Résolution d'équations diophantiennes par automates binaires

Épreuve pratique d'algorithmique et de programmation
Concours commun des Ecoles Normales Supérieures
Durée de l'épreuve : 4 heures

Juillet 2004

Ce problème est consacré à la recherche de solutions entières positives d'un système d'équations linéaires dont les coefficients sont des entiers relatifs.

Le préambule définit les données qui seront appliquées aux algorithmes du problème. Il est cependant **très fortement conseillé** de tester au préalable vos algorithmes sur des données plus petites. La partie 1 élabore des outils pour manipuler les automates. La partie 2 est consacrée au codage des solutions d'un système d'équations par un automate.

Nous appelons complexité d'un algorithme, l'ordre de grandeur du nombre d'instructions exécutées suite au lancement de l'algorithme, par exemple $O(n)$, $O(n^2)$.

Attention, il est possible que certains calculs prennent trop de temps : dans ce cas, ne vous obstinez pas.

Préambule

Posons *NUMERO* égal à votre numéro inscrit sur votre table d'examem. Soit (u_i) la suite récurrente définie par $u_{i+1} = 3321 \cdot u_i + 5701$.

Question 1 *Donnez les valeurs* **modulo** 1000 *de* u_0, u_1, u_2, u_{20}, u_{100}, u_{1000} *pour* $u_0 =$ *NUMERO.*

1 Automates binaires

Un *automate binaire* est une structure $A = \langle Q, \delta, F, q_0 \rangle$ où Q est un ensemble fini d'états, $F \subseteq Q$ est l'ensemble des états finaux, $\delta : Q \times \{0,1\} \to Q$ est la fonction de transition, et q_0 est l'état initial. Nous noterons $A[q]$ l'automate A avec q comme état initial.

Notons $\{0,1\}^*$ l'ensemble des mots sur l'alphabet $\{0,1\}$. La fonction de transition δ est étendue aux mots par :

$$\begin{array}{lcl} \delta^* : Q \times \Sigma^* & \to & Q \\ \delta^*(p, \epsilon) & = & p \\ \delta^*(p, \sigma \cdot a) & = & \delta(\delta^*(p, \sigma), a) \end{array}$$

où σ est un mot ($\sigma \in \{0,1\}^*$) et a est la lettre 0 ou 1. Un mot σ est accepté ssi $\delta^*(q_0, \sigma) \in F$. Notons $L(A)$ l'ensemble des mots acceptés par A.

Dans vos programmes, un automate $A = \langle Q, \delta, F, q_0 \rangle$ sera identifié par
– $Q = \{0, \ldots, N-1\}$,
– δ, un tableau d'entiers à deux dimensions,
– F, un tableau de booléens, et
– $q_0 = 0$.
Les tableaux représentant la fonction de transition et l'ensemble des états finaux pourront au besoin être surdimensionnés.

1.1 Calcul des automates A_N et B_N pour $N = 19$, 97 et 503

Soit N un entier. On définit l'automate $A_N = \langle \{0, \ldots, N-1\}, \delta, F, 0 \rangle$ par :

$$\begin{array}{lcl} \delta(q,a) & = & u_{2*q+a} \mod N \\ q \in F & \text{ssi} & (u_q \mod N) \leq \frac{N}{3} \end{array}$$

avec $u_0 = NUMERO$. L'automate B_N est obtenu de la même manière en prenant $u_0 = NUMERO + 11$.

Question 2 *Calculez le cardinal des ensembles F, $U = \{q \in Q | \exists a \in \{0,1\}, \delta(q,a) \in F\}$ et $V = \{q \in Q | \forall a \in \{0,1\}, \delta(q,a) \in F\}$ pour les automates A_N et B_N avec $N = 19$, 97 et 503.*

Question 3 *Combien de mots de longueurs $n = 0, 1, 2, 3, 4$ et 20 sont reconnus par les automates A_N et B_N pour $N = 19$, 97 et 503. (Indication : posons $H(q,n)$ le nombre de mots de longueur n reconnus par $A[q]$; si $q \in F$, alors $H(q,0) = 1$ sinon $H(q,0) = 0$; pour $n \geq 0$, $H(q, n+1) = H(\delta(q,0), n) + H(\delta(q,1), n)$).*

Question 4 *Nous considérons que les mots sont triés par leurs longueurs et que deux mots de même longueur sont triés par ordre lexicographique (0<1). Donnez la longueur du 10^e, 100^e, 1000^e, 10000^e mot reconnu par les automates A_N et B_N pour $N = 19$, 97 et 503.*

1.2 Opérations sur les automates

Soit $A = \langle Q, \delta, F, q_0 \rangle$. Posons $\overline{A} = \langle Q, \delta, \overline{F}, q_0 \rangle$ où $\overline{F} = Q \setminus F$. On peut monter que l'automate $\overline{A}$ reconnait exactement les mots non reconnus par A (i.e $\overline{L(A)} = L(\overline{A})$).

