

Série de révision

Exercice 1 :

1) Calculer l'ensemble de PREMIER pour chaque exemple

G1: $P \rightarrow da$ G2: $B \rightarrow cx \mid bx$

$P \rightarrow Bd$

PREMIER () : renvoie tous les terminaux qu'un non terminal peut commencer

PREMIER (α) : terminaux pouvant commencer une chaîne dérivée de α

- $a \in \text{PREMIER}(\alpha)$ si et seulement si : $\alpha \rightarrow a\beta$ ou β peut être composé de terminaux ou/et non terminaux.
- $\epsilon \in \text{PREMIER}(\alpha)$ si et seulement si : $\alpha \rightarrow \epsilon$
- Si X n'est pas un terminal et $X \rightarrow Y_1, Y_2, Y_3 \dots Y_n$. Avec Y_i symbole terminal ou non terminal alors : ajouter à l'ensemble de PREMIER (X) tous les éléments de PREMIER (Y_i) sauf ϵ .

G1 : $\text{PREMIER}(P) = \{d\}$

G2 : $\text{PREMIER}(B) = \{c, b\}$

$\text{PREMIER}(P) = \{c, b\}$

2) Calculer l'ensemble de SUIVANT pour chaque exemple

G1: $S \rightarrow Ba \mid Sc$ G2: $S \rightarrow Ba \mid Sc$

$B \rightarrow P \mid \epsilon$

$P \rightarrow dS$

SUIVANT(A) : l'ensemble de terminaux pouvant apparaître immédiatement à droite de A dans une dérivation.

L'ensemble de SUIVANT(A) contient :

- Un marqueur de fin de chaîne \$ est ajouté à l'axiome.
- Si la règle $X \rightarrow AaB$: si a est un terminal alors ajouter l'élément a à SUIVANT (A).
- Si la règle $A \rightarrow \alpha B$: alors ajouter l'ensemble SUIVANT(A) à SUIVANT(B).
- Si la règle $A \rightarrow \alpha B\beta$: alors ajouter le contenu de PREMIER(β) à SUIVANT (B)

(Si l'ensemble de PREMIER (β) contient ϵ alors les éléments de SUIVANTS(A) sont ajoutés aussi à SUIVANT(B))

G1 : $\text{SUIVANT}(S) = \{\$, c\}$

G2 : $\text{SUIVANT}(S) = \{\$, a, c\}$

$\text{SUIVANT}(B) = \{a\}$

$$\text{SUIVANT}(P) = \text{SUIVANT}(S) = \{\$, a, c\}$$

Exercice 2 :

Calculer les ensembles PREMIER et SUIVANT de non terminaux et construire la table d'analyse

G: $\text{TYPE} \rightarrow * \text{TYPE} \mid \text{ARRAY}$

$\text{ARRAY} \rightarrow \text{SIMPLE INDICES}$

$\text{INDICES} \rightarrow \text{INDEX INDICES} \mid \varepsilon$

$\text{INDEX} \rightarrow [\text{OPT_DIM}]$

$\text{OPT_DIM} \rightarrow \text{digit} \mid \varepsilon$

$\text{SIMPLE} \rightarrow \text{int} \mid \text{char} \mid (\text{TYPE})$

L'ensemble de PREMIER () :

$\text{PREMIER}(\text{TYPE}) = \{ * \}$

$\text{PREMIER}(\text{ARRAY}) = \{ \text{int}, \text{char}, (\}$

$\text{PREMIER}(\text{INDICES}) = \{ [, \varepsilon \}$

$\text{PREMIER}(\text{INDEX}) = \{ [\}$

$\text{PREMIER}(\text{OPT_DIM}) = \{ \text{digit}, \varepsilon \}$

$\text{PREMIER}(\text{SIMPLE}) = \{ \text{int}, \text{char}, (\}$

L'ensemble de SUIVANT () :

$\text{SUIVANT}(\text{TYPE}) = \{ \$,) \}$

$\text{SUIVANT}(\text{ARRAY}) = \{ \$,) \}$

$\text{SUIVANT}(\text{INDICES}) = \{ [,) , \$ \}$

$\text{SUIVANT}(\text{INDEX}) = \{ [) \$ \}$

$\text{SUIVANT}(\text{OPT_DIM}) = \{] \}$

$\text{SUIVANT}(\text{SIMPLE}) = \{ \$) [\}$

	*	[]	digit	()	int	char	\$
TYPE	TYPE \rightarrow *T TYPE TYPE \rightarrow ARRAY								
ARRAY					ARRAY \rightarrow SIMPLE INDICES		ARRAY \rightarrow SIMPLE INDICES	ARRAY \rightarrow SIMPLE INDICES	
INDICES		INDICES \rightarrow INDEX INDICES	INDICES $\rightarrow \epsilon$			INDICES $\rightarrow \epsilon$			INDICES $\rightarrow \epsilon$
INDEX		INDEX \rightarrow [OPT_DIM]							
OPT_DIM			OPT_DIM $\rightarrow \epsilon$	OPT_DIM \rightarrow digit					OPT_DIM $\rightarrow \epsilon$
SIMPLE					SIMPLE \rightarrow (TYPE)		SIMPLE \rightarrow int	SIMPLE \rightarrow char	

Exercice 3 :

Appliquer l'analyse ascendante par décalage- réduction pour cet exemple : $w = id1 * id2$

$G : E \rightarrow E+T | T$
 $T \rightarrow T * F | F$
 $F \rightarrow (E) | id$

Arbre syntaxique

```

id1 * id2
|   |   |
F   |   |
|   |   |
T   *   F
\   |   /
T
|
E

```

Pile	Entrée	Action
\$	id1*id2\$	Décalage
\$id1	*id2\$	Réduction par $F \rightarrow id$
\$F	*id2\$	Réduction par $T \rightarrow F$
\$T	*id2\$	Décalage
\$T*	id2\$	Décalage
\$T*id2	\$	Réduction par $F \rightarrow id$
\$T*F	\$	Réduction par $T \rightarrow T*F$
\$T	\$	Réduction par $E \rightarrow T$
\$E	\$	Accepter

Exercice 4 :

Générer le code à trois adresses pour le code suivant

```

if (a + b < c) {
    d = e - f;
} else {
    d = g + h;
}

```

```

t1 = a + b
if t1 < c goto label1
t2 = g + h
d = t2
goto label2
label1: t3 = e - f
d = t3
label2:

```

Exercice 5 :

Optimiser ce code après avoir effectué une analyse de vivacité

1)

$\{w, y, z\}$

$x = y + z;$

$\{x, w, y\}$

$z = y + x;$

$\{z, x, w\}$

$y = x * w;$

$\{x, z\}$

$w = z * 2;$

$\{x, w\}$

$z = x - w;$

$\{z\}$

2)

$\{y, z\}$

$x = y + z;$

$\{x, y\}$

$z = y + x;$

$\{x, z\}$

$w = z * 2;$

$\{x, w\}$

$z = x - w;$

$\{z\}$