

Cartouche du document

Année : ING 1

Matière : Théorie des langages

Activité : Travail dirigé

Objectifs

Ce travail dirigé a pour but :

- l'introduction à la théorie des langages
 - Grammaire
 - Dérivation
- La décomposition d'un problème en analyse lexicale, syntaxique et sémantique

Sommaire des exercices

- 1 - Dérivation de règles
- 2 - Analyse lexicale et syntaxique
- 3 - Le langage SQL

Corps des exercices

1 - Dérivation de règles

Énoncé :

En théorie des langages, quand les symboles sont déjà des suites de symboles, on parlera de vocabulaire plutôt que d'alphabet et de phrases plutôt que de mots.

On considère le petit langage naturel suivant :

- le vocabulaire $A = \{ \text{le, la, fille, jouet, regarde} \}$
- Une phrase est de la forme groupe nominal verbe groupe nominal.
- Un groupe nominal est de la forme déterminant nom.
- Les déterminants sont la ou le.
- Les noms sont fille ou jouet.
- Le verbe est regarde.

Question 1)

Énoncé de la question

Etablir la grammaire de ce langage

Solution de la question

Le fichier

// La grammaire de notre petit langage naturel

$T = \{ \text{le, la, fille, jouet, regarde} \}$

$N = \{PH, GN, N, D, V\}$
 $S = PH$
 $P = \{$
 $PH \rightarrow GN V GN$
 $GN \rightarrow D N$
 $N \rightarrow \text{fille} \mid \text{jouet}$
 $D \rightarrow \text{la} \mid \text{le}$
 $V \rightarrow \text{regarde}$
 $\}$

Question 2)

Énoncé de la question

Montrer par dérivation que *la fille regarde le jouet* est une phrase du langage.

Solution de la question

Dans les différentes dérivations qui suivent, on a composé à droite et à gauche par ϵ .

$PH \rightarrow GN V GN \rightarrow D N V D N \rightarrow \text{la fille regarde le jouet}$.

la fille regarde le jouet n'est composé que d'éléments terminaux. Il s'agit donc d'une phrase de notre langage.

Question 3)

Énoncé de la question

Montrer par dérivation que *le jouet regarde la fille* est une phrase du langage.

Solution de la question

$PH \rightarrow GN V GN \rightarrow D N V D N \rightarrow \text{le jouet regarde la fille}$

Question 4)

Énoncé de la question

Que pensez-vous de la deuxième phrase ?

Solution de la question

Elle n'est pas sémantiquement correcte

2 - Analyse lexicale et syntaxique

Énoncé :

On considère le mini-langage de programmation déjà étudié dans un précédent travail.

- 01) L'alphabet V du langage L est formé des lettres $\{a, \dots, z, A, \dots, Z\}$, des chiffres $\{0, \dots, 9\}$, du point $\{.\}$, de la quote $\{'\}$ et les caractères $\{+, -, *, /, =\}$.

- 02) Un identificateur est de la forme : $\text{identificateur} \rightarrow \text{lettre}(\text{lettre}|\text{chiffre})^*$
 - 03) Un nombre est de la forme : $\text{nombre} \rightarrow \text{chiffre}(\text{chiffre})^* | \text{chiffre}(\text{chiffre})^*.\text{chiffre}(\text{chiffre})^*$
 - 04) Un opérateur numérique est de la forme : $\text{opérateur} \rightarrow + | - | / | *$
 - 05) Une expression numérique est de la forme : $\text{en} \rightarrow (\text{nombre} | \text{identificateur})((\text{opérateur})(\text{nombre} | \text{identificateur}))^*$
 - 06) Une affectation numérique est de la forme : $\text{an} \rightarrow (\text{Identificateur})=(\text{en})$
 - 07) Un littéral est de la forme : $\text{littéral} \rightarrow '(\text{symbole})^*$
 - 08) Une expression alphanumérique est de la forme : $\text{ea} \rightarrow (\text{littéral}|\text{identificateur})((+)\text{littéral}|\text{identificateur})^*$
 - 09) Une affectation alphanumérique est de la forme : $\text{aa} \rightarrow (\text{Identificateur})=(\text{ea})$
 - 10) Une expression syntaxiquement exacte du langage défini sur V est de la forme : aa|an
- On cherche les expressions syntaxiquement exactes.

Question 1)

Énoncé de la question

Décomposer ce problème en analyse lexicale et analyse syntaxique

Solution de la question

L'analyse lexicale

Les lettres {a, ..., z, A, ..., Z}, des chiffres {0, ..., 9}, du point {.}, de la quote {'} et les caractères {+, -, *, /, =} définissent l'alphabet.

Les phrases suivantes définissent les règles pour fabriquer les lexèmes.

- Un identificateur est de la forme : $\text{identificateur} \rightarrow \text{lettre}(\text{lettre}|\text{chiffre})^*$
- Un nombre est de la forme : $\text{nombre} \rightarrow \text{chiffre}(\text{chiffre})^* | \text{chiffre}(\text{chiffre})^*.\text{chiffre}(\text{chiffre})^*$
- Un opérateur numérique est de la forme : $\text{opérateur} \rightarrow + | - | / | *$
- Un opérateur alpha-numérique est de la forme : $\text{plus} \rightarrow +$
- Un littéral est de la forme : $\text{littéral} \rightarrow '(\text{symbole})^*$
- Une affectation est de la forme : $\text{affectation} \rightarrow =$

L'analyse syntaxique

Pour l'analyse syntaxique, on parle de vocabulaire plutôt que d'alphabet

Le vocabulaire est formé des catégories de lexèmes obtenues après l'analyse lexicale. Ces catégories sont identificateur, nombre, littéral, opérateur, plus et affectation.

