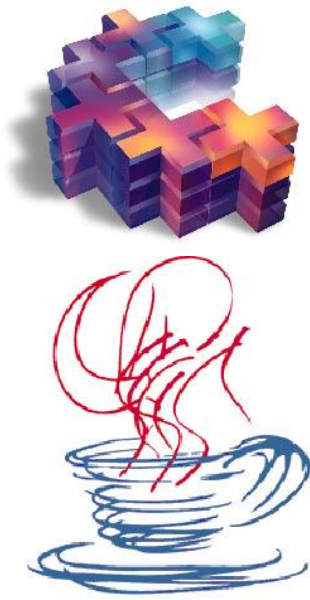


PRINCIPES ET TECHNIQUES DES COMPILATEURS

ANALYSE LEXICALE





PLAN

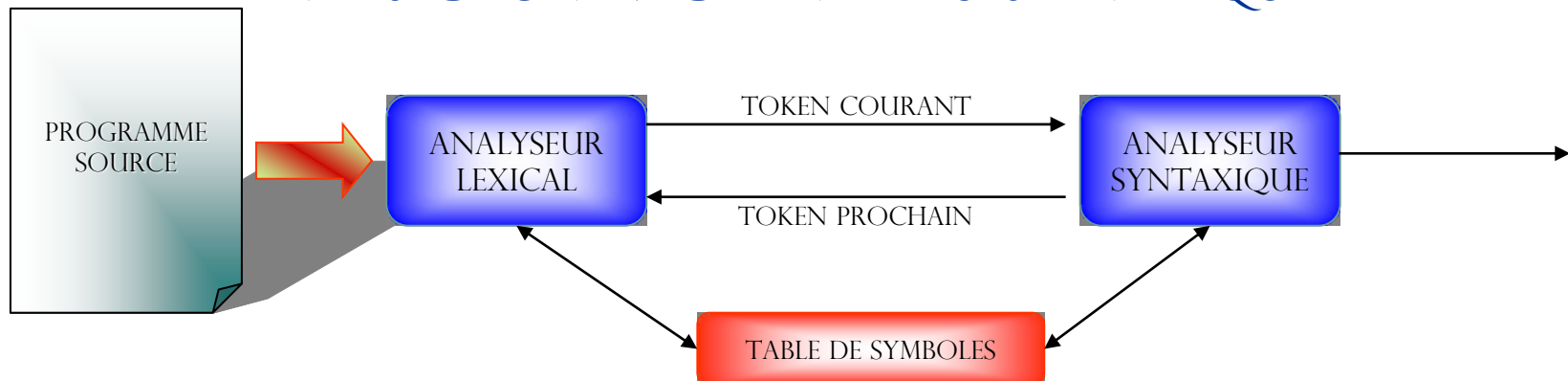
- Généralités
- Unité lexicale, Lexème et Modèles
- Définition régulière
- Diagramme de transition

RÔLE DE L'ANALYSEUR LEXICAL



- Lire les caractères du texte d'entrée.
- Supprimer les blancs, les commentaires, etc.
- Former des unités lexicales .
- Passer des couples <unité lexicale, valeur lexicale> à l'Analyseur Syntaxique.
- Relier les messages d'erreurs issus du compilateur au code source.

SCHÉMA USUEL D'IMPLÉMENTATION DE L'ANALYSEUR LEXICAL – INTERACTION AVEC L'ANALYSEUR SYNTAXIQUE



UNITÉS LEXICALES, LEXÈMES ET MODÈLES



A propos d'une unité lexicale reconnue dans le texte source on doit distinguer trois notions importantes :

- L'unité lexicale.
- le lexème.
- le modèle.

UNITÉS LEXICALES, LEXÈMES ET MODÈLES



UNITÉ LEXICALE

Pour la plupart des langages de programmation, les constructions suivantes sont traitées comme des unités lexicales :

- Mots clés.
- Opérateurs arithmétiques
- Opérateurs logiques
- Identificateurs.
- Séparateurs ('(', ')', ',', '...', etc.)

UNITÉS LEXICALES, LEXÈMES ET MODÈLES



LEXÈME

Un lexème est une suite de caractères du programme source qui concordent avec le modèle de l'unité lexicale. Exemple :

```
const max_length = 256;
```

Dans la déclaration précédente, la chaîne de caractères `max_length` est un lexème de l'unité lexicale Identifier.

UNITÉS LEXICALES, LEXÈMES ET MODÈLES



MODÈLE

le modèle sert à spécifier l'unité lexicale.

- Pour les mots réservés tels que `const`, `if`, `while`, etc. le lexème et le modèle coïncident généralement. Le modèle de l'unité lexicale `const` est la chaîne `const`.
- Pour une unité lexicale `rel_oper` qui représente les opérateurs relationnels, le modèle est l'ensemble des opérateurs relationnels : `<`, `<=`, `==`, `>=`, `>`, `!=`
- Pour décrire précisément les modèles des unités lexicales plus complexes tels que les identificateurs et les nombres, on utilise les expressions régulières .
- Des langages et outils permettent d'engendrer une reconnaissance efficace par automates finis des expressions régulières.

