

INF123 - Examen Année 2014-2015

Durée : 2h.

Tout document interdit à l'exception du mémo shell non annoté. Calculatrices interdites.
Pour chaque question, une partie des points peut tenir compte de la présentation.
Le barème est indicatif.

Toute réponse même incomplète sera valorisée à partir du moment où elle réalise au moins une partie de ce qui est demandé.

Les questions sont relativement indépendantes, et dans tous les cas vous pouvez utiliser les fonctions demandées dans les questions précédentes même si vous n'avez pas réussi à les écrire.

Indications générales

Pour tous les exercices de ce sujet, vous pourrez utiliser les fonctions suivantes, que l'on trouve dans la librairie `string.h` :

- `int strlen(char *s)` renvoie la longueur de la chaîne donnée en argument.
- `int strcmp(char *s1, char *s2)` renvoie 0 si les chaînes données en argument sont identiques, et un nombre non nul si elles sont différentes.
- `char *strcpy(char *dest, char *src)` copie la chaîne `src` dans la chaîne `dest`, et ce jusqu'au marqueur de fin de chaîne. (La valeur renvoyée est un pointeur vers `dest`, mais il vous est déconseillé de l'utiliser.)

1 Ensembles de chaînes (5 points)

On se donne pour cet exercice le type suivant, déjà utilisé en TP, qui permet de représenter un ensemble de chaînes de caractères (autrement dit chaque chaîne de caractères ne pourra y être présente qu'au plus une fois, et l'ordre des chaînes dans le tableau n'a pas d'importance).

```
#define NB_MAX_CHAINES 128
#define TAILLE_MAX_CHAINES 128

typedef struct {
    int cardinal;
    char tab[NB_MAX_CHAINES][TAILLE_MAX_CHAINES];
} ens_chaines;
```

On rappelle que le tableau `tab` n'est pas rempli intégralement : seules les `cardinal` premières lignes sont utilisées, et les chaînes de l'ensemble sont donc `tab[0]`, `tab[1]`, ... `tab[cardinal-1]`.

Question 1 (0,5 point) : Représenter une variable de type `ens_chaines` contenant les chaînes "INF123", "Exam" et "Super" par un schéma.

Il vous est ensuite demandé de programmer les fonctions suivantes à propos d'un ou plusieurs de ces ensembles.

1.1 Plus longue chaîne

Question 2 (1 point) : Donner le code de la fonction

```
int longueur_max(ens_chaines *c);
```

qui renvoie la longueur de la plus longue chaîne de l'ensemble `c` (elle renverra 0 si cet ensemble est vide).

1.2 Chaînes sélectionnées

Question 3 (1,5 points) : Donner le code de la fonction

```
void chaines_plus_courtes_que(ens_chaines *c, int n, ens_chaines *result);
```

qui construit dans **result** l'ensemble des chaînes présentes dans **c** et de longueur inférieure à **n**.

1.3 Intersection

Question 4 (2 points) : Donner le code de la fonction

```
void chaines_communes(ens_chaines *c1, ens_chaines *c2, ens_chaines *result);
```

qui construit dans **result** l'ensemble des chaînes présentes à la fois dans **c1** et dans **c2**. Attention, les chaînes de **c1** et de **c2** ne sont pas nécessairement rangées dans le même ordre.

2 Automates (7 points)

Dans cette partie, on s'intéresse à la reconnaissance des nombres entiers naturels, codés en binaire sans 0 inutile.

Par exemple, l'entier 4 peut être représenté par 100, 0100, 00100 etc. On cherche à reconnaître uniquement la première représentation et à refuser les autres.

2.1 Nombres entiers

Question 1 (0,5 point) : Dire, parmi les représentations binaires suivantes, lesquelles sont acceptées ou rejetées et pourquoi.

- 0
- 10
- 01

Question 2 (1,5 points) : Donner un automate qui reconnaît tous les entiers en binaire sans 0 inutile. Cet automate n'a pas de sortie associée aux transitions.

Préciser quel est l'état initial, quels sont les états accepteurs, et représenter les transitions sous forme graphique.

2.2 Nombres pairs

On s'intéresse maintenant plus spécifiquement à l'écriture binaire des nombres pairs supérieurs ou égaux à 4.

