Université A.Mira, Bejaia
Faculté des Sciences Exactes
Département d'Informatique

Cours de Compilation

Analyse Lexicale

Mme D.Boulahrouz
boukredera@hotmail.com

1. Le rôle d'un Analyseur Lexical

Analyseur Lexical

L'analyseur lexical *est chargé de* ***lire*** *le texte d'entrée, caractère par caractère, de la gauche vers la droite et* ***isoler*** *les mots et leur classe. Il* ***découpe*** *et* ***regroupe*** *les caractères d'entrée en mots ou* ***lexèmes***.

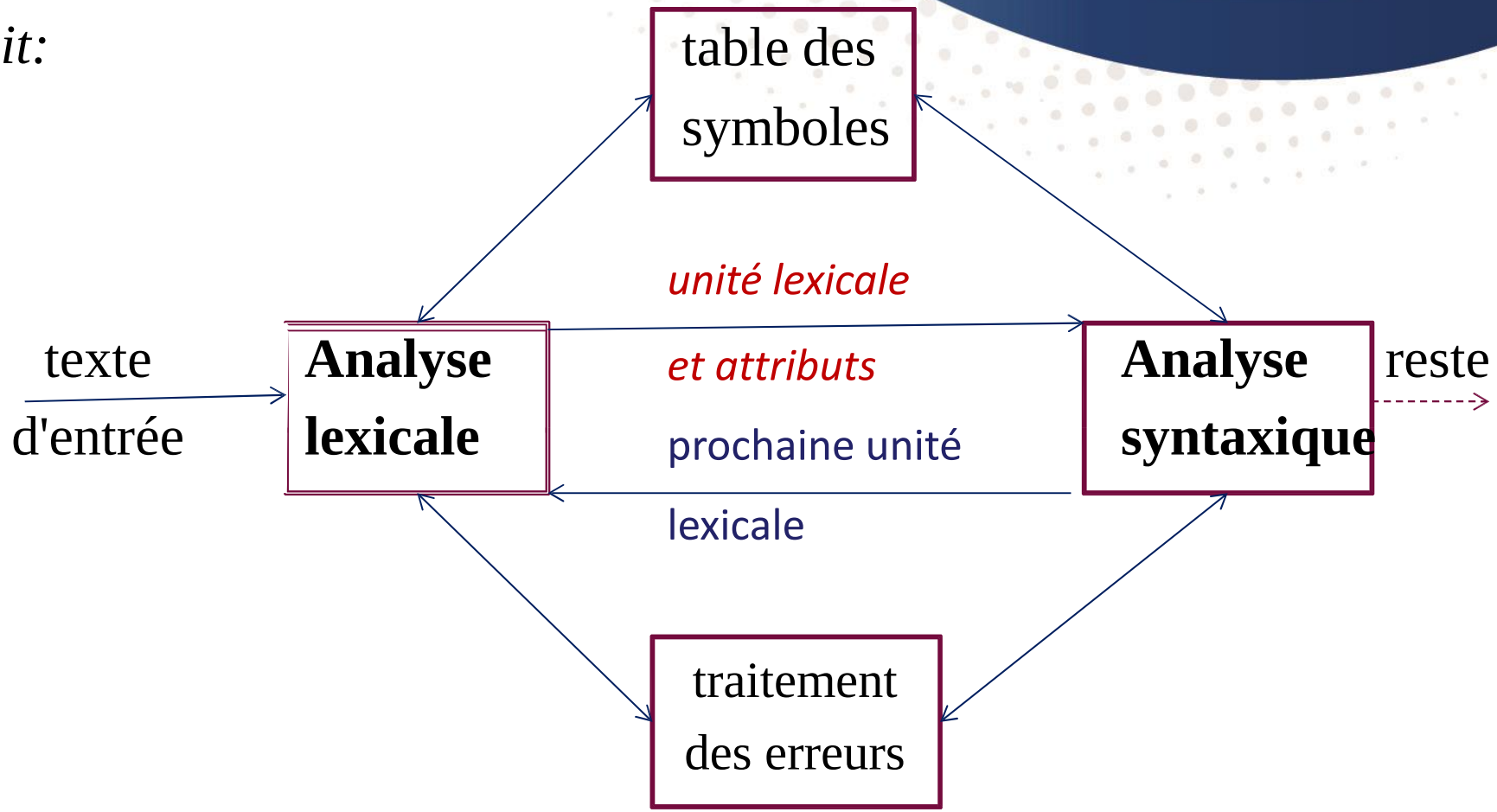
De plus, il doit:

- ***éliminer*** *les* ***blancs*** *(espaces, tabulations, fin de lignes) et les* ***commentaires***.