Question 5 *Donnez une description concise de l'algorithme d'évaluation de l'opération de complément. Donnez en la complexité en fonction de la taille de l'automate. Calculez $\overline{A_N}$, $\overline{B_N}$ pour $N = 19$, 97 et 503. Donnez le nombre de mots de longueur $n = 0, 1, 2, 3, 4$ et 20 acceptés par ces automates. Comparez vos résultats avec ceux de la question 3 ? Que remarquez-vous ?*

Soit $B = \langle Q', \delta', F', q'_0 \rangle$. Posons $A \cap B = \langle Q \times Q', \delta \otimes \delta', F \times F', (q_0, q'_0) \rangle$ avec $\delta \otimes \delta((q,q'),a) = (\delta(q,a), \delta'(q',a))$. Posons $A \cup B = \langle Q \times Q', \delta \otimes \delta', (F \times Q') \cup (Q \times F'), (q_0, q'_0) \rangle$. On peut montrer que $L(A \cap B) = L(A) \cap L(B)$ et $L(A \cup B) = L(A) \cup L(B)$.

Question 6 *Donnez une description concise des algorithmes d'évaluation des opérations d'intersection et d'union. Donnez en la complexité en fonction de la taille des automates. Calculez* $A_N \cap B_N$ *et* $A_N \cup B_N$ *pour* $N = 19$, 97 *et* 503. *Donnez le nombre de mots de longueur* $n = 0, 1, 2, 3, 4$ *et* 20 *acceptés par ces automates.*

1.3 Minimisation d'automates

Soit $A = \langle Q, \delta, F, q_0 \rangle$. Posons $\mathrm{Acc}(A) = \{q \in Q | \exists x \in \{0,1\}^*, q = \delta^*(q_0, x)\}$. Notez que le calcul de $\mathrm{Acc}(A)$ peut être simplement réalisé en initialisant $\mathrm{Acc}(A)$ à $\{q_0\}$ et en lui ajoutant tout état $\delta(q, a)$ tel que $q \in Acc(A)$ et $a \in \{0,1\}$; le calcul est terminé quand pour tout $q \in Acc(A)$ et pour tout $a \in \{0,1\}$: $\delta(q, a) \in Acc(A)$.

Question 7 *Donnez une description concise de algorithme de calcul de l'ensemble Acc. Donnez en la complexité en fonction de la taille de l'automate. Donnez le cardinal des ensembles* $Acc(A_N)$, $Acc(B_N)$, $Acc(\overline{A_N})$, $Acc(\overline{B_N})$, $Acc(A_N \cap B_N)$ *et* $Acc(A_N \cup B_N)$ *pour* $N = 19$, 97 *et* 503. *Que remarquez-vous ?*

Soit Réduit(A) l'automate A réduit aux états de $\mathrm{Acc}(A)$. Remarquer que ces deux automates reconnaissent exactement les mêmes mots.

Question 8 *Donnez une description concise de algorithme de calcul de Réduit*(A). *Donnez en la complexité en fonction de la taille de l'automate. Donnez le cardinal des ensembles* Q, F, U *et* V *(voir la question 2 pour les définitions de* U *et* V*) sur les automates Réduit*(A_N), *Réduit*(B_N), *Réduit*$(\overline{A_N})$, *Réduit*$(\overline{B_N})$, *Réduit*$(A_N \cap B_N)$ *et Réduit*$(A_N \cup B_N)$ *avec* $N = 19$, 97 *et* 503. *(Indication : les états de l'automate Réduit*(A) *doivent être renumérotés pour respecter notre choix de codage d'un automate).*

Soit L une relation d'équivalence sur Q ($L \subseteq Q \times Q$). Nous dirons que L est compatible avec l'automate A si $\forall q, q' \in Q, \forall a \in \{0,1\}$:

$$\begin{array}{lcl} L(q,q') \wedge q \in F & \Rightarrow & q' \in F \\ L(q,q') & \Rightarrow & L(\delta(q,a), \delta(q',a)) \end{array}$$

Notez que la compatibilité d'une relation L induit la définition d'un automate quotient où les états sont les classes d'équivalence de L. On peut montrer que l'automate A et son automate quotient reconnaissent exactement les mêmes mots. Quand L est la plus grand relation compatible avec l'automate, l'automate produit est appelé l'*automate minimal* de A et est noté Minimal(A).

