Université de Nice-Sophia Antipolis — Licence Info 2
Outils formels pour l'Informatique — 2013–2014

Contrôle continu 20 janvier

Durée : 1 heure 30

Toutes les réponses doivent être justifiées. Le correcteur attachera de l'importance à la qualité de rédaction.

Une feuille manuscrite A4 autorisée - Calculatrice interdite

1 Echauffement (2 points par question)

1.1 Complexité

Si on admet qu'une instruction élémentaire est réalisée en $t_0 = 10^{-12}$ seconde, quel est approximativement le temps de traitement d'une donnée de taille 10^{12} par un algorithme de complexité $\theta(n\ log_2(n))$? On rappelle que $2^{10} = 1024 \approx 10^3$.

Le temps d'exécution sera $n\ log_2(n) t_0$ secondes soit $10^{12}\ log_2(10^{12}) 10^{-12}$ secondes soit $log_2(10^{12}) = log_2(10^{3.4}) = 4\ log_2(10^3) \approx 40$ secondes, puisque $2^{10} = 1024 \approx 10^3$ et que par conséquent $log_2(10^3) \approx 10$.

. . . . . .

1.2 Suites récurrentes

Soit la suite $(u_n)_{n \geqslant 0}$ définie par
- $u_0 = 2$
- $u_1 = 5$
- Pour $n > 1$, $u_n = 5u_{n-1} - 6u_{n-2}$

Résoudre cette récurrence, c'est à dire exprimer u_n en fonction de n

---

L'ensemble des suites $(v_n)_{n \geqslant 0}$ qui vérifient la relation, Pour $n > 1$, $v_n = 5v_{n-1} - 6v_{n-2}$ forment un espace vectoriel de dimension 2 dont on va chercher une base de suites géométriques. Si une suite géométrique x^n satisfait cette égalité, cela signifie que $x^n = 5x^{n-1} - 6x^{n-2}$, c'est à dire que $x^2 = 5x - 6$ soit encore $x^2 - 5x + 6 = 0$, et en factorisant ou en cherchant les racines, $(x-3)(x-2) = 0$. Cela implique que les suites 2^n et 3^n forment la base recherchée et que l'on peut trouver des réels α et β tels que $u_n = \alpha\ 2^n + \beta\ 3^n$. On trouve alors la valeur des réels $alpha$ et $beta$ à partir des conditions initiales $n = 0$ et $n = 1$. On obtient alors
- $\alpha + \beta = 2$
- $2\alpha + 3\beta = 5$

d'où $\alpha = \beta = 1$ et $u_n = 2^n + 3^n$.

. . . . . .

1.3 Logique

Montrer que l'expression logique $(p \rightarrow q) \vee (\neg p \rightarrow q)$ est une tautologie.

Rappelons que $(p \rightarrow q)$ est par définition $\neg p \vee q$ et donc que $(p \rightarrow q) \vee (\neg p \rightarrow q) = (\neg p \vee q) \vee (p \vee q)$ soit $\neg p \vee p \vee q = 1 \vee q$ qui est toujours vrai.

. . . . . .

1.4 Dénombrement

Combien y a t il de suites de 5 chiffres (entre 0 et 9) qui contiennent exactement 3 occurrences du même chiffre ? Expliquez votre raisonnement. Par exemple $(1, 2, 1, 1, 3)$ et $(1, 2, 1, 3, 1)$ sont deux suites différentes qui satisfont cette propriété.

On commence par sélectionner les 3 positions où les chiffres doivent être identiques, soit $\binom{5}{3} = \frac{5.4}{2} = 10$ possibilités. On choisit alors un chiffre entre 0 et 9 pour occuper ces trois positions, on a encore 10 possibilités. Enfin, il reste à remplir les deux positions restantes à l'aide des chiffres restants soient 9.9 possibilités. Au total nous avons donc $10.10.9.9 = 8100$ possibilités.

. . . . . .

1.5 Induction

On appelle palindrome sur un alphabet A tout mot qui se lit indifféremment de gauche à droite ou de droite à gauche. Ainsi "radar", "abba" et "ressasser" sont des palindromes. Donner une définition inductive de l'ensemble des palindromes sur un alphabet à deux lettres $\{a, b\}$.

---

1. Les mots ϵ, a et b sont des palindromes.
2. Si u est un palindrome alors les mots aua et bub sont aussi des palindromes.

. . . . . .

2 Problème (20 points)

Les deux parties peuvent être traitées indépendamment.

2.1 Première partie : automates finis

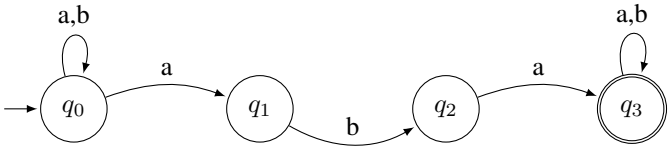
Soit L l'ensemble des mots sur l'alphabet à deux lettres $\{a, b\}$ contenant au moins une occurrence du facteur aba. On notera M son complémentaire, c'est à dire l'ensemble des mots qui n'ont aucune occurrence du facteur aba.

1. Donner une expression rationnelle qui représente le langage L

   $(a + b)$*$aba(a + b)$* est une expression rationnelle qui représente les mots contenant au moins une occurrence du facteur aba.

   . . . . . .

2. Dans un premier temps dessiner un automate non déterministe reconnaissant le langage L et préciser quel est le quintuplet définissant cet automate.



