FONCTIONS RATIONNELLES ET ADDITION

Jean Berstel
L.I.T.P. et Institut de Programmation, Paris

Introduction

L'un des usages de la différenciation hiérarchique des automates et des langages est de mieux cerner la complexité d'un objet ou la difficulté d'une transformation. Ainsi dit-on des automates à pile qu'ils sont performants et des automates finis qu'ils sont rudimentaires (pourtant plusieurs exposés de cette école nous ont montré les hauteurs où leur étude peut mener). Les diverses espèces de transducteurs, qu'ils se promènent dans les branches ou qu'ils gardent les pieds sur terre, n'échappent pas au chercheur zélé qui classifie là ou d'autres quantifient à tour de bras.

Disposant ainsi d'une hiérarchie des fonctions rationnelles résumée dans la figure 1 (et qui sera expliquée plus bas) l'exercice que je propose est le suivant: classer l'opération d'addition dans cette hiérarchie. Ceci est vite fait si l'on se borne à l'algorithme séquentiel de l'école élémentaire, mais s'avère plus périlleux lorsque l'on emploie des méthodes non standards.

1. Fonctions rationnelles

Soient A et B deux alphabets. Une fonction partielle $\alpha : A^* \longrightarrow B^*$ est une fonction rationnelle si α est une transduction

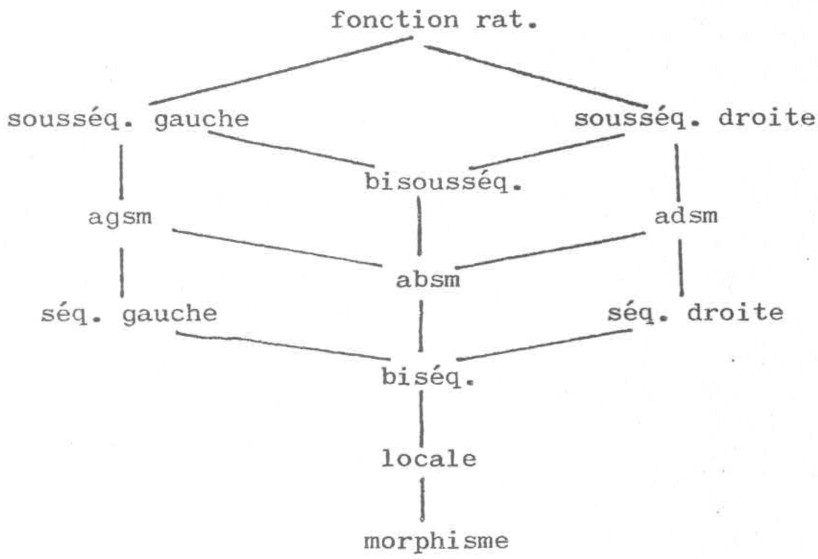


figure 1: catégories de fonctions rationnelles

rationnelle au sens d'Eilenberg[ ]. Les fonctions séquentielles gauches sont les fonctions réalisées par un transducteur séquentiel gauche. Les "agsm" (generalized sequential mappings with accepting states) sont les restrictions d'une fonction séquentielle à un langage rationnel. Une fonction sousséquentielle est le produit (de concaténation) d'une fonction séquentielle et d'une fonction rationnelle d'image finie. Pour plus de détails, voir Choffrut[ ]. L'avantage pratique de tout ce qui est (sous)séquentiel est l'aspect déterministe (i.e. l'automate sousjacent est déterministe), mais cet aspect n'est pas déterminant puisque, d'après le théorème d'Elgot et Mezei, toute fonction rationnelle peut être simulée par deux fonctions séquentielles.

La définition de fonction locale n'est pas aussi connue (elle ne figure pas dans [ ]). Soit $n>0$ un entier et A un alphabet. On définit une application

$$\theta : A^* \longrightarrow U^*$$

comme suit: d'abord θ est une bijection de $\{\varepsilon\} \cup A \cup A^2 \cup \ldots \cup A^n$ sur U .