Soit P un ensemble d'états, posons $\Delta(P) = (P \times P) \cup (\overline{P} \times \overline{P})$. L'algorithme de calcul de la plus grand relation d'équivalence compatible avec l'automate A repose sur les règles suivantes :

1. initialiser l'algorithme avec $L = \Delta(F)$,
2. soit C une classe de L, soit $a \in \{0,1\}$, posons $P = \{q \in Q | \delta(q,a) \in C\}$. Si $L \neq L \cap \Delta(P)$ alors remplacer L par $L \cap \Delta(P)$,
3. si L est compatible avec A, alors L est la relation recherchée.

Nous coderons une relation d'équivalence L par un tableau de $|Q|$ entiers : $L[q]$ est l'état le plus petit de la classe d'équivalence contenant q. On remarquera que la relation $Q \times Q$ est codée par un tableau dont toutes les valeurs sont nulles.

Question 9 *Soit P un ensemble d'états et L une relation d'équivalence sur les états. Donnez une description concise de l'algorithme de calcul de la relation $L \cap \Delta(P)$. Donnez en la complexité en fonction du nombre d'états. Posons $L_0 = \Delta(F)$, $L_{i+1} = L_i \cap \Delta(P_i)$ avec $P_i = \{q \in Q | L_i(\delta(q,0),0)\}$. Calculez les relations L_i pour $i = 0, \ldots, 6$ sur les automates A_N et B_N avec $N = 19$, 97 et 503. Donnez pour chaque relation le nombre de classes d'équivalence, ainsi que le nombre d'états équivalents à 0. Que remarquez-vous ?*

Question 10 *Donnez une description concise de l'algorithme de calcul de la plus grand relation compatible avec un automate. Donnez en la complexité en fonction de la taille de automate. Appliquer votre algorithme aux automates A_N et B_N avec $N = 19$, 97 et 503. Donnez le nombre de classes d'équivalence des relations obtenues, ainsi que le nombre d'états équivalents à 0.*

Question 11 *Donnez une description concise de l'algorithme de calcul de l'automate Minimal(A). Donnez en la complexité en fonction de la taille de automate. Calculez Minimal(A_N), Minimal(B_N), Minimal(Réduit(A_N)), Minimal(Réduit(B_N)), Réduit(Minimal(A_N)) et Réduit(Minimal(B_N)), avec $N = 19$, 97 et 503. Donnez le cardinal des ensembles Q, F, U et V (voir la question 2 pour les définitions de U et V) sur les automates calculés. Que remarquez-vous ?*

Question 12 *Quelle méthode préconisez-vous pour calculer Réduit(Minimal($\overline{A_N}$)), Réduit(Minimal($\overline{B_N}$)), Réduit(Minimal($A_N \cap B_N$)) et Réduit(Minimal($A_N \cup B_N$)) ? Appliquez votre méthode pour calculer la valeur des automates avec $N = 19$, 97 et 503. Donnez le cardinal des ensembles Q, F, U et V (voir la question 2 pour les définitions de U et V) sur les automates calculés. (Note : si pour certaines valeurs de N, le calcul prend trop de temps, alors ne vous obstinez pas.)*

2 Equations diophantiennes

Soit $\sigma = \sigma_0 \cdots \sigma_{k-1}$ un mot binaire (i.e. $\sigma \in \{0,1\}^*$). Posons

$$\text{Entier}(\sigma) = \sum_{i=0}^{k-1} \sigma_i \cdot 2^i$$

Notez que σ est une représentation en base 2 de l'entier Entier(σ). Par extension, nous définissons la fonction $\text{Entier}_n : \{0,1\}^* \to \mathbb{N}^n$ par $\text{Entier}_n(\sigma) = (x_0, \cdots, x_{n-1})$ avec

$$x_u = \sum_{i : i \cdot n + u < k} \sigma_{i \cdot n + u} \cdot 2^i$$

Par exemple, pour $\sigma = 1011000010$, $\text{Entier}(\sigma) = 269$, $\text{Entier}_2(\sigma) = (19, 2)$, $\text{Entier}_3(\sigma) = (3, 0, 5)$, $\text{Entier}_4(\sigma) = (5, 0, 1, 1)$.