   Le quintuplet $\{A, Q, I, T, \delta\}$ est alors défini par
   – L'alphabet $A = \{a, b\}$
   – L'ensemble des états $Q = \{q_0, q_1, q_2, q_3\}$
   – L'état initial $I = q_0$
   – L'ensemble des états finaux $T = \{q_3\}$
   – La fonction de transition δ définie par

	a	b
q_0	q_0, q_1	q_0
q_1	$\times$	q_2
q_2	q_3	$\times$
q_3	q_3	q_3

. . . . . .

3. Déterminisez l'automate obtenu à la question précédente.

On déterminisme l'automate précédent en complétant la fonction de transition et en déclarant terminal tous les états qui contiennent q_3.

	a	b
q_0	q_0, q_1	q_0
q_1	$\times$	q_2
q_2	q_3	$\times$
q_3	q_3	q_3
q_0, q_1	q_0, q_1	q_0, q_2
q_0, q_2	q_0, q_1, q_3	q_0
q_0, q_1, q_3	q_0, q_1, q_3	q_0, q_2, q_3
q_0, q_2, q_3	q_0, q_1, q_3	q_0, q_3
q_0, q_3	q_0, q_1, q_3	q_0, q_3

On dessine l'automate correspondant et on peut voir que l'on peut simplifier cet automate puisque les états $q_0q_1q_3$, $q_0q_2q_3$ et q_0q_3 peuvent être remplacés par un unique état terminal avec une boucle étiquettée a, b. Cela revient à reconnaitre un mot, à partir du moment où on a lu le premier facteur aba.

. . . . . .

4. En déduire un automate déterministe reconnaissant le langage M des mots sans facteur aba.

Pour avoir le complémentaire, il suffit tout d'abord de compléter l'automate déterministe précédent en lui rajoutant le puits, et ensuite d'échanger état terminaux et états non terminaux.

. . . . . .

2.2 Deuxième partie : récurrence

On considère l'application σ définie par
– $\sigma(a) = aba$
– $\sigma(b) = bb$
qui s'étend naturellement sur $\{a, b\}^*$ en définissant l'image d'un mot par σ par la concaténation des images des lettres qui le composent, $\sigma(w_1.w_2.\ldots.w_n) = \sigma(w_1).\sigma(w_2).\ldots.\sigma(w_n)$. Ainsi $\sigma(aaba) = \sigma(a).\sigma(a).\sigma(b).\sigma(a) = abaababbaba$.

On considère alors la suite de mots $(u_n)_{n\geqslant 0}$ définie par
– $u_0 = a$
– $u_n = \sigma(u_{n-1})$ pour $n > 0$

5. Calculer u_n pour $n = 1, 2, 3$

On a $u_1 = aba$, $u_2 = ababbaba$ et $u_3 = ababbababbbbababbaba$.

. . . . . .

6. Montrer que pour tout $n > 0$ on a $u_n = u_{n-1}\, b^{2^{n-1}}\, u_{n-1}$

On démontre cette propriété par récurrence sur n :
– Initialisation. $u_1 = aba = u_0b^1u_0$ et la propriété est donc vérifiée à l'ordre 1.

– Recurrence. Supposons que la propriété soit vérifiée à l'ordre n, soit $u_n = u_{n-1}\, b^{2^{n-1}}\, u_{n-1}$, et calculons u_{n+1}. Remarquons tout d'abord que pour tout k, $\sigma(b^k) = b^{2k}$.
On a

$$\begin{aligned} u_{n+1} &= \sigma(u_n) \\ &= \sigma(u_{n-1}\, b^{2^{n-1}}\, u_{n-1}) \\ &= \sigma(u_{n-1})\, \sigma(b^{2^{n-1}})\, \sigma(u_{n-1}) \\ &= u_n\, b^{2.2^{n-1}} u_n \\ &= u_n\, b^{2^n}\, u_n \end{aligned}$$

D'où le résultat.

. . . . . .

7. En déduire une récurrence sur la longueur $L_n = |u_n|$ des mots u_n

---

On déduit immédiatement de $u_{n+1} = u_n\, b^{2^n}\, u_n$ que pour $n \geqslant 0$, on a $L_{n+1} = 2\, L_n + 2^n$

. . . . . .

8. Montrer que pour $n > 0$, on a $L_n = (n+2)2^{n-1}$

La démonstration se fait par récurrence.
– Initialisation. $L_1 = 3 = 3\, 2^{1-1}$ et la propriété est donc vérifiée à l'ordre 1.
– Recurrence. Supposons que la propriété soit vérifiée à l'ordre n, soit $L_n = (n+2)2^{n-1}$, et calculons L_{n+1}. On a

$$\begin{aligned} L_{n+1} &= 2\, L_n + 2^n \\ &= 2\, (n+2)2^{n-1} + 2^n \\ &= (n+2)2^n + 2^n \\ &= (n+3)2^n \\ &= ((n+1)+2)2^{(n+1)-1} \end{aligned}$$

D'où le résultat.

. . . . . .

9. On note $a_n = |u_n|_a$ et $b_n = |u_n|_b$ respectivement le nombre de lettres a et de lettres b dans le mot u_n. Exprimer a_n et b_n en fonction de n.

On remarque que l'application σ ne fait que doubler à chaque étape le nombre de a et donc que $a_n = 2^n$. On en déduit alors que $b_n = L_n - a_n = (n+2)2^{n-1} - 2^n = n\, 2^{n-1}$

. . . . . .