathsf{E}_1) &= \{a_1, b_1, b_2\} \ , \\ \mathsf{Follow}(a_2, \mathsf{E}_1) = \mathsf{Follow}(b_2, \mathsf{E}_1) = \mathsf{Follow}(b_3, \mathsf{E}_1) &= \{a_2, b_3\} \ . \end{aligned}$$

Il est alors possible de définir l'automate des positions :

Définition 5. L'automate des positions *d'une expression rationnelle* E *est l'automate tel que :*
(i) *les états sont les positions de* E *et l'unique état initial* i *;*
(ii) *les états finaux sont les positions dans* $\mathsf{Last}(\mathsf{E})$ *;*
(iii) *il y a une transition de* i *aux positions* l *étiquetée par* a*, la lettre associée à* l*, si* $l \in \mathsf{First}(\mathsf{E})$*.*
(iv) *il y a une transition entre une position* l *et une position* l' *étiquetée par* a*, la lettre associée à* l'*, si* $l' \in \mathsf{Follow}(l, \mathsf{E})$*.*

Exemple 4 (*Ex. 1 cont.*)**.** L'automate des positions de l'expression $((a + b)^* \cdot b) \cdot (a + b)^*$ est montré sur la Figure 1.2. On remarque en particulier grâce à cet exemple que les transitions entrantes dans un état sont toujours étiquetées par la lettre de la position représentée par l'état.

Proposition 4. *L'automate des positions d'une expression rationnelle* E *est un automate standard à* $\ell(\mathsf{E}) + 1$ *états qui reconnaît* $|\mathsf{E}|$*.*

1.4 Forme normale

Dans la suite, afin de donner une borne au nombre de termes dérivés cassés d'une expression rationnelle, nous sommes amenés à considérer une

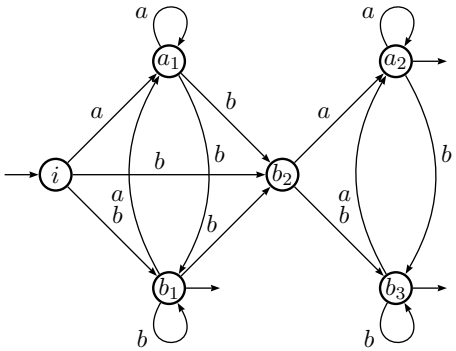

FIGURE 1.2 – L'automate des positions de $((a+b)^* \cdot b) \cdot (a+b)^*$

classe particulière d'expressions rationnelles : les expressions rationnelles en forme normale. Ces expressions ont été définies par Brüggemann-Klein [10] et elles ont l'intérêt de permettre une construction quadratique de l'automate des positions. En outre il est possible de construire à partir d'une expression rationnelle quelconque une expression équivalente en forme normale dont l'automate des positions est identique en temps linéaire. La définition des expressions en forme normale donne donc un algorithme quadratique pour construire l'automate des positions d'une expression.

Nous donnons dans ce mémoire une définition légèrement différente de celle donnée dans [10] des expressions en forme normale mais qui ne change pas les constructions qui permettent d'obtenir une expression en forme normale à partir de n'importe quelle expression rationnelle.

Définition 6 ([10])**.** *Une expression rationnelle* E *est* en forme normale *si, et seulement si, pour toute sous-expression* F^* *de* E*, on a* $\mathsf{c}(\mathsf{F}) = 0$ *.*

Si cette définition décrit les expressions en forme normale, ce qui nous intéresse particulièrement c'est la transformation d'expressions rationnelles quelconques en expressions en forme normale. Pour cela, dans [10], il y a deux opérateurs $\bullet$ et $\circ$ définis par induction. L'opérateur $\bullet$ transforme l'expression en une expression en forme normale en appelant l'opérateur $\circ$ lorsque $\bullet$ est appelé sur des expressions étoilées. L'opérateur $\circ$ transforme, à l'intérieur d'une expression rationnelle, les produits d'expressions rationnelles dont le terme constant est non nul en la somme des mêmes expressions.

Dans cette section nous donnons les définitions inductives originales de [10] puis nous proposons un mode de calcul légèrement différent pour la forme normale d'un expression rationnelle. Cette modification du calcul nous permettra par la suite d'établir des preuves inductives sur les expressions en forme normale.

