



# Analyse lexicale

Université Assane Seck  
UFR Sciences et technologie  
Département Informatique

- 1 Introduction
- 2 Terminologie
- 3 Mise en œuvre

1 Introduction

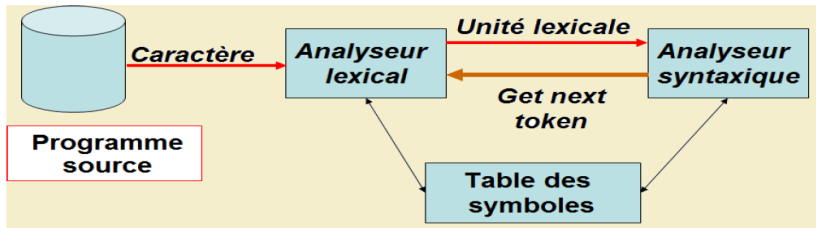
2 Terminologie

3 Mise en œuvre

# Définition

## Analyse lexicale

L'analyseur lexical (Scanner) fusionne les caractères lus dans le code source en groupes de mots qui forment logiquement des unités lexicales (tokens) du langage.



1 Introduction

2 Terminologie

3 Mise en œuvre

# Terminologie

**Une unité lexicale** est une suite de caractères qui est une signification collective

- Exemples

- Les chaînes  $>, <, =, \dots$  sont des opérateurs relationnels. L'unité lexicale est *OPREL* par exemple.
- Les chaînes *toto, ind, tab, supprimer, ...* sont des identificateurs
- Les chaînes *if, else, while, ...* sont des mots clés
- Les symboles  $;, . :$  sont des séparateurs

# Terminologie

- **Un modèle** est une règle associée à une unité lexicale qui décrit l'ensemble des chaînes de programme qui peuvent correspondre à cette unité lexicale
- On appelle **lexème** toute suite de caractère du programme source qui concorde avec le modèle d'une unité lexicale

# Terminologie

- Quelques exemples

Unité lexicale	Modèle	Lexème
IDENT	Toute suite non vide de chiffres, lettres ou _ ne commençant pas par chiffre	truc, ajouter, malick
NOMBRE	Toute suite non vide de chiffres précédé de + ou -	-12, 96
REEL	Tout lexème correspondant à l'unité lexicale NOMBRE.NOMBRE-{+,-}	12.8, -9.00

- Pour décrire les unités lexicales, on utilisera les expressions régulières



1 Introduction

2 Terminologie

3 Mise en œuvre

# Mise en œuvre d'un analyseur lexical

- Les unités lexicales sont décrites par des expressions régulières
- Soit IDENT défini ci-dessus
- $\text{IDENT} = (a|b|.....|z|A|B|...|Z|_)(a|b|.....|z|A|B|...|Z|0|1|....|9|_)^*$
- C'est un peu répétitif et peu passionnant
- On s'autorise alors des définitions régulières et le symbole \_ sur des types ordonnés (lettres, chiffres ...)

# Mise en œuvre d'un analyseur lexical

Une définition régulière est une suite de la forme  $\left\{ \begin{array}{l} d_1 = r_1 \\ \dots \\ d_n = r_n \end{array} \right.$

où chaque  $r_i$  est une expression régulière sur  $\Sigma \cup \{d_1, d_2, \dots, d_n\}$  et chaque  $d_i$  est un nom différent

Exemple: l'unité lexicale IDENT (identificateurs) en C devient:

$$\left\{ \begin{array}{ll} \text{lettre} & = A - Z | a - z \\ \text{chiffre} & = 0 - 9 \\ \text{sep} & = \\ \text{IDENT} & = (\text{lettre} | \text{sep})(\text{lettre} | \text{chiffre} | \text{sep})^* \end{array} \right.$$

# Implémentation d'analyseur lexical

- Langage C (A.E.F, structures de contrôle)
- FLEX (définitions régulière). Voir TP.

# Erreur lexicale

- Peu d'erreurs sont détectées au niveau lexical
- Il y a erreur lorsque l'analyseur rencontre une suite associée à aucun modèle lexical
- Exemples: en langage C l'analyseur lexical détecte une erreur dans le cas
  - `li` génère une erreur
  - `*8rer` génère une erreur
- Par contre **`esle`** à la place de **`else`** ne génère pas d'erreur