

TP d'introduction à Lex

Exercice 0 : Nombres entiers et nombres réels

Ecrire et tester sur plusieurs exemples le programme lex vu en cours.

Exercice 1 : Commentaires dans les scripts

Dans de nombreux langages de script (et dans la plupart de fichiers de configuration), un commentaire commence par le symbole #. Écrire un programme LEX qui affiche tous les commentaires du fichier lu sur l'entrée standard. Par exemple :

```
> cat script.sh
#!/bin/bash

# Initialisation
i=0
# Boucle de 0 a 100
while [ $i -lt 100 ]
do
    i=$((i+1)) # Increment
    echo $i    # Affichage
> lex commentaire-script.l
> gcc lex.yy.c -o commentaire-script -ll
> ./commentaire-script < script.sh
commentaire: "#!/bin/bash"
commentaire: "# Initialisation"
commentaire: "# Boucle de 0 a 100"
commentaire: "# Increment"
commentaire: "# Affichage"
>
```

Exercice 2 : Décodage

Considérons le codage ϕ qui transforme des mots sur l'alphabet $\{a, b, c\}$ en des mots sur l'alphabet $\{0, 1\}$ défini de la façon suivante :

$$\begin{aligned}\phi(a) &= 0 \\ \phi(b) &= 01 \\ \phi(c) &= 11.\end{aligned}$$

Par exemple, $\phi(abca) = 0001110$.

Écrire un programme LEX qui décode un mot lu sur l'entrée standard. En d'autres termes, le programme lit sur l'entrée standard un mot u sur l'alphabet $\{0, 1\}$ et écrit sur la sortie standard le mot v sur l'alphabet $\{a, b, c\}$ tel que $\phi(v) = u$.

Exercice 3 : Un clone de la commande unix `wc`

La commande unix `wc` (*word count*) affiche le décompte de lignes, de mots et d'octets pour chaque fichier en entrée.

```
> wc /etc/passwd
 31   44 1380 /etc/passwd
>
```

Nous souhaitons écrire un clone simplifié de la commande `wc`. Écrire un programme LEX qui compte et affiche le nombre de caractères, le nombre de mots et le nombre de lignes dans le fichier lu sur l'entrée standard :

```
> lex counter.l
> gcc lex.yy.c -o counter -ll
> ./counter
a bb ccc
d ee fff
g hh iii
```

taper Ctrl-D

```
lines=3 words=9 chars=27
>
> ./counter < counter.l
l=29 w=50 c=386
>
```