UNITÉS LEXICALES, LEXÈMES ET MODÈLES



UNITÉ LEXICALE	LEXÈMES	DESCRIPTION INFORMELLE DES MODÈLES
const	const	const
if	if	if
rel_oper	< , <= , == , != , >= , >	< <= == != >= >
identifier	e pi length	Lettre suivie de lettres ou de chiffres ou le caractère '_'

LES DÉFINITIONS RÉGULIÈRES



Les définitions régulières permettent de donner des noms à des ER définies sur un alphabet Σ à partir de symboles de base et de les utiliser comme s'ils étaient des symboles de Σ .

- $d_1 \rightarrow r_1$
- $d_2 \rightarrow r_2$
- ...
- $d_n \rightarrow r_n$

d_i est un nom distinct et chaque r_i est une ER sur les symboles de $\Sigma \cup \{d_1, d_2, \dots, d_{i-1}\}$.

RAPPEL : LES EXPRESSIONS RÉGULIÈRES



EXEMPLE

letter $\rightarrow A | B | \dots | Z | a | b | \dots | z$
digit $\rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$
identifiant $\rightarrow \text{letter} (\text{letter} | \text{digit})^*$
number $\rightarrow \text{digit}^+$



DIAGRAMMES DE TRANSITIONS

- Les diagrammes de transitions décrivent les actions qui sont réalisées quand l'analyseur syntaxique appelle un analyseur lexical pour fournir la prochaine unité lexicale.
- Un état initial du diagramme.
- En entrant dans un état on lit le prochain caractère. Si l'étiquette d'un arc sortant de l'état courant concorde avec le caractère d'entrée on passe à l'état pointé par cet arc. Autrement on signale une erreur.

DIAGRAMMES DE TRANSITIONS



EXEMPLES

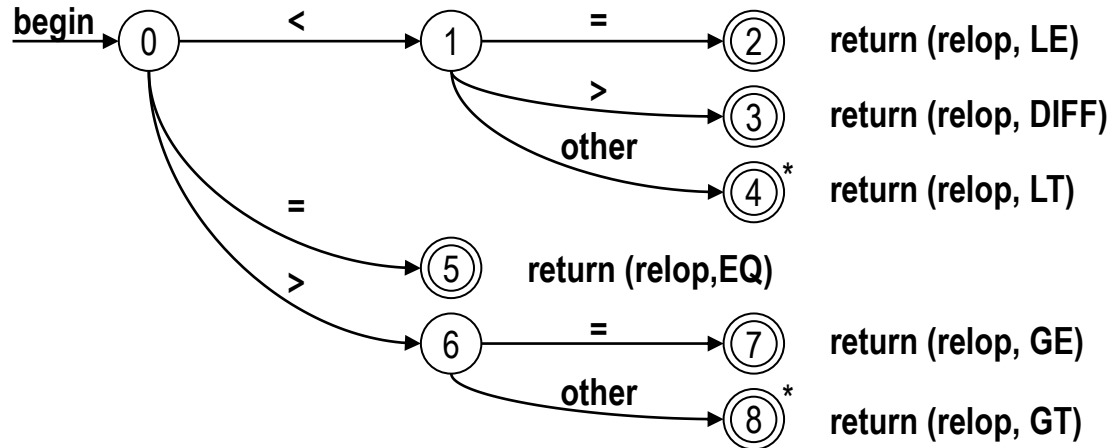


DIAGRAMME DE TRANSITION POUR LES OPÉRATEURS DE RELATION

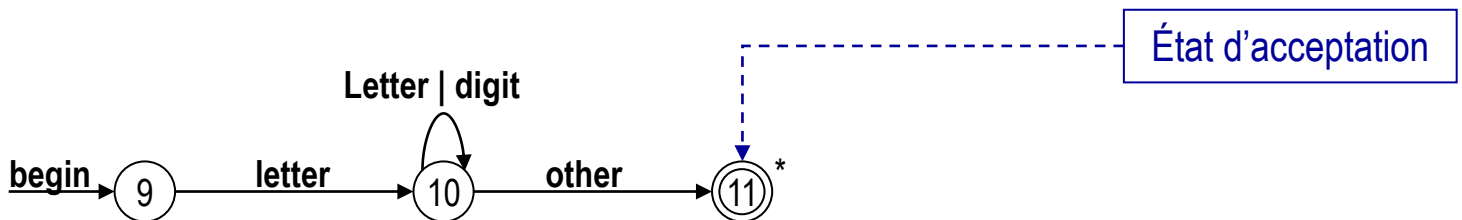


DIAGRAMME DE TRANSITION POUR LES IDENTIFICATEURS

RÉALISATION DE DIAGRAMMES DE TRANSITIONS

