TD n°2, I45 – Projet
L2 Sciences et Technologies
Université du Sud Toulon-Var
Année universitaire 2005 – 2006

Emmanuel Bruno

Le module d'analyse lexicale

Objectif

L'analyseur lexical doit être capable de lire dans un fichier texte et de trouver le prochain « symbole », de mémoriser sa valeur et de lui associer une catégorie. Par exemple, à partir de l'instruction `12+X = 3`, la fonction d'analyse lexicale doit reconnaître successivement :

Symbole	Valeur
<s_entier>	« 12 »
<s_plus>	« + »
<s_identificateur>	« X »
<s_egal>	« = »
<s_entier>	« 3 »

Exercice 1

Sachant que les symboles des opérateurs sont ceux indiqués dans le tableau ci-après, écrire la liste des couples `(symbole, valeur)` retournée par l'analyseur lexical pour l'expression : `(31+A*BC)/total`.

Symbole	Opérateur
<s_plus>	« + »
<s_moins>	« - »
<s_etoile>	« * »
<s_slash>	« / »
<s_parent_g>	« (»
<s_parent_d>	«) »
<s_egal>	« = »
<s_sup>	« > »
<s_sup_egal>	« >= »
<s_inf>	« < »
<s_inf_egal>	« <= »
<s_diff>	« <> »

Exercice 2 (expressions régulières)

Ecrire l'expression régulière permettant de représenter un nombre entier. Idem pour un identificateur.

Exercice 3 (lecture d'un nombre)

Soient les variables globales :

- `lex_val` qui contient la chaîne de caractères correspondant au symbole lu,
- `lex_sym` qui contient le type du symbole lu (ici, `s_entier` ou `s_reel`),
- `lex_ch` qui contient le dernier caractère lu.

Ecrire un algorithme qui lit un nombre dans un fichier et met à jour les variables `lex_val` et `lex_sym` en fonction du nombre qui est lu. On supposera que le fichier est déjà ouvert et que le premier caractère est déjà dans la variable `lex_ch`.

Identificateurs et mots réservés

Outre les nombres, l'analyseur doit pouvoir extraire les identificateurs et vérifier si la chaîne lue est un nom de variable ou un mot réservé. Pour cela, on dispose d'un tableau de `nb_mots_cles` enregistrements contenant la chaîne associée à chaque mot-clé (le champ `val` des enregistrements) et le symbole correspondant (le champ `sym`) :

```
tableau table_mots_cles =
      [(val=« Algorithme », sym=s_algorithme),
      (val=« si », sym=s_si),
      …]
```

Exercice 4 (lecture d'un identificateur)

On suppose toujours que le fichier à lire est déjà ouvert et que le premier caractère est déjà dans la variable globale `lex_ch`. Pour lire un identificateur, on procède

en deux étapes :

- sur le même principe que la recherche d'un nombre, écrire un algorithme qui recherche une chaîne de caractères et met à jour les variables globales `lex_val` et `lex_sym`. Le symbole pour un identificateur est `s_ident`.
- écrire un algorithme qui vérifie si la chaîne lue (dans `lex_val`) correspond au champ `val` d'un enregistrement du tableau `table_mots_cles` ; si c'est le cas, la variable globale `lex_sym` est mise à jour avec le symbole correspondant.

Exercice 5 (procédure du lecture du prochain symbole)

Nous allons maintenant définir l'algorithme de la procédure principale d'analyse lexicale. On suppose toujours que le fichier est déjà ouvert et que le premier caractère est déjà dans la variable `lex_ch`. Quels sont les cas qui peuvent se présenter ? Ecrire l'algorithme.

Exercice 6 (lecture d'un nombre et stockage de sa valeur)

Proposer une alternative à la solution de l'exercice 3 permettant de mettre dans la variable `lex_num`, de type réel, la valeur numérique du nombre lu.