- ***détecter les erreurs*** *et associer des messages d'erreurs.*

1. Le rôle d'un Analyseur Lexical

L'interaction entre l'analyseur ***lexical*** *et l'analyseur* ***syntaxique*** *peut être schématisée comme suit:*



2. Principe d'un Analyseur Lexical

L'analyseur lexical *consiste à partir d'un programme qui est une suite de caractères séparés par des blancs (ou des séparateurs) à :*

- ***Spécifier*** *les différentes* ***entités lexicales*** *(les mots) du langage, parmi ces entités, on reconnaît :*
  - ✓ *Les identificateurs.*
  - ✓ *Les mots clés (Les mots réservés).*
  - ✓ *Les constantes.*
  - ✓ *Les séparateurs.*
- ***Éliminer*** *les blancs, ainsi que les commentaires.*
- ***Codifier*** *les différentes entités lexicales.*
- ***Construire*** *la table des symboles.*
- ***Gérer*** *les erreurs Lexicales*

3. Terminologie

- ***Unité lexicale (token)***: *Une unité lexicale est une suite de caractères qui a une signification collective : identificateur, entier, opérateur etc. c'est un* ***symbole terminal*** *de la grammaire du langage.*
- ***Modèle:*** *est une règle associée à une unité lexicale qui décrit l'ensemble des chaînes qui peuvent correspondre à cette unité lexicale:* ***expression régulière***
- ***lexème:*** *est une suite de caractères du texte d'entrée qui concorde avec le modèle d'une unité lexicale.*

Exemple: *35 est un lexème (un mot) qui appartient à l'unité lexicale (la classe) nombre.*

3. Terminologie

Remarques:

*Dans de nombreux langages, les **classes** suivantes couvrent la plupart des unités lexicales:*

1. *Une unité lexicale pour chaque mot clé.*
2. *Des unités lexicales pour les opérateurs, soit individuellement, soit par classes.*
3. *Une unité lexicale pour les identificateurs (noms de variables, fonctions, tableaux, structures...).*
4. *Une ou plusieurs unités lexicales pour les nombres et les chaînes.*
5. *Une unité lexicale pour chacun des signes de ponctuation, tels que les parenthèses gauche et droite, la virgule , le point-virgule...*

3. Terminologie

Exemple1

position := 50 + vitesse*temps

*Les **7 lexèmes** constitués sont : «position» «:=» «50» «+» «vitesse» «*» « temps»*

*Les **03 Unités lexicales** correspondantes sont: **nb, id et op***

- *«positions», «vitesse» et «temps» sont des **lexèmes** de l'unité lexicale **id***
- *«*», «+» et «:=» sont des **lexèmes** de l 'unité lexicale **op***
- *« 50 » est un **lexème** de l 'unité lexicale **nb***

3. Terminologie

Attributs des unités lexicales

Lorsque l'analyseur Lexical (scanner) reconnaît une unité lexicale, il la transmet à 'analyseur syntaxique (parser) avec plusieurs informations (attributs) qui s'y rattachent .

Exemple : *position := 50 + vitesse*temps*

Le scanner transmet au parser :

(id, «position») (op, affectation) (nb, 50) (op, add) (id, «vitesse») (op, mult) (id, «temps»)

3. Terminologie

Exemple2

Soit la portion du programme Pascal suivante:

```
Begin
    i:= i+j;
    If (i=j) then i:=0;
End.
```

Déterminer les Tokens (Unités lexicales) et les lexèmes correspondants

3. Terminologie

Exemple2

Soit la portion du programme Pascal suivante:

```
Begin
    i:= i+j;
    If (i=j) then i:=0;
End.
```

Tokens	Lexèmes
Mot clé	Begin, if, then, end
Identificateur	i, j
Opérateur arithmétique	+
Séparateur	; .
Parenthèses	( )
Affectation	:=
Nombre	0
Opérateur Comparaison	=

6. Reconnaissance des unités lexicales

Comment l'analyseur lexical reconnait-il les unités lexicales du programme à compiler?