Question 3 (1 point) : Donner 2 exemples de nombres de cette forme, et 2 exemples de nombres qui n'en sont pas.

Quelles sont les caractéristiques de l'écriture binaire de ces nombres ?

Question 4 (2 points) : Construire un second automate qui reconnaît les nombres pairs supérieurs ou égaux à 4, toujours en rejetant les représentations qui comportent des zéros inutiles.

Cet automate n'a pas non plus de sortie.

2.3 Transformation d'un mot binaire

Question 5 (2 points) : Construire un automate dont l'alphabet d'entrée est $\Sigma = \{0, 1\}$, l'alphabet de sortie est $\{a, b\}$ et dont le comportement est le suivant :

- pour chaque caractère lu en entrée on produit exactement un caractère en sortie
- la sortie est **b** chaque fois qu'il y a en entrée un 1 précédés de deux 0 ; sinon la sortie est **a**.

Il n'y a pas d'état final.

Exemples

| entrée | sortie |
|--------|--------|
| 01 | aa |
| 001 | aab |
| 01001 | aaaab |

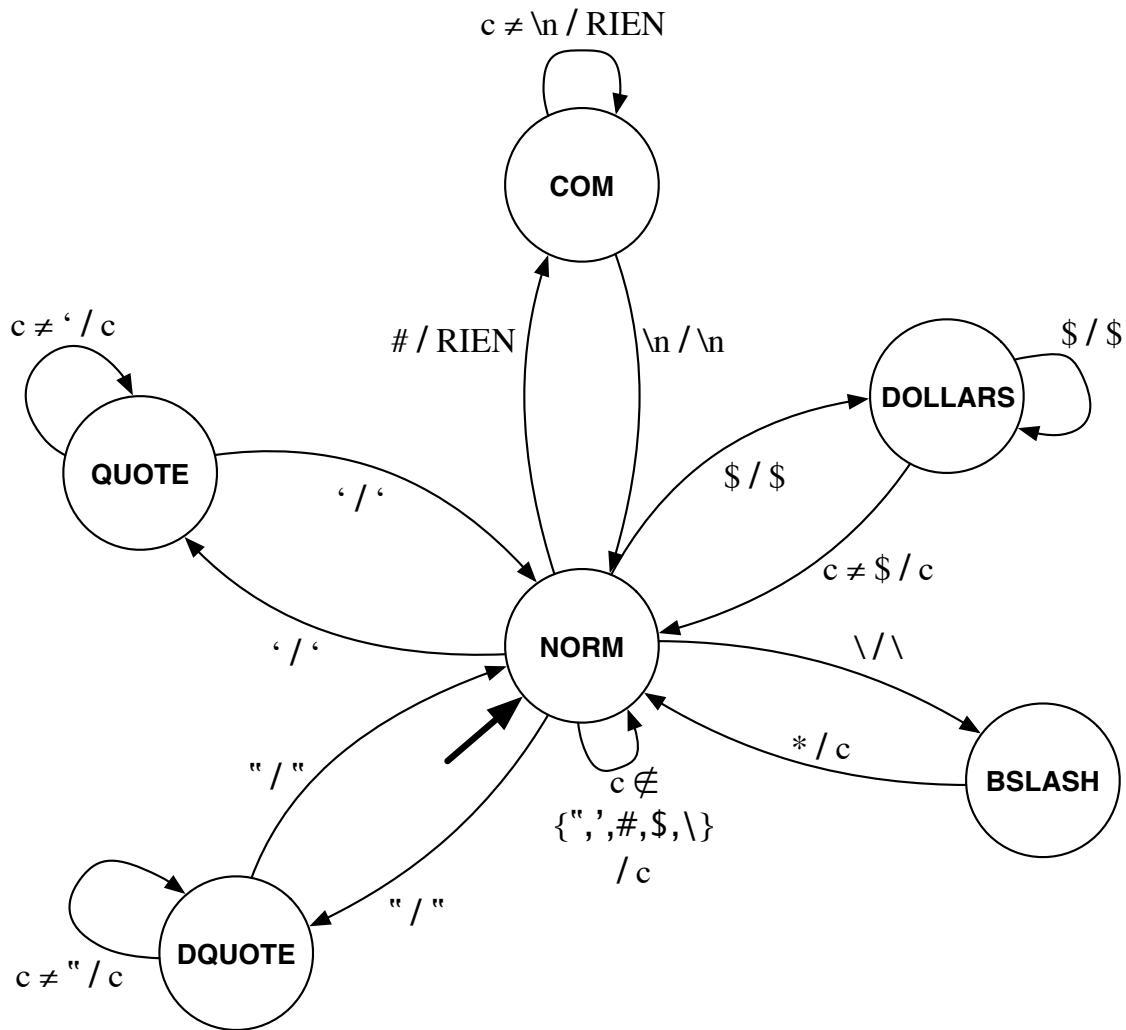


FIGURE 1 – Automate pour la suppression des commentaires dans un script SHELL.

3 Suppression des commentaires dans un script SHELL (8 points)

Dans la figure 1, nous proposons un automate qui permet de supprimer les commentaires dans un script SHELL. Nous avons adopté les conventions suivantes :

- c désigne le caractère lu.
- $/$ sépare l’entrée de la sortie
- RIEN désigne la sortie vide.
- $*$ désigne un caractère quelconque.
- $c \notin \{", ', \#, \$, \backslash\}$ signifie que c est un caractère différent des caractères spéciaux $", ', \#, \$$ et \backslash .

Cet automate permet non seulement de supprimer les portions du script comprises entre $\#$ et la fin d’une ligne, mais aussi de gérer des situations plus complexes, par exemple :

- les $\#$ entre guillemets simples ou doubles
- l’utilisation de $\#\#$

Cela reste un automate simplifié par rapport à celui qui serait utilisé dans un “vrai” interpréteur, par exemple la ligne `\#!/bin/bash` n’est pas traitée correctement.

Quoiqu’il en soit, l’objectif de cet exercice est uniquement de programmer la modélisation par automate qui vous est donnée, et il n’y a pas besoin d’en comprendre les détails pour répondre aux questions.