Exercice 7 (addition de nombres entiers)

Ecrire un algorithme permettant de lire dans un fichier et de calculer, quand `n` est inconnu :

$$entier_1 + entier_2 + entier_3 + \ldots + entier_n$$

TD n°2, I45 – Projet
L2 Sciences et Technologies
Université du Sud Toulon-Var
Année universitaire 2005 – 2006

Emmanuel Bruno

Le module d'analyse lexicale : correction

Exercice 1

Symbole	Valeur
<s_parent_g>	« (»
<s_entier>	« 31 »
<s_plus>	« + »
<s_identificateur>	« A »
<s_etoile>	« * »
<s_identificateur>	« BC »
<s_parent_d>	«) »
<s_slash>	« / »
<s_identificateur>	« total »

Exercice 2 (expressions régulières)

- *entier :* [0-9][0-9]*
- *identificateur :* [a-zA-Z][0-9a-zA-Z_]*

Exercice 3 (lecture d'un nombre)

```
lex_val=«  »
répéter
     lex_val=concaténation de lex_val et lex_ch
     lex_ch=lire un caractère dans un fichier
jusqu'à ce que ((lex_ch<'0') ou (lex_ch>'9'))
lex_sym=s_entier
si lex_ch='.' alors
     lex_sym=s_reel
     repeter
          lex_val=concaténation de lex_val et lex_ch
          lex_ch=lire un caractère dans un fichier
     jusqu'à ce que ((lex_ch<'0') ou (lex_ch>'9'))
fin si
```

Exercice 4 (lecture d'un identificateur)

```
lex_val=«  »
répéter
     lex_val=concaténation de lex_val et lex_ch
     lex_ch=lire un caractère dans le fichier
jusqu'à ce que (((lex_ch<'a') ou (lex_ch>'z)
          et ((lex_ch<'A') ou (lex_ch>'Z'))
          et ((lex_ch<'0') ou (lex_ch>'9'))
          et (lex_ch<>'_'))
lex_sym=s_ident
k=0
tant que ((k<nb_mots_cles)
          et (lex_val<>table_mots_cles[k].val)) faire
     incrémenter k
fin tq
si (k<nb_mots_cles) alors
     lex_sym=table_mots_cles[k].sym
fin si
```

Exercice 5 (procédure de lecture du prochain symbole)

```
au cas où lex_ch vaut :
     '*' alors
          lex_val='*'
          lex_sym=s_etoile
          lex_ch=lire un caractère dans le fichier
     …
     '<' alors
          lex_ch=lire un caractère dans le fichier
          si lex_ch='=' alors
               lex_sym='s_inf_egal
          sinon
               si lex_ch='>' alors
```

```
                        lex_sym=s_diff
                  sinon
                        lex_sym=s_inf
                  fin si
            fin si
      Un chiffre alors
            algorithme lire_un_nombre
      Une lettre alors
            algorithme lire_un_identificateur
      Autre chose alors
            lex_sym=s_vide
fin cas
```

Exercice 6 (lecture d'un nombre et stockage de sa valeur)

```
// puissance de 10 indiquant la position de la
décimale
rang=1
// valeur entière de la partie décimale du nombre
decimale=0
lex_num=0
lex_sym=s_entier
répéter
      lex_num=lex_num*10+(ascii(lex_ch)-ascii('0'))
      lex_ch=lire un caractère dans le fichier
jusqu'à ce que ((lex_ch<'0') ou (lex_ch>'9'))
si lex_ch='.' alors
      lex_sym=s_reel
      lex_ch=lire un caractère dans le fichier
      répéter
            decimale=decimale*10+
                        (ascii(lex_ch)-ascii('0'))
            rang=rang*10
            lex_ch=lire un caractère dans le fichier
      jusqu'à ce que ((lex_ch<'0') ou (lex_ch>'9'))
      lex_num=lex_num+(decimale/rang)
fin si
```

Exercice 7 (addition de nombres entiers)

```
total=0
répéter
      lex_num=0
      répéter
            lex_num=lex_num*10+
                        (ascii(lex_ch)-ascii('0')
            lex_ch=lire un caractère dans le fichier
      jusqu'à ce que ((lex_ch<'0') ou (lex_ch>'9'))
      si lex_ch='+'
            total=total+lex_num
            lex_ch=lire un caractère dans le fichier
      fin si
jusqu'à ce que ((lex_ch<'0') ou (lex_ch>'9'))
```