Ensuite, si $w = a_1 a_2 \ldots a_{n+m}$ avec $a_i \in A$ est un mot de longueur $> n$, alors

$$\theta(w) = \theta(a_1 \ldots a_n)\theta(a_2 \ldots a_{n+1}) \ldots \theta(a_{m+1} \ldots a_{m+n})$$

Une <u>fonction</u> n-<u>locale</u> $\alpha : A^* \longrightarrow B^*$ est une application qui se factorise en:

$$\begin{tikzcd} A^* \arrow[rr, "\alpha"] \arrow[dr, "\theta"'] & & B^* \\ & U^* \arrow[ur, "\sigma"'] & \end{tikzcd}$$

où θ est comme ci-dessus et σ est un morphisme. Il est immédiat qu'une fonction locale est biséquentielle. La réciproque est fausse. Ces fonctions locales ont été étudiées par de nombreux auteurs qui ont examiné leur comportement sur des mots infinis dans les deux sens notamment. Comme point d'entrée à cette étude, citons Nasu[ ].

2. <u>L'addition</u>

L'écriture d'un entier dans une base se fait traditionnellement de sorte que les chiffres de plus faible poids se trouvent à droite. Si A est l'alphabet des chiffres utilisés, l'addition est une application:

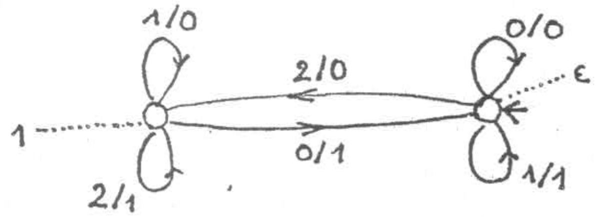
$$A^* \times A^* \longrightarrow A^*$$

$$(u,v) \longmapsto u \oplus v$$

où $u \oplus v$ désigne une écriture symbolisant la somme des entiers représentés par u et v. Une certaine manipulation - rationnelle - permet de ramener, le plus souvent, les mots u et v à la même longueur (mais ceci n'est pas toujours possible: en effet, si l'on code comme Büchi[ ] les entiers sur $A = \{1,2\}$ en associant à $w = a_n a_{n-1} \ldots a_1 a_0$ l'entier $a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_0$, la représentation d'un entier est unique; mais la construction ci-après s'applique quand même).

Le couple (u,v) se représente alors bien par un seul mot w sur un alphabet B de taille double et dont chaque lettre b représente les sommes égales de deux chiffres de A (on joue sur des propriétés bien connues des sémanticiens). Ainsi, le couple de mots binaires (sur $A=\{0,1\}$) (111101, 001101) est représenté par 112202 sur l'alphabet $B = \{0,1,2\}$. L'opération d'addition proprement dite se ramène alors simplement à réécrire le mot $w \in B^*$ comme mot de A^*. Cette démarche en deux temps a été nommée l'addition démultipliée par M. P. Schützenberger (communication personnelle).

Voici le transducteur sousséquentiel droit pour l'addition binaire:



On doit interpréter les mots attachés aux états comme suit: ce sont les mots à préfixer à la sortie de l'automate si à la fin de la lecture l'état correspondant est atteint. Ainsi l'addition binaire usuelle est-elle une fonction sousséquentielle droite. Elle n'est pas séquentielle droite, et elle n'est pas sousséquentielle gauche: on sait bien que l'on ne peut pas additionner deux nombres 'de la gauche vers la droite'.

3. L'addition molle

Il existe une addition - qui semble appartenir au folklore - qui, du point de vue adopté ici, est remarquable. Cette méthode d'additionner deux nombres ignore les reports (qui constituent évidemment le point crucial de l'addition ordinaire). En échange, la représentation des nombres est ambiguë. Exposons la méthode en base 10 (cela marche en toute base b, le cas de la base 2 est légèrement plus compliqué). Tout entier $n \geq 0$ s'écrit de manière non nécessairement unique sous la forme

$$n = n_t 10^t + n_{t-1} 10^{t-1} + \ldots + n_0$$

avec, et c'est là la différence avec l'écriture usuelle,

$$0 \leq n_i \leq 11$$

Etant donné un deuxième nombre

$$m = m_t 10^t + \dots + m_0$$

on addition n et m comme suit: on définit (p_i) et (q_i) par

$$n_i + m_i = p_{i+1}10 + q_i \qquad 0 \leq q_i \leq 9$$

et $$s_i = p_i + q_i \, .$$

Comme on a $p_i = 0$, 1 ou 2, on a $0 \leq s_i \leq 11$ et $s=n+m$ est encore écrit 'mollement'. L'intérêt 'pratique' de cette addition (que Vuillemin appelle addition molle) provient du fait que les p_i, q_i puis les s_i peuvent être évalués en parallèle, donc que l'addition molle se fait en deux "cycles".

