Suites récurrentes de nombres et automates

Géraud Sénizergues

LaBRI et Université Bordeaux 1

URL:http://dept-info.labri.u-bordeaux.fr/∼ges

1-INTRODUCTION

Récurrences linéaires sur les entiers.

Exemples:

$$U(0) = 1 \;\; U(n+1) = 2 \cdot U(n)$$

$$F(0) = 1, \;\; F(1) = 1, \;\; F(n+2) = F(n+1) + F(n)$$

1-INTRODUCTION

Récurrences polynomiales sur les entiers.

Exemples:

$$P(0) = 1 \;\; P(n+1) = (P(n))^2$$

$$L(0) = 2 \;\; FC(0) = 1 \;\; L(n{+}1) = L(n){+}1 \;\; FC(n{+}1) = L(n) \cdot FC(n).$$

II- Composition de suites

$$F_2(n) = F(F(n))$$

$F(n) : 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377 \ldots$
$F_2(n) : 1, 1, 2, 3, 8, 34, 377, \ldots$
Exercice 1: Montrer que F_2 vérifie une récurrence polynomiale.
Exercice 2: La suite FC vérifie une récurrence polynomiale. Est-elle décomposable comme une composée de deux suites définies par des récurrences linéaires ?

II- Composition de suites

Une solution de l'exercice 1.

$$\begin{pmatrix} F(n+1) \\ F(n) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

donc

$$\begin{pmatrix} F(F(n)+1)) \\ F_2(n) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{F(n)} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Posons $\begin{pmatrix} e(n) & f(n) \\ g(n) & h(n) \end{pmatrix} := \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{F(n)}$

alors

$$F_2(n) = g(n)$$

Comme

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{F(n+1)} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{F(n)} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{F(n-1)}$$

on obtient la relation:

$$\begin{pmatrix} e(n+1) & f(n+1) \\ g(n+1) & h(n+1) \end{pmatrix} = \begin{pmatrix} e(n) & f(n) \\ g(n) & h(n) \end{pmatrix} \cdot \begin{pmatrix} e(n-1) & f(n-1) \\ g(n-1) & h(n-1) \end{pmatrix}$$

ce qui fournit un systèmes de récurrences polynomiales définissant$F_2(n)$.

II- Composition de suites-Extension aux mots

Soit $TM : \mathbb{N} \to \{0,1\}^*$

$$TM(0) = 0 \;\; TM(0)' = 1$$

$$TM(n+1) = TM(n) \cdot TM'(n) \;\; TM'(n+1) = TM'(n) \cdot TM(n)$$

Récurrence caténative des entiers vers les mots.

$$\begin{array}{rcl} TM & : & 0, 01, 0110, 01101001, 0110100110010110, \ldots \\ TM' & : & 1, 10, 1001, 10010110, 1001011001101001, \ldots \end{array}$$

II- Composition de suites-Extension aux mots

Indication pour résoudre l'exercice 2. Montrer que FC est décomposable sous la forme

$$FC(n) = U(u(n))$$

où $u : \mathbb{N} \to \{a,b\}^*$ vérifie une récurrence caténative
et $U : \{a,b\}^* \to \mathbb{N}$ vérifie une récurrence linéaire

II- COMPOSITION DE SUITES-EXTENSION AUX MOTS

Une solution de l'exercice 2.

$$u(0) = b \;\; v(0) = ab \;\; u(n+1) = u(n) \cdot v(n) \;\; v(n+1) = a \cdot v(n)$$

$$U(\varepsilon) = 1 \;\; V(\varepsilon) = 0$$

$$U(a \cdot w) = U(w) + V(w) \;\; V(a \cdot w) = V(w)$$

$$U(b \cdot w) = U(w) \;\; V(b \cdot w) = U(w)$$

On vérifie que

$$u(n) = babaab \cdots ba^n b \quad v(n) = a^{n+1} b$$

$$U(u(n)) = FC(n) = (n+1)!$$

Théorème

Considérons une application $f : \mathbb{N} \to \mathbb{N}$. Les propriétés suivantes sont équivalentes:

1- f est la composée d' une suite de mots vérifiant une récurrence caténative par une suite d'entiers (indicée par les mots) vérifiant une récurrence linéaire.

2- f vérifie une récurrence polynomiale.

III-ÉGALITÉ DE SUITES RÉCURRENTES

Exercice 3: Soient deux suites $U(n), V(n)$ données par des récurrences linéaires à coefficients entiers naturels. Donner un algorithme qui décide si

$$\forall n \in \mathbb{N}, U(n) = V(n)?$$

III-ÉGALITÉ DE SUITES RÉCURRENTES

Problème 4: Soient deux suites $U(n), V(n)$ données par des récurrences polynomiales à coefficients entiers naturels.
Donner un algoritme qui décide si

$$\forall n \in \mathbb{N}, U(n) = V(n)?$$

IV-PROBLÈMES OUVERTS

1-Problème de l' égalité: Soient deux suites $U(n), V(n)$ données par des récurrences d' ordre k i.e. U, V sont des composés de $k-1$ suites d'ordre 2. Peut-on trouver un algorithme qui décide si

$$\forall n \in \mathbb{N}, U(n) = V(n)?$$

2-Problème du terme commun (Pisot, Skolem): Soient deux suites $U(n), V(n)$ données par des récurrences linéaires. Peut-on trouver un algorithme qui décide si

$$\exists n \in \mathbb{N}, U(n) = V(n)?$$

3-Problème de la presqu'égalité: Soient deux suites $U(n), V(n)$ données par des récurrences d'ordre $k \geq 2$. Peut-on trouver un algorithme qui décide si

$$\exists m \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq m \Rightarrow U(n) = V(n)?$$

4-La convolution de deux suites récurrentes ploynomiales est-elle récurrente polynomiale ?

V-REFERENCES

Quelques références sur les suites récurrentes d' entiers/mots:

Linear recurrences: (J. Berstel and C. Reutenauer 88)

P-recurrences:(R.P. Stanley 80), (M. Petkovšek, H.S. Wilf and D. Zeilberger 96)

Sequences and finite automata:(J.P. Allouche and J. Shallit 03)

L-systems:(L. Kari and G. Rozenberg and A. Salomaa 97)