

Au menu de l'UE Compilation

Introduction

I - Programmation dirigée par la syntaxe

II - Analyse lexicale - Analyse syntaxique

III – Description formelle d'un langage

 III.1 – Expressions régulières (ou rationnelles) – Automates finis

 III.2 – Grammaires algébriques (ou hors-contexte)

A - Programmation par automates finis

I – Analyse lexicale par automate fini

 I.1 – Reconnaissance des items lexicaux

 I.2 – Analyse lexicale et actions

 I.3 – Analyse lexicale et erreurs

II – Analyse syntaxique par automate fini

 II.1 – Reconnaissance des données licites

 II.2 – Analyse syntaxique et actions

 III.3 - Analyse syntaxique et erreurs

III - Programmation d'un automate fini déterministe

 III.1 – Programmation directe

 III.2 – Programmation par interpréteur de tables

III.3 – Traitement des erreurs dans l'analyse syntaxique

B - Analyse syntaxique descendante de gauche à droite (DGD)

I - Limite des automates finis

II - Analyseur DGD procédural - Points de génération

C – Construction d'un compilateur

I - Compilateur

II - Table des symboles - Compilation des déclarations

III - Compilation des expressions - Calcul de type

IV - Compilation des instructions

V - Compilation des procédures

VI - Compilation séparée - Édition de liens

D - Automates à pile - Grammaires LL(1)

I – Analyse DGD par automate à pile

II - Analyse DGD et grammaire LL(1)

III – Analyseur associé à une grammaire LL(1)

I-Analyse lexicale par automate fini

I.1 – Reconnaissance des items lexicaux

Rappel analyse lexicale

► Objectifs :

- ▶ Lecture d'une suite de caractères
- ▶ Reconnaissance des items lexicaux nécessaires à l'analyse syntaxique à partir de V_T
- ▶ Ici $V_T = \{\text{caractères autorisés}\}$

=> **Modélisable par automate fini**

Automate fini résultant

- ▶ **Union des automates de reconnaissance de chaque item lexical**
- ▶ Chaque « branche » de l'automate = reconnaissance d'un item lexical
- ▶ **Arrêt après** la reconnaissance d'**un item lexical**
- ▶ Traitement erreur (**caractère non autorisé**) = item lexical AUTRES
- ▶ **Attention** aux caractères autorisés mais ne faisant pas partie d'un item lexical (espaces, retours à la ligne, tabulations, ...) : doivent apparaître dans l'automate