Une inspection facile montre que l'addition molle est une fonction (2-)locale. Bien sûr, l'opération de normalisation, i.e. le retour à des digits dans l'ensemble $\{0,\dots,9\}$,nécessite l'emploi d'une fonction sous-séquentielle droite.

4. Addition en base de Fibonacci

Les nombres de Fibonacci sont définis par

$$F_0=1, \quad F_1=2, \quad F_{n+2} = F_{n+1} + F_n$$

Soit $A = \{0,1,2,\dots,k\}$, avec $k \geq 1$. A tout mot

$$w = a_m a_{m-1} \dots a_0 \qquad (a_i \in A)$$

on associe un entier $[w]$ de la façon suivante:

$$[w] = a_m F_m + a_{m-1} F_{m-1} + \dots + a_0 F_0$$

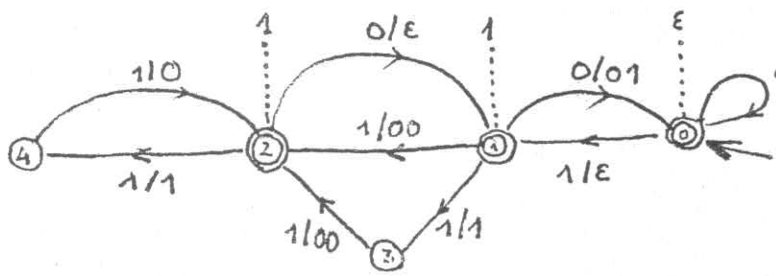
Les propriétés suivantes sont connues (voir Knuth[ ]): L'application $[\;]: A^* \longrightarrow \mathbb{N}$ est surjective même dans le cas $k=1$. Soit $K = (\{0,1\}^* \setminus 0\{0,1\}^*) \setminus \{0,1\}^*11\{0,1\}^*$. La restriction de $[\;]$ à K est une bijection de K sur $\mathbb{N}$ (K est une transversale rationnelle).

Appelons normalisation l'application de A^* dans A^* qui à w associe l'unique mot $v \in K$ vérifiant $[w]=[v]$ (C'est donc l'application $[\;][\;]^{-1} \cap K$).

Théorème: La normalisation est une fonction rationnelle.

On voit facilement que la normalisation n'est ni sousséquentielle gauche, ni droite.

Comme illustration de l'énoncé (dont Sakarovitch a tiré des conclusions sensationnelles [ ]), donnons le normalisateur pour k=1 que je dois à M. P. Schützenberger (communication personnelle):



Ce qui nous importe ici, c'est le corollaire suivant:

Corollaire: L'addition en base de Fibonacci est une opération rationnelle.

Preuve: Après normalisation, les deux mots représentant les nombres à additionner sont dans K. Démultiplions-les: le mot résultant est dans $\{0,1,2\}^*$. Il suffit alors de normaliser. □

La question de la séquentialité de l'addition se pose alors naturellement. L'opération totale, impliquant une normalisation, ne peut être séquentielle. Mais par contre, on peut réaliser la démultiplication sans normalisation par une fonction sousséquentielle gauche.

Enfin et pour finir, on doit mentionner le problème de l'addition molle en base de Fibonacci. Il est remarquable qu'une telle méthode d'addition existe. Pour la réaliser, nous devons par contre augmenter l'ambiguité de la représentation bien plus que dans le cas classique: 8 chiffres supplémentaires au lieu de 2 dans la méthode usuelle. Il n'est pas certain que ce soit la meilleure borne.

Références

[1] J. Büchi, Weak second-order arithmetic and finite automata, Z. Math. Logik Grundl. Math. 6(1960),66-92

[2] C. Choffrut, A generalization of Ginsburg and Rose's characterization of gsm mappings, in H. Maurer (ed.), Automata, Languages and Programming, Sixth Colloq., Lect. Notes Comput. Sci. 71, Springer 1979, 88-103

[3] S. Eilenberg, "Automata, Languages and Machines", Academic Press 1974

[4] D. Knuth, "The Art of Computer Programming", Vol 1, Addison-Wesley 1968

[5] M. Nasu, Indecomposable local maps of tesselation automata, Math. Systems Th. 13(1979), 81-93

[6] J. Sakarovitch, Easy multiplications, Rapport technique, Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan 1981