Les définitions inductives de [10] sont les suivantes :

$$0^\circ = 0\,, \quad 1^\circ = 0\,, \quad \text{for all } a \in A \quad a^\circ = a \ , \tag{1.1}$$

$$(\mathsf{F} + \mathsf{G})^\circ = \mathsf{F}^\circ + \mathsf{G}^\circ \ , \tag{1.2}$$

$$(\mathsf{F} \cdot \mathsf{G})^\circ = \begin{cases} \mathsf{F} \cdot \mathsf{G} & \text{si } \mathsf{c}(\mathsf{F}) = \mathsf{c}(\mathsf{G}) = 0 \ , \\ \mathsf{F}^\circ \cdot \mathsf{G} & \text{si } \mathsf{c}(\mathsf{F}) = 0 \text{ et } \mathsf{c}(\mathsf{G}) = 1 \ , \\ \mathsf{F} \cdot \mathsf{G}^\circ & \text{si } \mathsf{c}(\mathsf{F}) = 1 \text{ et } \mathsf{c}(\mathsf{G}) = 0 \ , \\ \mathsf{F}^\circ + \mathsf{G}^\circ & \text{si } \mathsf{c}(\mathsf{F}) = \mathsf{c}(\mathsf{G}) = 1 \ , \end{cases} \tag{1.3}$$

$$(\mathsf{F}^*)^\circ = \mathsf{F}^\circ \ . \tag{1.4}$$

et :

$$0^\bullet = 0\,, \quad 1^\bullet = 1\,, \quad \text{pour tout } a \in A \quad a^\bullet = a \ , \tag{1.5}$$

$$(\mathsf{F} + \mathsf{G})^\bullet = \mathsf{F}^\bullet + \mathsf{G}^\bullet \ , \tag{1.6}$$

$$(\mathsf{F} \cdot \mathsf{G})^\bullet = \mathsf{F}^\bullet \cdot \mathsf{G}^\bullet \ , \tag{1.7}$$

$$(\mathsf{F}^*)^\bullet = ((\mathsf{F}^\bullet)^\circ)^* \ . \tag{1.8}$$

Il est alors possible d'établir :

Proposition 5 ([10])**.** *Pour toute expression rationnelle* E*, l'expression* $\mathsf{E}^\bullet$ *est en forme normale et est équivalente à* E*.*

Les définitions inductives (1.1)-(1.8) posent le problème suivant : même si les sous-expressions d'une expression en forme normale sont en forme normale, il n'y a pas la même notion d'héritage pour la transformation réalisée par l'opérateur $\circ$. Par exemple :

$$((a^*b^*)^*)^\circ = a + b \qquad \text{alors que} \qquad (c\,(a^*b^*)^*)^\circ = c\,(a^*b^*)^* \ .$$

Cette difficulté est facilement surmontable si l'on se rend compte que dans le calcul de $\mathsf{E}^\bullet$, l'opérateur $\circ$ n'est utilisé qu'en conjonction de l'opérateur $\bullet$ (in (1.8)). Il est alors possible de définir un nouvel opérateur $\square$ inductivement par :

$$\forall \mathsf{E} \in \mathsf{RatE}(\mathsf{A}) \qquad \mathsf{E}^\square = (\mathsf{E}^\bullet)^\circ \ .$$

Il est alors possible de donner une définition inductive de $\square$ grâce aux propriétés suivantes de $\circ$ et de $\bullet$:

Proposition 6 ([2]). *Soit* E *une expression rationnelle. On a :*

$$\mathsf{c}(\mathsf{E}^\circ) = 0 \ , \tag{1.9}$$

$$\mathsf{c}(\mathsf{E}) = 0 \quad \Longrightarrow \quad \mathsf{E} = \mathsf{E}^\circ \qquad \textit{(c'est-à-dire :}\ \mathsf{E} = \mathsf{E}^\circ \Leftrightarrow \mathsf{c}(\mathsf{E}) = 0) \ , \tag{1.10}$$

$$(\mathsf{E}^\circ)^\circ = \mathsf{E}^\circ \qquad \textit{(i.e., } \circ \textit{ est idempotent)} \ , \tag{1.11}$$

$$\mathsf{c}(\mathsf{E}^\bullet) = \mathsf{c}(\mathsf{E}) \ . \tag{1.12}$$

Proposition 7 ([2]). *Soient* F *et* G *deux expressions rationnelles.On a :*

$$(\mathsf{F} + \mathsf{G})^\square = \mathsf{F}^\square + \mathsf{G}^\square \ , \tag{1.13}$$

$$(\mathsf{F} \cdot \mathsf{G})^\square = \begin{cases} \mathsf{F}^\square + \mathsf{G}^\square & \textit{si} \quad \mathsf{c}(\mathsf{F}) = \mathsf{c}(\mathsf{G}) = 1 \\ \mathsf{F}^\bullet \cdot \mathsf{G}^\bullet & \textit{sinon} \end{cases} \ , \tag{1.14}$$

$$(\mathsf{F}^*)^\square = \mathsf{F}^\square \ . \tag{1.15}$$

Démonstration. Des équations (1.2) et (1.6), on a (1.13).
Si $(\mathsf{F} \cdot \mathsf{G})^\square = ((\mathsf{F} \cdot \mathsf{G})^\bullet)^\circ = (\mathsf{F}^\bullet \cdot \mathsf{G}^\bullet)^\circ$; le reste de la preuve de (1.14) demande un examen des cas suivants (en utilisant implicitement le fait que $\mathsf{c}(\mathsf{H}^\bullet) = \mathsf{c}(\mathsf{H})$ pour toute expression H) :

