

**ANALYSEUR SYNTAXIQUE LL(1)**

 1. *Toy grammars*

Considérons les trois grammaires suivantes qui sont simplement là pour l'entraînement à manipuler les algorithmes de suppression de la récursivité à gauche, de factorisation, de calcul des ensembles PREMIER et SUIVANT, et de construction de tables d'analyse.

$$\begin{array}{l}
 G_1 : \quad S \rightarrow T a \mid S a S b \\
 \quad \quad T \rightarrow S b \mid T b \mid b
 \end{array}
 \quad
 \begin{array}{l}
 G_2 : \quad A \rightarrow B a c \mid B b \\
 \quad \quad B \rightarrow A d \mid B c \mid \varepsilon
 \end{array}
 \quad
 \begin{array}{l}
 G_3 : \quad E \rightarrow + E T \mid T \\
 \quad \quad T \rightarrow * T F \mid F \\
 \quad \quad F \rightarrow a
 \end{array}$$

**Exercice 1. Récursivité gauche et factorisation**

- (1) Rendre ces trois grammaires non récursives à gauche et factorisées, si elles ne le sont pas déjà. Vous obtiendrez ainsi les grammaires  $G'_1$ ,  $G'_2$  et  $G'_3$ .

**Exercice 2. Ensembles PREMIER**

Idée générale de l'algorithme de calcul des ensembles PREMIER :

- Étapes d'initialisation :
  - Si  $A \rightarrow \varepsilon$ , ajouter  $\varepsilon$  à  $\text{PREMIER}(A)$
  - Si  $A \rightarrow a \alpha$ , ajouter  $a$  à  $\text{PREMIER}(A)$
- Étape récurrente jusqu'à ce que rien ne soit ajouté :
  - Si  $A \rightarrow \alpha_1 \dots \alpha_n$  :  
ajouter  $\text{PREMIER}(\alpha_1)$  à  $\text{PREMIER}(A)$   
de plus, si  $\varepsilon \in \text{PREMIER}(\alpha_1)$ , ajouter  $\text{PREMIER}(\alpha_2)$  à  $\text{PREMIER}(A)$ ...et ainsi de suite.

- (1) Calculer les ensembles PREMIER pour chacune des grammaires  $G'_1$  et  $G'_2$ .

**Exercice 3. Ensembles SUIVANT**

Idée générale de l'algorithme de calcul des ensembles SUIVANT :

- Étape d'initialisation :
  - Mettre  $\perp$  dans l'ensemble SUIVANT de l'axiome.
  - Si  $A \rightarrow \alpha B \beta$ , ajouter  $\text{PREMIER}(\beta) \setminus \{\varepsilon\}$  à  $\text{SUIVANT}(B)$ .
- Étape récurrente jusqu'à ce que rien ne soit ajouté :
  - Si  $A \rightarrow \alpha B$  ou  $A \rightarrow \alpha B \beta$ , avec  $\varepsilon \in \text{PREMIER}(\beta)$ , ajouter  $\text{SUIVANT}(A)$  à  $\text{SUIVANT}(B)$ .

- (1) Calculer les ensembles SUIVANT pour chacune des grammaires  $G'_1$  et  $G'_2$ .

**Exercice 4. Tables d'analyse LL(1)**

Idée générale de l'algorithme de construction de la table d'analyse LL(1) : Pour chaque  $R_i \in P$  de la forme  $A \rightarrow \alpha$  :

- Pour chaque terminal  $a \in \text{PREMIER}(\alpha)$ , ajouter  $R_i$  à la cellule de ligne  $A$  et de colonne  $a$ , notée  $T(A, a)$ .
- Si  $\varepsilon \in \text{PREMIER}(\alpha)$ , ajouter  $R_i$  à  $T(A, b)$  pour chaque terminal  $b \in \text{SUIVANT}(A)$ . Aussi, si  $\varepsilon \in \text{PREMIER}(\alpha)$  et que  $\perp \in \text{SUIVANT}(A)$ , ajouter  $R_i$  à  $T(A, \perp)$ .

Indiquer un cas d'erreur pour toutes les cellules de  $T$  restées vides.

- (1) Construire la table d'analyse LL(1) de  $G'_1$  et  $G'_2$ . Ces deux grammaires sont-elles LL(1) ?

## 2. LA GRAMMAIRE DU LANGAGE L

**Exercice 5.** *Analyse LL(1)*

On considère la grammaire du langage L suivante, où PG est l'axiome :

(1) PG	→	ODV LDF	(35) IVIDE	→	;
(2) ODV	→	LDV ;	(36) EXP	→	CONJ OOUEXP
(3) ODV	→	$\varepsilon$	(37) OOUEXP	→	EXP
(4) LDV	→	DV LDVB	(38) OOUEXP	→	$\varepsilon$
(5) LDVB	→	, DV LDVB	(39) CONJ	→	NEG OETCONJ
(6) LDVB	→	$\varepsilon$	(40) OETCONJ	→	& CONJ
(7) DV	→	int idv OTT	(41) OETCONJ	→	$\varepsilon$
(8) OTT	→	[ nb ]	(42) NEG	→	! COMP
(9) OTT	→	$\varepsilon$	(43) NEG	→	COMP
(10) LDF	→	DF LDF	(44) COMP	→	E COMPB
(11) LDF	→	$\varepsilon$	(45) COMPB	→	= E COMPB
(12) DF	→	idf LP ODV IB	(46) COMPB	→	< E COMPB
(13) LP	→	( OLDV )	(47) COMPB	→	$\varepsilon$
(14) OLDV	→	LDV	(48) E	→	T EB
(15) OLDV	→	$\varepsilon$	(49) EB	→	+ T EB
(16) I	→	IAFF	(50) EB	→	- T EB
(17) I	→	IB	(51) EB	→	$\varepsilon$
(18) I	→	ISI	(52) T	→	F TB
(19) I	→	ITQ	(53) TB	→	* F TB
(20) I	→	IAPP	(54) TB	→	/ F TB
(21) I	→	IRET	(55) TB	→	$\varepsilon$
(22) I	→	IECR	(56) F	→	( EXP )
(23) I	→	IVIDE	(57) F	→	nb
(24) IAFF	→	VAR = EXP ;	(58) F	→	APPF
(25) IB	→	{ LI }	(59) F	→	VAR
(26) LI	→	I LI	(60) F	→	lire ( )
(27) LI	→	$\varepsilon$	(61) VAR	→	idv OIND
(28) ISI	→	si EXP alr IB OSINON	(62) OIND	→	[ EXP ]
(29) OSINON	→	sin IB	(63) OIND	→	$\varepsilon$
(30) OSINON	→	$\varepsilon$	(64) APPF	→	idf ( LEXP )
(31) ITQ	→	tq EXP fr IB	(65) LEXP	→	EXP LEXPB
(32) IAPP	→	APPF ;	(66) LEXP	→	$\varepsilon$
(33) IRET	→	ret EXP ;	(67) LEXPB	→	, EXP LEXPB
(34) IECR	→	ecr ( EXP ) ;	(68) LEXPB	→	$\varepsilon$

(1) Montrer que la grammaire donnée est LL(1).