Question 13 *Donnez une description concise de l'algorithme de calcul du n-uplet* $Entier_n(\sigma)$. *Donnez en la complexité en fonction de la taille du mot* σ *et* n. *Soit* σ *un mot de longueur* 20 *avec* $\sigma_i = 1$ *si* $u_i \mod 819 < 409$ *sinon* $\sigma_i = 0$. *Posons* $u_0 = NUMERO$. *Donnez la valeur de* $Entier_n(\sigma)$ *pour* $n = 1, \ldots, 5$.

Soit A un automate binaire et n un entier positif non nul. L'automate A code l'ensemble des n-uplets $\text{Entier}_n(\sigma)$ avec $\sigma \in L(A)$. Voici la définition d'un automate codant les solutions entiers positifs d'une équation linéaire.

Soit $\sum_{i=0}^{n-1} c_i \cdot x_i = d$ une équation linéaire à coefficients dans $\mathbb{Z}$. Soit $A = \langle Q, \delta, F, q_0 \rangle$ l'automate défini par :
- $Q = \{\emptyset\} \cup (\{0, \ldots, n-1\} \times \mathbb{Z})$,
- pour $a = 0, 1$, $\delta(\emptyset, a) = \emptyset$,
- pour $a = 0, 1$ et $\langle i, e \rangle \in \{0, \ldots, n-1\} \times \mathbb{Z}$

$$\delta(\langle i, e \rangle, a) = \begin{cases} \langle i+1, e + a \cdot c_i \rangle & \text{si } i < n-1 \\ \langle 0, \frac{e + a \cdot c_{n-1}}{2} \rangle & \text{si } i = n-1 \wedge e + a \cdot c_{n-1} = 0 \mod 2 \\ \emptyset & \text{sinon} \end{cases}$$

- $F = \{0, \ldots, n-1\} \times \{0\}$,
- $q_0 = \langle 0, -d \rangle$

Bien que par définition l'automate a un nombre infini d'états, $\text{Acc}(A)$ est fini. On peut montrer que

$$\text{Acc}(A) \subseteq \{\emptyset\} \cup (\{0, \ldots, n-1\} \times \{-M, \ldots, M\})$$

avec $M = 2 * (d + \sum_{i=0}^{n-1} |c_i|)$.

Soit n un entier strictement positif. Nous définissons l'équation E_k à n variables :

$$\sum_{i=0}^{n-1} c_i \cdot x_i = d$$

avec $c_i = (u_i \mod 13) - 6$, $b = (u_n \mod 13) - 6$ et $u_0 = NUMERO + 17 * k$.

Question 14 *Donnez une description concise de l'algorithme de calcul de l'automate codant les solutions d'une équation. Vous attacherez une attention particulier à la manière d'identifier les états de l'automate par un entier. Donnez en la complexité en fonction de la taille des coefficients et du nombre de variables. Pour l'équation* E_0 *avec* $n = 4, \ldots, 8$, *donnez le cardinal de* $Acc(A)$.

Question 15 *Pour l'équation* E_0 *avec* $n = 4, \ldots, 8$, *donnez la valeur de* $Entier_n$ *du* 10^e, 100^e, 1000^e, 10000^e *mot reconnu par les automates des équations.*

Question 16 *Soit* S_k *le système d'équations* $E_0, \ldots, E_k$. *Quelle méthode préconisez-vous pour calculer un automate codant les solutions du système* S_k *ayant un minimun d'états ? Appliquez votre méthode sur les systèmes* $S_2, \ldots, S_5$ *avec* $n = 4, \ldots, 8$ *et donnez le nombre d'états des automates. (Note : si pour certains systèmes, le calcul prend trop de temps, alors ne vous obstinez pas.)*

Question 17 *Comment peut-on vérifier qu'un système n'a pas de solutions ? Appliquez votre technique pour calculer la plus petite valeur telle que* S_k *n'a pas de solutions pour* $n = 4, \ldots, 8$.