Si $\mathsf{c}(\mathsf{F}) = 0$ ou $\mathsf{c}(\mathsf{G}) = 0$ alors $\mathsf{c}(\mathsf{E}) = 0$ et, l'équation (1.10) donne donc $(\mathsf{F}^\bullet \cdot \mathsf{G}^\bullet)^\circ = \mathsf{F}^\bullet \cdot \mathsf{G}^\bullet$.

Si $\mathsf{c}(\mathsf{F}) = \mathsf{c}(\mathsf{G}) = 1$, alors $(\mathsf{F}^\bullet \cdot \mathsf{G}^\bullet)^\circ = (\mathsf{F}^\bullet)^\circ + (\mathsf{G}^\bullet)^\circ = \mathsf{F}^\square + \mathsf{G}^\square$.

Finalement, $(\mathsf{F}^*)^\square = ((\mathsf{F}^*)^\bullet)^\circ = (((\mathsf{F}^\bullet)^\circ)^*)^\circ = ((\mathsf{F}^\bullet)^\circ)^\circ = (\mathsf{F}^\bullet)^\circ$ par l'équation (1.11). □

Corollaire 8 ([2]). *L'expression* $\mathsf{E}^\square$ *peut être calculée par induction sur la profondeur de* E *en utilisant les équations (1.13)–(1.15) et les cas de base suivants :*

$$0^\square = 0 \, , \quad 1^\square = 0 \, , \qquad \forall a \in A \quad a^\square = a \ .$$

Il est maintenant clair que (1.13)–(1.15), avec (1.5)–(1.7) et (1.8) réécrit en :

$$(\mathsf{F}^*)^\bullet = (\mathsf{F}^\square)^* \ , \tag{1.8'}$$

est une nouvelle description inductive du calcul de $\mathsf{E}^\bullet$, à la différence que toute sous-expression de $\mathsf{E}^\square$ est en forme normale. Par exemple :

$$((a^*b^*)^*)^\square = a + b \qquad \text{et} \qquad (c\,(a^*b^*)^*)^\square = c\,(a+b)^* \ .$$

Cette nouvelle description donne, comme la description originale, un algorithme linéaire pour calculer $\mathsf{E}^\bullet$.

Comme dans la Proposition 6 pour l'opérateur $\circ$, nous pouvons donner des propriétés qui nous seront utiles par la suite pour les opérateurs $\square$ et $\bullet$.

Proposition 9 ([2]). *Soit* E *une expression rationnelle, alors on a :*

$$\mathsf{c}(\mathsf{E}^{\square}) = 0 \ , \tag{1.16}$$

$$\mathsf{c}(\mathsf{E}) = 0 \iff \mathsf{E}^{\bullet} = \mathsf{E}^{\square} \ , \tag{1.17}$$

$$\ell(\mathsf{E}^{\square}) = \ell(\mathsf{E}^{\bullet}) = \ell(\mathsf{E}) \ , \tag{1.18}$$

$$\mathsf{E} \textit{ en forme normale} \implies \mathsf{E} = \mathsf{E}^{\bullet} \ . \tag{1.19}$$

1.5 Multiplicités

Dans cette section nous allons définir le cadre pour l'étude des automates et des expressions rationnelles à multiplicités. Ces notions ne seront utilisées que dans les Chapitres 5 et 8. Toutes les expressions rationnelles et les automates rencontrés en dehors de ces chapitres devront être compris sous le sens booléen défini précédemment. Les définitions pour les séries formelles suivent celles données dans [8].

1.5.1 Séries formelles

Dans la suite on notera $\mathbb{K}$ un semi-anneau quelconque dont l'addition est notée $\oplus$ et la multiplication par une simple concaténation. Une fonction d'un monoïde M dans $\mathbb{K}$ peut être vue comme une somme formelle infinie de monômes dont les variables sont les éléments de M. C'est pour cela que l'on appelle *séries formelles* sur M à coefficients dans $\mathbb{K}$ les fonctions de M dans $\mathbb{K}$.

