Examen

Poly, notes de cours et transparents du cours autorisés *Durée : 2h*

6 janvier 2010

Note Importante : il vous est demandé, dans certaines questions de cet examen, d'écrire du code en langage C. Veillez à écrire votre code de manière la plus lisible possible. Le code illisible ne sera pas lu !

1 Représentation graphique d'un automate (14 pts)

Un automate fini peut être représenté dans un fichier texte en représentant sur chaque ligne une transition ou un état d'acceptation de ce dernier. Une transition est représentée par un état d'origine, un état de destination et un symbole. Les états et les symboles sont matérialisés par des entiers. L'état initial de l'automate est l'état d'origine de la première transition. Vous trouverez dans la partie gauche de la figue 1 un automate représenté sous forme matricielle et au centre de la même figure, le même automate au format textuel. On pourra remarquer que les transitions et les états d'acceptation peuvent apparaître dans n'importe quel ordre dans la représentation textuelle.

		0	1
→	0	1	2
	1		2
←	2		

```
0 1 0
2
1 2 1
0 2 1
```

```
<table>
<tr><td></td><td></td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td></td><td>2</td></tr>
<tr><td>2</td><td>2</td><td></td><td></td></tr>
</table>
```

Fig. 1 – Trois représentations d'un automate

Q.1.1. Ecrire une grammaire hors-contexte non ambiguë G qui permet de générer le langage des automates représentés sous forme textuelle. On considèrera que le retour chariot est représenté par le symbole □.

Q.1.2. Dessiner l'arbre de dérivation que G associe à l'automate de la figure 1.

Q.1.3. Ecrire un analyseur syntaxique en C pour G.

Un automate peut aussi être représenté en HTML sous la forme d'un tableau, comme représenté dans la partie droite de la figure 1[1].

[1]On rappelle que la description d'un tableau en HTML commence par la balise ouvrante `<table>` et se termine par la balise fermante `</table>`. Le tableau lui-même est composé d'une

La représentation HTML est très proche de la représentation matricielle, à ceci près que chaque ligne du tableau commence par décrire le statut de l'état décrit sous la forme d'un entier : 1 pour un état initial 2 pour un état d'acceptation 3 pour un état initial d'acceptation et 0 pour les autres.

On souhaite écrire un compilateur qui prend en entrée une représentation textuelle d'un automate déterministe A et produit la représentation matricielle de A en HTML. Pour cela on utilise la représentation intermédiaire suivante :

```
typedef struct a{
  int transitions[MAX_ETATS][MAX_SYMBOLES];
  int etats[MAX_ETATS]; /* indique le statut des états (initial, d'acceptation ...) */
} automate;
```

Q.1.4. Modifier l'analyseur écrit ci-dessus, de manière à ce qu'il construise une représentation intermédiaire de l'automate analysé.

Q.1.5. Ecrire la fonction `genere_html(automate *a)` qui prend en argument la représentation intermédiaire d'un automate et produit la représentation HTML de ce dernier.

On désire modifier notre compilateur de manière à ce qu'il puisse traiter les automates non déterministes.

Q.1.6. Modifier la représentation intermédiaire, la représentation HTML, l'analyseur et la fonction `genere_html` de manière à pouvoir traiter les automates non déterministes.

2 Compilation (6 pts)

Soit le programme suivant p écrit en langage L :

```
main()
entier $a;
{
$a = lire();
si $a egal 0 alors
ecrire(0);
sinon
ecrire(1);
}.
```

Q.2.1. Dessiner l'arbre abstrait que votre compilateur associe au programme p.

Q.2.2. Ecrire la séquence d'instructions pour la machine M produite à l'issue de la compilation du programme p. On suppose que la fonction `main` se trouve à l'adresse 5.

---

suite de lignes. Chaque ligne débute par la balise ouvrante `<tr>` et se termine par la balise fermante `</tr>`. Une ligne est composée d'une suite de cellules, chaque cellule commence par la balise `<td>` et se termine par la balise fermante `</td>`. Une cellule peut contenir toute sorte de choses, dans notre cas, nous considèrerons qu'elles ne peuvent contenir qu'un entier.