I-Analyse lexicale par automate fini

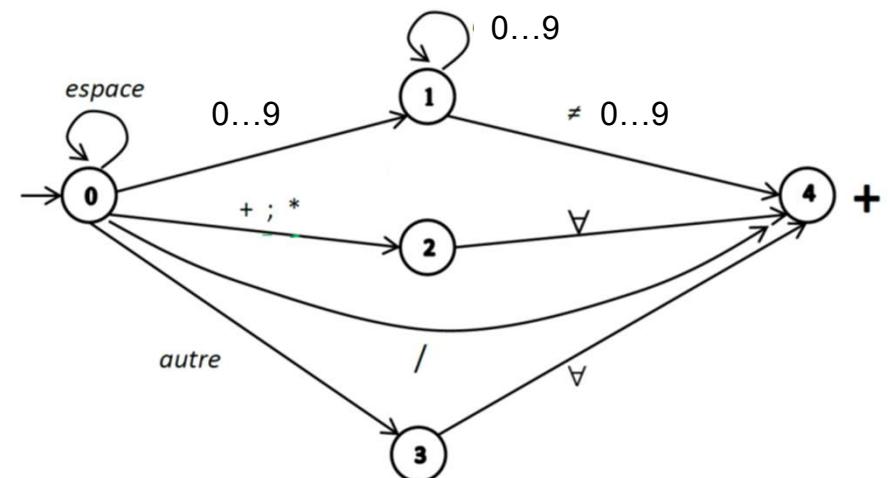
I.1 – Reconnaissance des items lexicaux

Exemple langage expressions

Soit un langage d'expressions arithmétiques

- ▶ Items lexicaux à reconnaître = { *nbentier*, *PLUS*, *ETOILE*, *PTVIRG*, *BARRE* } où *BARRE* est l'item de fin de donnée.
 - ▶ Ici $V_T = \{0\ldots9, +, *, ;, /, \text{espace}\}$
 - ▶ La reconnaissance de *nbentier* nécessite de **connaitre un caractère en avance**
- => Généralisé à tous les items (sauf celui de fin de donnée **BARRE**)**

Automate fini résultant



Union des automates de reconnaissance de chaque item lexical
+ généralisation du caractère connu en avance

I-Analyse lexicale par automate fini

I.1 – Reconnaissance des items lexicaux

Exercice 1 : donner l'analyseur lexical (reconnaissance des items) du langage Monnaie sous la forme d'un automate.

- ▶ Items à reconnaître :
 $\{\text{ANCIEN}, \text{NOUVEAU}, \text{PAR}, \text{ident}, \text{nbentier}, \text{nbrel}, \text{PLUS}, \text{MOINS}, \text{PTVIRG}, \text{BARRE}\}$
- ▶ Où
 - ▶ les expressions régulières de *nbentier* et *nbrel* sont resp.
 $('0' | '1' | \dots | '9')^+$ et $('0' | '1' | \dots | '9')^+ . ('0' | '1' | \dots | '9')^*$
 - ▶ un item *ident* est une suite non vide de lettres majuscules ou minuscules
 - ▶ *ANCIEN*, *NOUVEAU*, et *PAR* sont des mots réservés

I-Analyse lexicale par automate fini

I.2 – Analyse lexicale et actions

5

Actions au cours de l'analyse lexicale

Action = code à exécuter sur une transition

► Codage des items lexicaux

► Mise à jour des attributs lexicaux

Ex pour le langage expressions

► Codes items à produire :

- $V_T = \{ 0\dots9, +, *, ;, /, \text{espace} \}$
- final int

NBENTIER = 0, PLUS = 1, PTVIRG = 2,
ETOILE = 3, BARRE = 4, AUTRES = 5;

► Attribut lexical à définir :

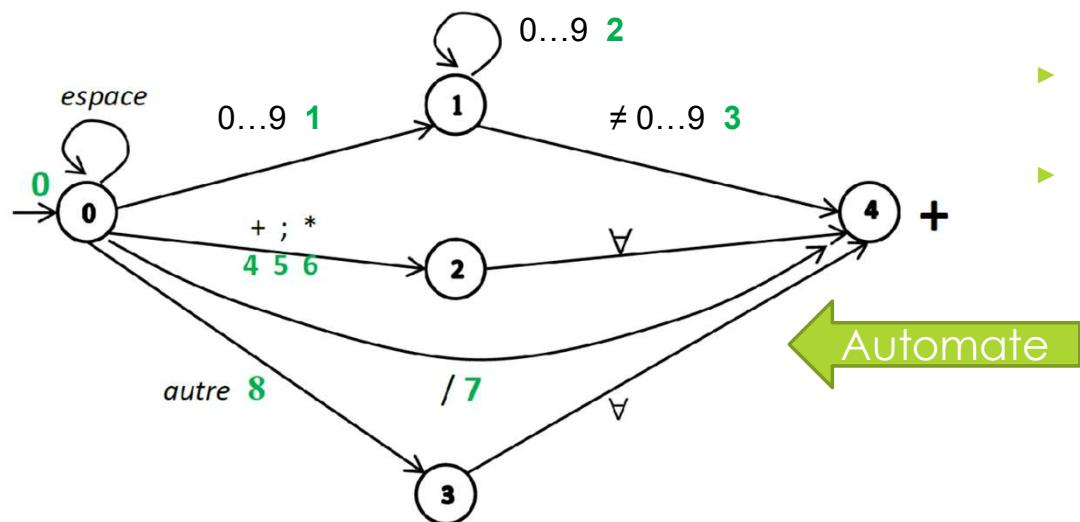
- valEnt associé à l'item nbentier

I-Analyse lexicale par automate fini

I.2 – Analyse lexicale et actions

6

- **Exemple** : analyseur lexical du langage des expressions.



