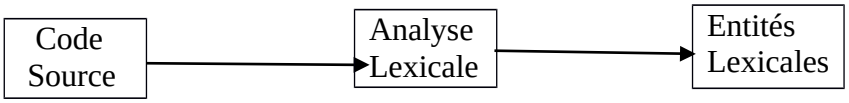
Chapitre I :

Analyse Lexicale



L'analyse lexicale consiste à partir d'un programme qui est une suite de caractères séparés ou non par des blancs à :

- Spécifier les différentes entités (mots) du langage, on parlera d'entités lexicales. Parmi ces entités, on reconnaît :
  . Les identificateurs
  . Les mots clés (réservés)
  . Les constantes
  . Les séparateurs
- Eliminer les blancs, s'ils existent, ainsi que les commentaires.
- Coder les différentes entités lexicales.
- Construire la table des informations (table des symboles).

Remarques :

- Les mots clés et les séparateurs représentent une donnée du programme à compiler. Les identificateurs et les constantes sont donnés par l'utilisateur.
- Dans certains langages, les mots clés ne sont pas réservés, ils peuvent être utilisés comme identificateurs.
- Certains langages n'acceptent pas que les mots clés soient utilisés en tant qu'identificateurs. Ils sont alors directement insérés dans la table des symboles.

I.1 Implémentation d'un analyseur lexical

Les entités lexicales sont décrites à l'aide de grammaires régulières, qui sont reconnues par des automates simple déterministes à états finis.

I.1.1 Automate :

Soit A=< T, N, S, F, I >, où :

T : ensemble des terminaux.

N : ensemble des états de l'automate.

S : état initial de l'automate.

F : ensemble des états finaux.

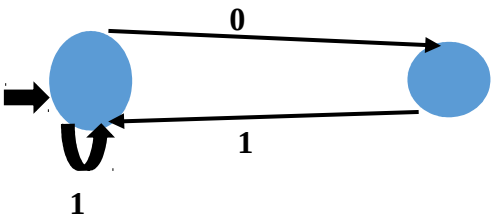
I : ensemble des transitions.

On appelle automate d'états finis, un système qui fait correspondre à une entité (chaine € T*), une réponse qui peut prendre l'une des deux valeurs, vrai ou faux.

I(Si, ai) = Sj Si, Sj ∈ N et ai ∈ T

I(Si, ε¿ = Si ∀ Si ∈ *N*

Exemple :



I.1.2 Représentation d'un automate en mémoire :

- On représente un automate par une matrice de transition T. Une ligne pour chaque état de l'automate, une colonne pour chaque caractère (Terminal de l'automate).

- Une constante pour l'état initial.

- Un vecteur d'états finaux.

Exemple :

N \ T	0	1
S0	A	S0
A	∅	S0

S0 : état initial

F : | S0 | A |

I.2 Algorithme de réconnaissance d'une entité lexicale :

(* En entrée, nous disposons d'une entité lexicale, nous utilisons l'automate, afin de vérifier si elle appartient au langage où pas *).

Debut

Lire (entité) ;

tc := 1ère caractère de l'entitè ;

Tantque (Ec≠∅¿et (non fin d'entité)

Faire Ec :=T[Ec, tc] ;

tc :=tc+1 ;

Fait ;

Si Ec=∅ Alors 'Erreur : entité incorrecte'

Sinon Si Ec ∉ F

Alors 'Erreur : entité incorrecte'

Sinon 'entité correcte' ;

Codifier entité ;

Insérer dans la table des symboles (TS) ;

Fsi ;

Fsi ;

Fin.

Exemple 1 :

a- Donner un automate simple déterministe engendrant le langage forme de 'a' et 'b', mais contenant au moins deux 'a' où deux 'b' consécutifs.
b- Déduire l'algorithme de reconnaissance.
c- Analyser les chaines abaa¿et baba#.

Exemple 2 :

Soit le langage L⊂{a, b, c}+ formé des entités lexicales X et Y définies comme suit :

- X commence par 'c' et ne contient pas deux 'a' consécutifs.
- L'entité Y commence par 'a' et se termine par deux 'b' consécutifs.

a- Construire un automate simple déterministe pour la reconnaissance de ces entités.
b- Donner l'algorithme de reconnaissance des entités.