TD 10 : Révisions

Pour vos révisions, je vous conseille de travailler les exercices dont je fournis la correction. Il y aura très certainement des grammaires LR / LL / attribuées à l'examen.

1. Langages et Automates

Exercice 1

On considère un automate dont la table de transitions est la suivante :

Etat	A	B	C	D	E	F	G	H	I	J
0	B	C	D	E	F	G	H	I	A	G
1	E	F	H	H	I	B	B	C	E	E
	i		f			f			f	

Question 1 Dessiner l'automate correspondant à la table de transition.

Question 2 Donner un automate minimal équivalent.

2. Grammaires LL et LR

Exercice 1

Soit la grammaire G engendrant des listes, d'axiome L, d'alphabet terminal $V_T = \{\texttt{a}, \texttt{(}, \texttt{)}\}$:

$$\begin{array}{lcl} \text{L} & \to & \text{(S)} \mid \texttt{a} \\ \text{S} & \to & \text{L S} \mid \varepsilon \end{array}$$

Question 1 Construire l'automate LR(0) de la grammaire G.

Question 2 La grammaire G est elle LR(0) ? SLR(1) ? Justifier.

L'état de l'automate qui est en conflit contient les règles:

$$\begin{array}{lcl} \text{S} & \to & \text{L}\bullet\text{S} \\ \text{S} & \to & \text{L}\bullet \\ \text{S} & \to & \bullet\text{LS} \\ \text{S} & \to & \bullet\text{L} \\ \text{L} & \to & \bullet\text{(S)} \\ \text{L} & \to & \bullet\text{()} \\ \text{L} & \to & \bullet\text{a} \end{array}$$

Dans cet état, on sait que le sommet de la pile contient un L. On essaie de voir dans quel cas il est bon de faire une lecture, et dans quel cas il est bon de faire une réduction:

- la réduction va transformer le L en sommet par un S en sommet. C'est donc une bonne opération si le premier caractère en lecture est dans Suivant(S).

- la lecture va poser un caractère terminal par dessus le L. Pour que ca fonctionne, il faut que cette lecture permette d'avancer dans la reconnaissance d'une des règles de notre état (sauf S → L•). Il faut donc que cette lettre soit dans Premier (S) ∪ Premier(LS) ∪ Premier(L) ∪ Premier((S)) ∪ Premier(()) ∪ Premier(a). (c'est à dire $\bigcup$ Premier(x) pour toutes les chaînes x qui sont à droite des • de mon état).

  Remarque: ceci n'est pas égal à Suivant(L). En effet, Suivant(L) inclut Suivant(S) à cause de la règle S → L, et l'ensemble de caractères que l'on obtiendrait inclurait donc ceux qui marchent uniquement après avoir transformé le L en S dans la pile. Ce n'est donc pas ce qu'on cherche là, car on veut justement **garder** L au sommet sans réduire…

  En espérant que ca soit plus clair, n'hésitez pas à me solliciter pour des questions par mail.

Question 3 Détailler le déroulement de l'analyseur SLR(1) pour la grammaire G sur l'entrée `( a ( ) ) $`

On a fait un exemple similaire en cours.

Exercice 2

On considère la grammaire G suivante, d'axiome P', d'alphabet terminal $V_T = \{\texttt{+}, \texttt{-}, \texttt{,}, \texttt{i}\}$:

$$
\begin{array}{lcl}
\text{P'} & \rightarrow & \text{P} \\
\text{P} & \rightarrow & \text{S N} \\
\text{N} & \rightarrow & \text{E}\ \texttt{,}\ \text{I} \mid \text{I} \\
\text{E} & \rightarrow & \text{I} \mid \varepsilon \\
\text{S} & \rightarrow & \texttt{+} \mid \texttt{-} \mid \varepsilon \\
\text{I} & \rightarrow & \texttt{i}
\end{array}
$$

Question 1 Calculer les ensembles *Premier* et *Suivant* nécessaires aux calculs de directeurs.

On n'a besoin de calculer des directeurs que pour les symboles non terminaux ayant plusieurs dérivations possibles. Ici, N, E, et S.

Directeur$(N \rightarrow E\texttt{,}I)$ = Premier$(E\texttt{,}I)$ = Premier$(E) \cup \{\ \texttt{,}\ \}$ = Premier$(I) \cup \{\ \texttt{,}\ \} = \{\ \texttt{i},\ \texttt{,}\ \}$

Directeur$(N \rightarrow I)$ = Premier$(I) = \{\ \texttt{i}\ \}$

Ces deux directeurs ont une intersection non vide, on pourrait donc s'arrêter là et dire que la grammaire n'est pas LL(1). La suite quand même (il faut toujours bien faire attention aux suites de symboles qui peuvent dériver en ε, car dans ce cas il y a plus de choses à rajouter…):

Directeur$(E \rightarrow I)$ = Premier$(I) = \{\ \texttt{i}\ \}$

Directeur$(E \rightarrow \varepsilon)$ = Premier$(\varepsilon) \cup$ Suivant$(E) = \emptyset \cup$ Premier$(\texttt{,}I) = \{\ \texttt{,}\ \}$

Directeur$(S \rightarrow +)$ = Premier$(+) = \{\ \texttt{+}\ \}$

Directeur$(S \rightarrow -)$ = Premier$(-) = \{\ \texttt{-}\ \}$

Directeur$(S \rightarrow \varepsilon)$ = Suivant(S) = Premier(N) = Premier$(E) \cup \{\ \texttt{,}\ \} \cup$ Premier$(I) = \{\ \texttt{,}, \texttt{i}\}$

Question 2 La grammaire est-elle LL(1) ?

