1. Présentation générale
2. Analyse lexicale
   1. Rappel sur les automates et Expressions régulières
   2. LEX
3. Analyse syntaxique
   1. Analyse ascendante
   2. Analyse descendante
   3. YACC
4. Analyse sémantique
   1. Diverses classes de contrôles
   2. Grammaires attribuées
   3. Analyse sémantique avec YACC
5. Arbre abstrait
6. Machine virtuelle

Partie frontale

(Dépend du langage source mais pas du langage cible)

L’analyseur lexical (AL) reconnait les lexèmes d’un programme et la classe (unité lexicale (token)) à laquelle chacun appartient.

Ex : Delta2 :=x\*3.14 + f(4 ) ;

L’AL découvre :

* Delta2 et nous indique que c’est un identificateur (IDF)
* Des espaces et les ignore
* := l’AL nous indique que c’est un opérateur d’affectation (OPAFF)

L’analyseur syntaxique (AS) vérifie qu’une suite de lexèmes (ou plutôt d’unités lexicales) a bien été générée en respectant une grammaire.

Partie finale

(Dépend du langage cible mais pas du langage source)

Programme source -> Analyseur Lexical -> fournit une unité lexicale -> Analyseur syntaxique

Demande une unité lexicale <-

1. Quelques petits problèmes :

X :=3+f( 4 );

Y := 5 \* 2;

Affectation -> IDF OPAFF expression point-virgule

Expression -> expression PLUS e1 (élément1)

1. Reconnaissance des identificateurs

If(a+b\*(3.14-c)==42\*g(4)){…}

Mot clé if -> Mot réservé

~~Identificateur if~~

Problème surprenant pouvant apparaitre si on conçoit mal un langage de programmation

Programme Fortran de la NASA. Pilotage de la sonde Mariner vers Vénus.

a) FORTRAN : AS TE RIX = 15

b) Déclaration implicite de variable.

I -> entier

J -> entier

I3 -> entier

ASTERIX -> réel

IASTERIX -> entier

c) boucle

Do 25 I=1,10 (boucle)

Do 15 I=1.10 DO15I=1.10 (affectation)

Int analyse\_lexicale(){

Boucle{

Reconnaitre IDF

Ou Reconnaitre Entier

Ou Reconnaitre REEL

Ou Reconnaitre <,>,<=,=>, !=, ==

Ou Reconnaitre :=

}

}

Zoom sur l’analyse syntaxique

La forme des règles de grammaire détermine la classe de la grammaire dans la hiérarchie de *Chomsky.*

Grammaire context-free (hors contexte) =

Faire l’AS de 2-3+1 c’est vérifier si cette phrase peut être générée par la grammaire G2