- **Items à produire :**
 - NBENTIER, PLUS, ETOILE, PTVIRG , BARRE et AUTRES
- **Attributs à définir :**
 - valEnt
- **Actions pour les attributs associés (mise à jour carLu omise)**
 - String s = "";
 - 1 : s = "" + carLu ;
 - 2 : s = s + carLu ;
 - 3 : valEnt = Integer.parseInt(s); item = NBENTIER ;
 - 4 : item = PLUS ;
 - 5 : item = PTVIRG ;
 - 6 : item = ETOILE ;
 - 7 : item = BARRE ;
 - 8 : item = AUTRES ; System.out.println("caractère non autorisé");

I-Analyse lexicale par automate fini

I.2 – Analyse lexicale et actions

7

Exercice 1 : donner l'analyseur lexical
(reconnaissance + **actions**) du langage
Monnaie sous la forme d'un automate.

Actions au cours de l'analyse lexicale

- ▶ **Codage des items lexicaux**
- ▶ **Mise à jour des attributs lexicaux**

Pour le langage Monnaie

▶ **Codes items à produire :**

0=ANCIEN	6=PTVIRG
1=NOUVEAU	7=PLUS
2=PAR	8=MOINS
3=ident	9=BARRE
4=nbentier	10=AUTRES
5=nbréel	

▶ **Attributs lexicaux à définir :**

- ▶ **int valEnt** = valeur de l'item nbentier
- ▶ **double valReel** = valeur de l'item nbréel
- ▶ **int numIdCourant** = index dans **tabIdent** du dernier ident lu
- ▶ **String[] tabIdent** = table des identificateurs réservés et lus
- ▶ **String chainIdent(int numId)** = chaîne de l'ident

I-Analyse lexicale par automate fini

I.2 – Analyse lexicale et actions

Gestion des identificateurs

Utilisation du tableau *tabIdent*

- ▶ **Initialisation :**
 - ▶ Identificateurs des mots réservés
 - ▶ Respect de l'ordre de codage des items
- ▶ **Distinction identificateurs réservés et autres**
NBRES = nbre d'identificateurs réservés

- ▶ **Mise à jour :**
 - ▶ Ajout des identificateurs de monnaie lus
 - ▶ **Pas de doublons dans *tabIdent***

Pour le langage Monnaie :

- ▶ **Initialisation** : **NBRES** = 3

0=ANCIEN	6=PTVIRG
1=NOUVEAU	7=PLUS
2=PAR	8=MOINS
3=ident	9=BARRE
4=nbentier	10=AUTRES
5=nbréel	

<i>tabIdent</i>	
0	ANCIEN
1	NOUVEAU
2	PAR
3	FSuisse
4	Livre
5	Yen

- ▶ **Mise à jour :**

- ▶ FSuisse nouveau 0.62 ;
- ▶ Livre nouveau 1.58 ;
- ▶ Yen nouveau 0.009 ;
- ▶ **FSuisse** par 100 ancien 62 nouveau 62.5 ; /

▶ **Erreur lexicale : caractère non autorisé**

- ▶ Génération de l'item lexical AUTRES
- ▶ Gestion de AUTRES par l'analyseur syntaxique

▶ **Erreur lexicale : problème lors du calcul d'un attribut lexical**

- ▶ Exemples : entier trop grand, table des idents pleine...
- ▶ Arrêt de l'analyse : System.exit ou exception lancée indirectement