

## 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 »

### Analyse lexicale du langage Algo

#### 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 de 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 :

$$\text{entier}_1 + \text{entier}_2 + \text{entier}_3 + \dots + \text{entier}_n$$