

L3  
ULI6BT Théorie des langages et compilation  
durée 2h

Les notes de cours et TD sont autorisées.

*Chaque candidat doit, en début d'épreuve, porter son nom dans le coin de la copie réservé à cet usage; il le cachettera par collage après la signature de la feuille d'émargement. Sur chacune des copies intercalaires, il portera son numéro de place.*

### Exercice I. Automate Fini

Soit le langage fini  $X = \{aba, abba\}$ .

**Question 1.** Construire un automate fini déterministe qui reconnaît  $X$ .

**Question 2.** Décrire en français le langage  $\{a, b\}^*X$ .

**Question 3.** Construire un automate fini non-déterministe qui reconnaît le langage  $\{a, b\}^*X$ .  
Le déterminer.

**Question 4.** À l'aide de l'automate précédent, proposer une stratégie pour analyser un texte  $t$  (c-à-d, une suite quelconque de caractères) et signaler toutes les occurrences des mots  $aba$  et  $abba$  dans le texte  $t$ .

### Exercice II. Grammaire attribuée

Pour commander un robot sur une grille, on lui donne un mot instruction sur l'alphabet  $\{N, S, E, O, [, ], 0, 1, \dots, 9\}$ .

Chacune des lettres code un déplacement élémentaire dans l'une des quatre directions ( $N$  pour nord,  $S$  pour sud,  $\dots$ ). Les crochets servent à encadrer des groupes de lettres, alors que les nombres (i.e., les suites de chiffres) indiquent des répétitions qui s'appliquent au groupe précédent de lettres.

Par exemple, la chaîne  $NS4[NE]12S$  indique de faire  $N$ , 4 fois  $S$ , 12 fois  $NE$  et enfin  $S$ . On autorise aussi les crochets imbriqués comme dans la chaîne  $N[N5E]34[S] [[S]3]5NE$ .

**Question 5.** Écrire une grammaire engendrant l'ensemble des mots instructions valides.

**Question 6.** Indiquer l'arbre d'analyse obtenu sur le mot instruction  $N[[N2E]10S]2$ .

On souhaite maintenant savoir à quel déplacement total correspond un mot instruction (par convention, le déplacement de  $N$  sera  $(0, 1)5$ ,  $S$   $(0, -1)$ ,  $E$   $(1, 0)$  et  $O$   $(-1, 0)$ ).

**Question 7.** Indiquer le déplacement total induit par le mot précédent.

Décorer l'arbre d'analyse du mot en donnant le résultat du calcul obtenu en chacun de ses nœuds.

**Question 8.** Écrire une grammaire attribuée qui permet d'obtenir le déplacement total d'un mot instruction (en code Antrl ou en pseudo-code).

### Exercice III. Analyse LL

Soit la grammaire  $G$  d'axiome  $S$  et de terminaux  $\{id, nb, [, ]\}$  définie par :

$$\begin{cases} S \rightarrow id A \\ A \rightarrow [ B ] A \mid [ B ] \\ B \rightarrow nb \end{cases}$$

**Question 9.** Donner un arbre d'analyse et les dérivations droite et gauche correspondantes, pour le mot suivant : `id[nb][nb]` .

**Question 10.** Pourquoi cette grammaire n'est pas  $LL(1)$  ?

**Question 11.**

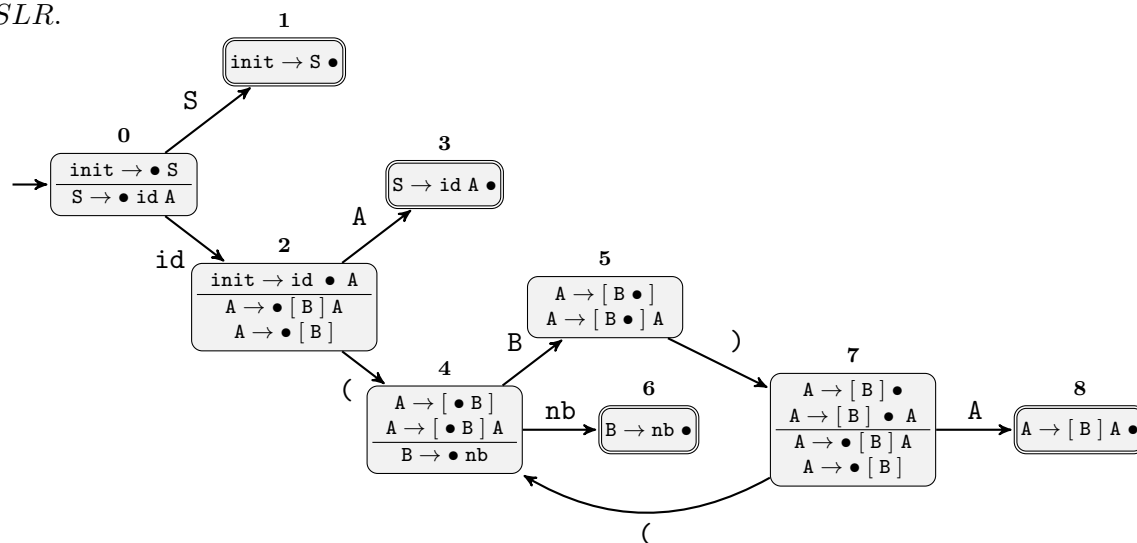
- Proposer une grammaire  $LL(1)$  équivalente à  $G$ .
- Déterminer les variables effaçables. Donner la table des ensembles **Premier** et celle des ensembles **Suivant**.
- Construire la table d'analyse et vérifier ainsi que la grammaire proposée est bien  $LL(1)$ .

#### Exercice IV. Analyse $SLR$

On considère la version augmentée de la grammaire  $G$  précédente :

$$\begin{cases} \text{init} & \rightarrow S \\ S & \rightarrow \text{id } A \\ A & \rightarrow [ B ] A \mid [ B ] \\ B & \rightarrow \text{nb} \end{cases}$$

On donne l'automate fini caractéristique des items  $LR(0)$  de la grammaire  $G$  et sa table d'analyse  $SLR$ .



	\$	id	nb	[	]	S	A	B
0		d 2				1		
1	accepter							
2				d 4			3	
3	r S → id A							
4			d 6					5
5					d 7			
6					r B → nb			
7	r A → [ B ]			d 4			8	
8	r A → [ B ] A							

**Question 12.** Dérouler l'analyse  $SLR$  sur l'entrée `id[nb][nb]` .

**Question 13.** Expliquer de façon claire et détaillée comment la ligne relative à l'état 0 dans la table  $SLR$  est obtenue. Même question pour la ligne de l'état 1 et celle de l'état 7.