

note

Questions

Évaluation: examen oral

4 tp (1 d'intro et trois à rendre) -1/2 par tp non rendu (évaluation oral du tp)

## éléments importants

### outils

	outils pour l'analyse lexicale	outils pour l'analyse syntaxique
type de langage	langage régulier	langage algébrique
description	expressions régulières	grammaires algébriques
formalisme sous-jacent	AF(N)D	automates à pile

## def compilation \* Analyser une description \* Synthétiser une exécution \*  
Conserver la sémantique

On part d'une forme source pour arriver vers une forme objet

### def interprétation

- Expliquer et donner une sémantique
- Parcourir un graphe, noeud;instruction

---

### objets

Arbre de dérivation

## Analyses « universelles »

Fonctionnent pour toute grammaire algébrique :

- ▶ algorithme de Cocke-Younger-Kasami pour des grammaires en forme normale de Chomsky ;
- ▶ algorithme de Earley pour des grammaires algébriques quelconques.

Complexité en  $O(n^3)$ , où  $n$  est la longueur du mot à analyser.

On préférerait  $O(n)$ ... quitte à se restreindre à certains types de grammaires.

Figure 1: outils\_analyse\_tableau\_02

## Analyses dédiées à certaines grammaires

	Analyse descendante	Analyse ascendante
nom	LL(k)	LR(k)
dérivation construite	gauche	droite
ordre constr arbre	préfixe	postfixe
opérations	lecture et expansion	lecture et réduction
on part	de l'axiome	du mot à reconnaître
outil	javaCC	Cup

- ▶ plus efficaces ;
- ▶ cas particulier de l'automate des items ;
- ▶ ne fonctionnent que pour certaines classes de grammaires.

64/64

Figure 2: outils\_analyse\_tableau\_03

Trois concepts de base: Analyse: 1. Lexique 2. Syntaxe 3. sémantique 4. édition\_des\_lien

statique\_ou\_dynamique

But du langage, combler un fossé sémantique (=niveau d'abstraction, résolution de problème)

Idéal si un langage de programmation est proche du langage naturel

Notation postfixé

fonction\_stricte

Compilation\_indépendante

auto\_interprète

## Outils de génération de compilateur:

Flex: analyseur lexicale Bison: analyseur syntaxique

---

On ne s'occupe pas explicitement du parcours d'un arbre

### Définition (langage engendré)

Soit  $G$  une grammaire algébrique d'axiome  $S$ . Le langage engendré par  $G$  est défini par:  $L(G) = \{ m \text{ appartenant à } V_T^* \mid S \Rightarrow^* m \}$

Déterminer si un mot  $m$  appartient au langage engendré, c'est déterminer si:  $S \Rightarrow^* m$ ?

### Définition (dérivation directe)

Soient  $Beta$  et  $Beta'$  appartenant à terminaux/non-terminaux.  $Beta$  se dérive directement en  $Beta'$  selon  $G$ , noté  $Beta \Rightarrow G Beta'$ , s'il existe des mots  $gamma$ ,  $gamma'$  appartenants aussi à terminaux/-non et une production  $X \rightarrow alpha$  tel que  $Beta = gamma X gamma' \quad Beta' = gamma alpha gamma$

### Définition (dérivation)

Soient  $Beta$  et  $Beta'$  appartenant à  $t/nt$ . Une suite de mots  $Beta_0$  à  $Beta_N$  est une dérivation de  $Beta$  en  $Beta'$ . Si  $Beta = Beta_0$  et  $Beta' = Beta_N$  et que toute les transitions de  $N_i$  à  $N_{i+1}$  sont des production de  $G$ .

### Définition

La dérivation est une fermeture transitive de la dérivation directe

## Grammaire régulière

Rappel: une production ne peut avoir qu'un symbole terminal (à gauche ou à droite) dans son membre de droite.

## Arbre syntaxique

Arbre dont les feuilles sont des composante d'un mot / d'une phrase

## Grammaire pathologique

Quand il y a une erreur dans la grammaire. C'est comme un bug dans un code.

Les générateur d'analyseur syntaxique ne peuvent prendre au mieux qu'une grammaire algébrique

Pas facile de prouver qu'une grammaire est ambiguë. Or on a besoin d'une grammaire stable (sinon, un code se ferait interprété de plusieurs façons différentes de façon aléatoire).

## Automate à pile

C'est un 7-tuplet car contient les éléments d'un automate à état fini mais avec un alphabet et un symbole initial pour la pile.

## Automate item

L'item est le mot clé. L'item d'une grammaire  $G$  est de la forme  $x \rightarrow \alpha^* \beta$   
L'étoile (le point représente la position du pointeur)

Si on a  $n$  choix possible à chaque étape pour l'exploration de l'arbre, on a  $n^m$ .  
Pour simplifier, on supprime tout non-déterminisme pour avoir seulement un choix par étape. S'il y a une expansion (la pile va grandir) elle doit être unique.  
L'automate des items y arrive pas vraiment.

Associer un AP à une grammaire algébrique.

Analyseur\_récuratif