Les phrases suivantes définissent les règles pour valider la suite de catégories de lexèmes obtenue après l'analyse lexicale.

- $\text{en} \rightarrow (\text{nombre} | \text{identificateur})((\text{opérateur})(\text{nombre} | \text{identificateur}))^*$
- $\text{an} \rightarrow \text{identificateur affectation en}$

- $ea \rightarrow (\text{litteral|identificateur})(\text{plus}(\text{litteral|identificateur}))^*$
- $aa \rightarrow \text{identificateur affectation } ea$
- $\text{expValide} \rightarrow an \mid aa$

Question 2)

Énoncé de la question

Etablir qu'au niveau syntaxique, le langage est la réunion de deux langages.

Solution de la question

Si on note L_1 le langage formé des expressions valides de type numérique et L_2 le langage formé des expressions valides de type alpha-numérique alors le langage des expressions valides $L = L_1 \cup L_2$.

3 - Le langage SQL

Énoncé :

Soit le sous langage de SQL, permettant d'écrire des requêtes simplifiées de projection et de sélection sur une seule table ayant des champs de type chaîne et/ou de type entier:

SELECT champ₁,...,champ_n FROM table WHERE condition

où condition peut prendre l'une des formes suivantes :

- champ_i comp constante
- (condition oper condition)

avec

- oper $\in \{OR, AND\}$
- comp $\in \{=, >, <\}$
- constante $\in \{\text{'chaîne'}, \text{entier}\}$ où chaîne est une suite finie de lettres ou de chiffres et entier une suite finie de chiffres.
- champ_i et table sont des identificateurs. On prendra pour identificateur, la même définition que dans les TDs

Question 1)

Énoncé de la question

Donner une grammaire permettant d'identifier les différents lexèmes nécessaires pour les requêtes de ce langage SQL.

Solution de la question

// lettre = {a, ..., z, A, ..., Z }

// chiffre = {0, ..., 9 }

Les terminaux : $T = \{ \text{SELECT, FROM, WHERE, OR, AND, ,, ', <, =, >, (,) } \}$

$\cup \text{lettre} \cup \text{chiffre}$

Les non terminaux : $N = \{ \text{lexemeSQL}, \text{paro}, \text{parf}, \text{select}, \text{from}, \text{where}, \text{comparaison}, \text{oper}, \text{identificateur}, \text{constante}, \text{constanteChiffre}, \text{intConstante} \}$

L'axiome : $S = \text{lexemeSQL}$

Règles de production : $P = \{$

Règle : $\text{lexemeSQL} \rightarrow \text{motCle} \mid \text{comparaison} \mid \text{oper} \mid \text{identificateur} \mid \text{constante} \mid \text{virgule}$

Règle : $\text{virgule} \rightarrow ,$

Règle : $\text{identificateur} \rightarrow \text{lettre}$

Règle : $\text{identificateur} \rightarrow \text{identificateur} (\text{lettre} \mid \text{chiffre})$

Règle : $\text{constanteChiffre} \rightarrow \text{chiffre}$

Règle : $\text{constanteChiffre} \rightarrow \text{constantechiffre} \text{ chiffre}$

Règle : $\text{intConstante} \rightarrow \text{lettre} \mid \text{chiffre}$

Règle : $\text{intConstante} \rightarrow \text{intConstante} (\text{lettre} \mid \text{chiffre})$

Règle : $\text{constante} \rightarrow \text{'intConstante'} \mid \text{constanteChiffre}$

Règle : $\text{oper} \rightarrow \text{AND} \mid \text{OR}$

Règle : $\text{comparaison} \rightarrow < \mid = \mid >$

Règle : $\text{select} \rightarrow \text{SELECT}$

Règle : $\text{from} \rightarrow \text{FROM}$

Règle : $\text{where} \rightarrow \text{WHERE}$

Règle : $\text{paro} \rightarrow ($

Règle : $\text{parf} \rightarrow)$

$\}$

Question 2)

Énoncé de la question

En fonction des lexèmes obtenus lors de la question précédente, proposer une grammaire hors contexte qui reconnaît les requêtes syntaxiquement bien formées.

Les lexèmes sont donc les éléments terminaux de cette grammaire

Solution de la question

Les terminaux : $T = \{ \text{select}, \text{from}, \text{identificateur}, \text{virgule}, \text{comparaison}, \text{oper}, \text{constante}, \text{paro}, \text{parf}, \text{where} \}$

Les non terminaux : $N = \{ \text{requete}, \text{listeChamps}, \text{condition} \}$

L'axiome : $S = \text{requete}$

Règles de production : $P = \{$

Règle : $\text{requete} \rightarrow \text{sele} \text{ libelleChamp} \text{ from} \text{ iden} \text{ where} \text{ condition}$

Règle : $\text{listeChamps} \rightarrow \text{identificateur} \mid \text{listeChamps} \text{ virgule} \text{ idenficateur}$

Règle : condition \rightarrow identificateur comparaison constante | paro condition oper condition
parf
}