6. Reconnaissance des unités lexicales

Etant donné un mot, m, et un langage L, engendré par une Grammaire G, est-ce que m $\in$ L ?

6. Reconnaissance des unités lexicales

03 Approches

Approche Simplifiée

Construction à la main basée sur les diagrammes de transition.

Approche plus Rigoureuse

basée sur les automates d'états finis (AEF).

Approche Automatique

Utilisation d'outils: générateurs d'analyseur lexicaux(LEX, FLEX…

Ecriture manuelle de l'analyseur lexical

*À partir du **1er caractère** d'une entité (mot, Lexème), on se branche à un algorithme particulier.*

Les diagrammes de transition sont une aide importante pour l'écriture d'analyseurs lexicaux.

7. Réalisation d'un Analyseur Lexical

Exemple2

Soit le langage L={aab,ab}

```
Tc: caractère courant;
Ts: caractère suivant;
#: Fin de la chaine;
Début
Tc← 1er caractère;
Si Tc = 'a' alors Tc← Ts
                  si Tc='a' alors Tc← Ts
                                    si Tc='b' alors Tc← Ts
                                                      si Tc=# alors « Chaine Acceptée »
                                                                sinon Erreur
                                                                   finsi
                                                      sinon Erreur finsi
                                    sinon si Tc='b' alors Tc← Ts
                                                      si Tc=# alors « Chaine Acceptée »
                                                                sinon Erreur
                                                                finsi
                                    finsi
 sinon  Erreur
 finsi
Fin.
```

Exemple3

7. Réalisation d'un Analyseur Lexical

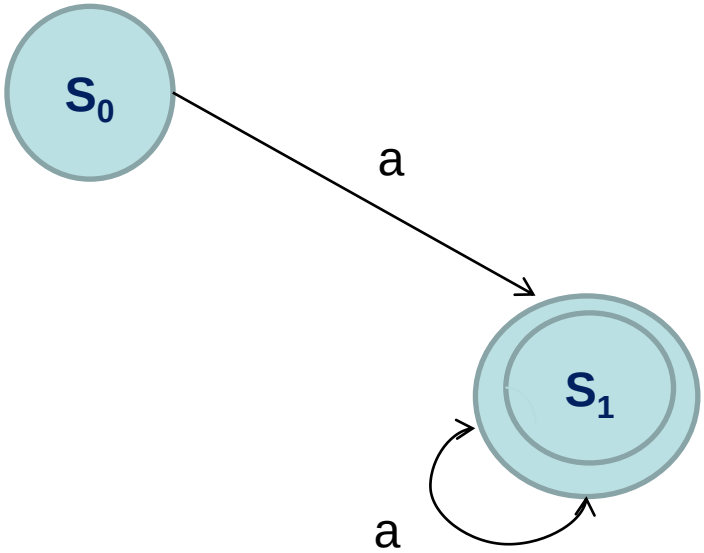
Ecrire un analyseur lexical qui reconnait les mots
L={a, aa, aaa, aaaa,…., a^n}

7. Réalisation d'un Analyseur Lexical

Exemple3

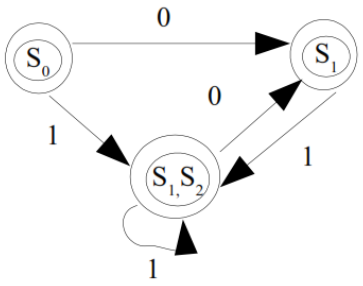
Ecrire un analyseur lexical qui reconnait les mots L={a, aa, aaa, aaaa,…., a^n}

```
Début
Tc← 1er caractère;
Si Tc = 'a' alors Tc← Ts
      TQ Tc='a' faire
            Tc← Ts
        FTQ
        si Tc=# alors « Chaine   Acceptée »
         sinon Erreur
          finsi
sinon Erreur
Fin.
```