3.1 Analyse de l'automate

Question 1 (1 point) :

- Quel est l'état initial de cet automate ?
- Quel est l'alphabet d'entrée de cet automate ?
- Quel est l'alphabet de sortie de cet automate ?
- Parmi les caractères suivants :

", ', #, \$ et \

lesquels doivent être échappés par un backslash en langage C ?

3.2 Implémentation de l'automate

Dans les questions suivantes, vous devrez obligatoirement programmer les fonctions de l'automate **en version non-tabulée**, c'est-à-dire, **en utilisant des conditions** (if ou switch).

Vous utiliserez aussi les macros suivantes :

```
#define NORM 0
#define QUOTE 1
#define DOLLARS 2
#define COM 3
#define BSLASH 4
#define DQUOTE 5

#define RIEN 0

typedef int etat;
```

Question 2 (2 points) :

Donner le code C de la fonction

```
char sortie(etat etatcourant, char entree);
```

qui donne les sorties de l'automate donné en figure 1.

Question 3 (2 points) :

Donner le code C de la fonction

```
etat transition(etat etatcourant, char entree);
```

qui traduit la fonction de transition de l'automate donné en figure 1.

Question 4 (3 points) :

En utilisant les fonctions **sortie** et **transition**, donnez le code C de la fonction **main** d'un programme qui supprime les commentaires dans un script SHELL.

- Ce programme utilise au plus 2 arguments de la ligne de commande. Il a le comportement suivant :
- S'il y a plus de deux arguments, il affiche un message d'erreur.
 - S'il y a deux arguments, ces deux arguments sont des noms de fichiers SHELL. Il écrit alors le contenu du premier dans le deuxième **sans les commentaires SHELL**.
 - S'il y a un seul argument, alors c'est le nom d'un fichier SHELL et il affiche (sur la sortie standard) le contenu du fichier **sans les commentaires SHELL**.
 - S'il n'y a pas d'argument, alors il lit l'entrée standard et affiche ce qu'il lit **sauf les commentaires SHELL**.

Il vous est aussi demandé d'afficher les messages d'erreur adéquats lorsqu'une ouverture de fichier échoue.