Soit s une série sur M à coefficients dans $\mathbb{K}$. On note $<s, x>$ le coefficient – ou la multiplicité – de $x \in M$ dans la série s. Le *support* de s, noté $\mathsf{supp}\,(s)$, est l'ensemble des éléments de M dont la multiplicité est différente de $0_{\mathbb{K}}$.

De manière plus précise nous utilisons par la suite des séries formelles sur le monoïde libre engendré par un alphabet. L'ensemble des séries formelles engendrées par le monoïde libre A^*, noté $\mathbb{K}\langle\langle A^* \rangle\rangle$, est un semi-anneau.

Dans ce cadre, la *série caractéristique* d'un langage L est la série s telle que $<s, u> = 1_{\mathbb{K}}$ pour les mots u de L et $<s, u> = 0_{\mathbb{K}}$ sinon.

Exemple 5 (*Ex. 1 cont.*). La série caractéristique du langage A^*bA^* est la série :

$$s = b + ab + ba + bb + aab + \dots$$

1.5.2 $\mathbb{K}$-automates et $\mathbb{K}$-représentations

Les séries formelles permettent d'imaginer des automates qui ne reconnaissent plus seulement un langage mais également un coefficient pour chaque

mot de ce langage, c'est-à-dire une série. Nous donnons une définition simple de tels automates :

Définition 7. *Soit A un alphabet. Un* $\mathbb{K}$-automate *sur A est un quintuplet $\mathcal{A} = \langle Q, A, E, I, T\rangle$ où :*
(i) *Q est un ensemble d'états ;*
(ii) *I est une fonction de Q dans $\mathbb{K}$ appelée fonction initiale. Pour $q \in Q$, si $I(q)$ est non nul alors q est initial avec pour valeur initiale $I(q)$;*
(iii) *T est une fonction de Q dans $\mathbb{K}$ appelée fonction finale.*
(iv) *E est l'ensemble des transitions, c'est-à-dire des arcs orientés entre états étiquetés par des monômes dont le support est une lettre.*

Le *comportement* d'un $\mathbb{K}$-automate $\mathcal{A}$ est la série $|\mathcal{A}|$ telle que le coefficient du mot u soit la somme des coefficients des calculs réussis du mot u dans $\mathcal{A}$. En particulier, si l'on considère les automates booléens définis auparavant comme des $\mathbb{B}$-automates, alors le comportement d'un tel automate est la série caractéristique du langage reconnu par l'automate. Les séries formelles qui sont le comportement de $\mathbb{K}$-automates finis sont appelées *séries $\mathbb{K}$-reconnaissables* .

Exemple 6 (*Ex. 1 cont.*)**.** Prenons $(\mathbb{Z}, +, \times)$ comme semi-anneau. Nous notons le monôme $1_{\mathbb{Z}}a$ par l'étiquette a. Le $\mathbb{Z}$-automate de la Figure 1.3 est tel qu'un mot u est l'étiquette d'autant de calculs que le nombre d'occurrences de b dans u. En effet, la transition du milieu, étiquetée par b peut être traversée une et une seule fois et chaque b implique un calcul différent. En outre chaque chemin a pour valeur 1. L'automate $\mathcal{A}_1$ 'compte' donc le nombre de b et réalise la série dont le support est A^*bA^* et telle que $\langle s, u\rangle = |u|_b$.

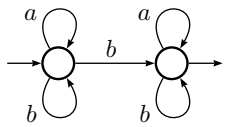

FIGURE 1.3 – Un $\mathbb{Z}$ automate, $\mathcal{A}_1$ sur l'alphabet $\{a, b\}$

Une $\mathbb{K}$-*représentation* de dimension n est un triplet (λ, μ, ν), où λ et ν sont respectivement, un vecteur ligne de dimension n et un vecteur colonne de dimension n sur $\mathbb{K}$, et μ est un morphisme de A^* dans $\mathbb{K}^{Q\times Q}$.

La série $s \in \mathbb{K}\langle\langle A^*\rangle\rangle$ réalisée par (λ, μ, ν) est définie par : pour tout $u \in A^*$, $\langle s, u\rangle = \lambda \cdot \mu(u) \cdot \nu$. Les $\mathbb{K}$-représentations sont en fait des objets similaires aux $\mathbb{K}$-automates : il est possible d'associer une représentation à chaque $\mathbb{K}$-automate : un automate dont les états sont $Q = \{q_1, \ldots, q_n\}$ est représenté par (λ, μ, ν) de dimension n si :

(i) λ_i est la valeur initiale de l'état q_i,
(ii) ν_i est la valeur finale de l'état q_i,
(iii) $(\mu(a))_{i,j}$ est le poids de la transition entre q_i et q_j étiquetée par a (si le poids est nul il n'y a pas de transition).