7. Réalisation d'un Analyseur Lexical

Approche par Automates



7. Réalisation d'un Analyseur Lexical

7.2 Approche par Automates

- *Les étapes de réalisation de l'analyseur lexical qui utilise un automate sont comme suit :*
  - **a. *Représentation de l'automate*** *: on représente l'automate par* ***une matrice de transition****, les lignes représentent les* ***états*** *de l'automate, les colonnes représentent* ***l'alphabet****, et un élément de la matrice correspond à une* ***transition.***
  - **b. *Représentation des états finaux*** *: La représentation des états finaux se fait dans un vecteur associer à la matrice.*

7. Réalisation d'un Analyseur Lexical

Exemple

Exemple : soit le langage suivant : $L1=\{a+b\}$

Représentation de l'automate : L'automate correspond au langage $L1$ est comme suit :

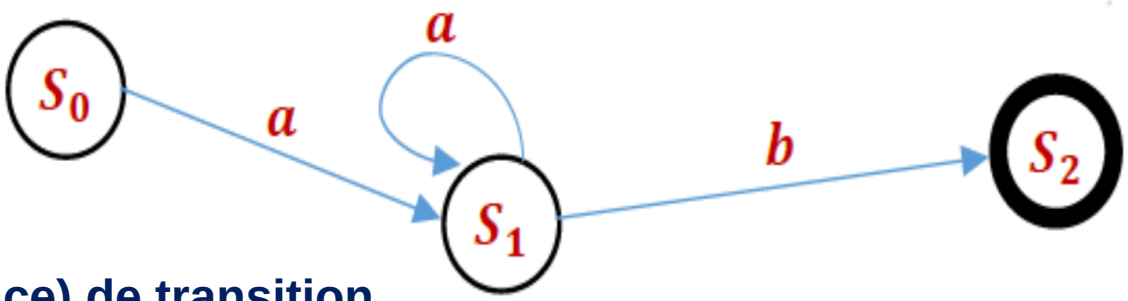


Table (matrice) de transition

	a	b
$S0$	$S1$	-1
$S1$	$S1$	$S2$
$S2$	-1	-1

Etat initial : S0

Vecteur des états finaux $Vf=\{S_2\}$

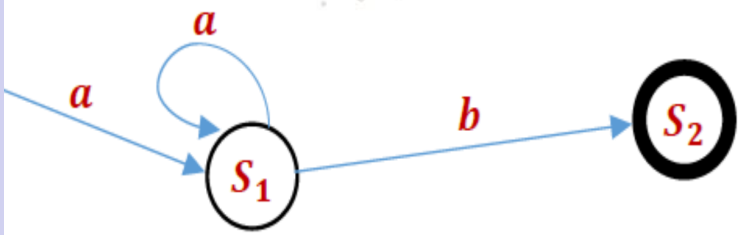
7. Réalisation d'un Analyseur Lexical

Exemple

Exemple : soit le langage suivant : $L1=\{a+b\}$

Algorithme d'analyse

tc:*terme courant*;
ts:*terme suivant*;
Ec :*Etat courant*;
M:*matrice de transition*;
V:*vecteur des états finaux*;
debut
  $tc \leftarrow$*1er caractère*;
  $Ec \leftarrow$*L'état initial*;
  Tant que $(tc \neq '\#'$ *et* $M[Ec,tc] \neq -1)$ *faire*
    $Ec \leftarrow M[Ec,Tc]$; $tc \leftarrow ts$;
  Fin tant que
Si $(tc='\#'$ *et* $Ec \in V)$ *alors Chaine acceptée*
sinon erreur
Fin si.
fin.



	a	b
$S0$	$S1$	-1
$S1$	$S1$	$S2$
$S2$	-1	-1

Vecteur des états finaux $Vf=\{S_2\}$

7. Réalisation d'un Analyseur Lexical

Approche Automatique

générateurs d'analyseur lexicaux(LEX, FLEX…

7. Réalisation d'un Analyseur Lexical

7.3 Approche Automatique

- ✓ *Il existe des outils comme l'utilitaire **LEX** qui accepte en entrée des spécifications d'unités lexicales sous forme d'expressions régulières et produit en sortie un programme écrit en langage C qui une fois compilé reconnait ces unités lexicales → Ce programme est **l'analyseur lexical**!*


THE END