

Examen

Documents interdits

Durée: 2h

18 décembre 2013

Note importante : répondre aux questions suivantes de manière la plus lisible possible, en particulier, dans la question 6 où il vous est demandé d'écrire du code en langage C. Les parties illisibles ne seront pas lues !

1 Questions générales

Q.1.1. Montrer que tout langage fini est un langage régulier

Q.1.2. Montrer que tout langage régulier est un langage hors-contexte

Q.1.3. Montrer que la concaténation de deux langages hors-contextes est un langage hors-contexte

Q.1.4. Montrer que l'union de langages hors-contextes est un langage hors-contexte

2 Minimisation

Minimiser l'automate suivant :

	a	b	
→	1	3	5
	2	2	5
←	3	2	7
←	4	6	7
	5	4	1
	6	2	1
←	7	5	6

3 Calcul de résiduels

Calculer les résiduels suivants en détaillant les différentes étapes :

Q.3.1. $b^*a^*(a+b)^*/b$

Q.3.2. $(aba+abb)^*/a$

Q.3.3. $((ab)^* + b)^*/b$

Q.3.4. $a^*b^*a^*/aa$

4 Automate des résiduels

Q.4.1. Calculer tous les résiduels du langage décrit par l'expression régulière $E = ((a + b)(a + b)(a + b))^*$.

Q.4.2. Construire l'automate minimal reconnaissant le langage dénoté par E en suivant la méthode des résiduels.

5 Automates à pile et grammaires hors-contexte

Q.5.1. Construire un automate à pile A reconnaissant le langage $L = \{m \in \{a, b\}^* \mid |m|_a \neq |m|_b\}$.

Q.5.2. Ecrire une grammaire hors-contexte qui reconnaît L .

6 Question d'implémentation

Écrire la fonction `fsm *fsm_complement(fsm *A)` qui prend en entrée un automate déterministe A reconnaissant le langage L et retourne un automate reconnaissant le langage \bar{L} complémentaire de L : $\bar{L} = \Sigma^* - L$.

L'algorithme pour générer l'automate du complémentaire est le suivant :

- Ajouter un état puits p à A .
- Pour chaque état e de A
 - Pour chaque symbole s de Σ , modifier la fonction de transition δ de A de sorte que si $\delta(e, s) = \emptyset$ alors $\delta(e, s) = p$
 - Si e est un état d'acceptation, modifiez le pour qu'il ne le soit plus
 - Sinon, modifiez le pour qu'il le soit

Pour rappel, voici les structures utilisées en TP ainsi que quelques fonctions qui pourraient vous être utiles.

```
typedef struct { fsm_state *data; state_list *next; } state_list;
typedef struct { fsm_trans *data; trans_list *next; } trans_list;
typedef struct { int n; trans_list *ltrans; int is_accept; } fsm_state;
typedef struct { int sym; fsm_state *orig; fsm_state *dest; } fsm_trans;
typedef struct { state_list *lstates; fsm_state *init; } fsm;
fsm_trans *fsm_add_trans(fsm_state *orig, int sym, fsm_state *dest);
fsm_state *fsm_add_state(fsm *a);
int fsm_nb_sym(fsm *a);
fsm *fsm_copy(fsm *a);
```