Directeur$(N \rightarrow E\texttt{,}I) \cap$ Directeur$(N \rightarrow I) \neq \emptyset$ donc la grammaire n'est pas LL(1).

3. Analyses, grammaires attribuées

Exercice 1

On considère la grammaire suivante pour représenter des nombres en base décimale, octale ou binaire:

$$\begin{array}{lcl} N & \to & LB \\ L & \to & LD \mid D \\ D & \to & 0 \mid 1 \mid \ldots \mid 9 \\ B & \to & b \mid o \mid d \end{array}$$

Question 1 Attribuer la grammaire de manière à calculer la valeur du nombre représenté (en base 10).

Rappel: on peut utiliser à la fois des attributs hérités et des attributs synthétisés. Il faut veiller à ce que chaque attribut ait une définition claire. Il faut donc donner la valeur de chaque attribut (qui dépend de la valeur des autres attributs présents dans la règle).

$N{\uparrow}^{res} \to L{\downarrow}_{base}{\uparrow}^{res} B{\uparrow}^{base}$
le *res* du *N* est égal au *res* de *L*,
et L hérite de la *base* donné par l'attributs synthétisé de *B*

$L{\downarrow}_{base}{\uparrow}^{res} \to L{\downarrow}_{base}{\uparrow}^{x} D{\uparrow}^{chiffre}$
on peut utiliser *chiffre*, x et *base* pour définir *res*: $res = base * x + chiffre$

$L{\downarrow}_{base}{\uparrow}^{res} \to D{\uparrow}^{chiffre}$
on peut utiliser *chiffre*, x et *base* pour définir *res*: $res = chiffre$

$D{\uparrow}^{chiffre} \to 0 \mid 1 \mid \ldots \mid 9$, où *chiffre* vaut la valeur du chiffre
$B{\uparrow}^{2} \to b$
$B{\uparrow}^{8} \to o$
$B{\uparrow}^{10} \to d$

Question 2 La grammaire permet d'écrire par exemple $42b$, ce qui est incorrect puisqu'en base 2 on n'a droit qu'au 0 et au 1. Attribuer la grammaire de manière à pouvoir effectuer ces vérifications.

$$\begin{array}{lcl} N & \to & L{\downarrow}_{base} B{\uparrow}^{base} \\ L{\downarrow}_{base} & \to & L \downarrow_{base} D \uparrow^{chiffre} \mid D \uparrow^{chiffre} \text{ condition}(chiffre < base) \\ D{\uparrow}^{chiffre} & \to & 0 \mid 1 \mid \ldots \mid 9 \\ B{\uparrow}^{2} & \to & b \\ B{\uparrow}^{8} & \to & o \\ B{\uparrow}^{10} & \to & d \end{array}$$

Exercice 2

Un littéral décimal en Ada est défini par :

litteral_decimal ::= entier [.entier] [exposant]
entier ::= chiffre {[trait_bas] chiffre}
exposant ::= E [+] entier | E − entier

Rappel : [e] signifie que e est optionnel ; {e} signifie que e peut être répété un nombre quelconque de fois (éventuellement 0).

Question 1 Donner une grammaire attribuée permettant de calculer la valeur dénotée par un littéral décimal.

Exercice 3

On considère la grammaire des expressions arithmétiques (simples) suivante :

$$
\begin{array}{lcl}
E & \rightarrow & E+F \mid F \\
F & \rightarrow & F*T \mid T \\
T & \rightarrow & (E) \mid id
\end{array}
$$

Attribuer la grammaire pour écrire l'expression sous forme préfixée. Par exemple, la chaîne $2+5*8$ se réécrira en la chaîne $+2*5\ 8$. Vérifier que votre solution est correcte avec $(2+5)*8$, qui devrait donner $*+2\ 5\ 8$.

$$
\begin{array}{lcl}
E\uparrow^{"+".c1.c2} & \rightarrow & E\uparrow^{c1}+F\uparrow^{c2} \\
E\uparrow^{c} & \rightarrow & F\uparrow^{c} \\
F\uparrow^{"*".c1.c2} & \rightarrow & F\uparrow^{c1}*T\uparrow^{c2} \\
F\uparrow^{c} & \rightarrow & T\uparrow^{c} \\
T\uparrow^{c} & \rightarrow & (E\uparrow^{c}) \\
T\uparrow^{"id"} & \rightarrow & id \quad \text{(id représente en fait un nombre)}
\end{array}
$$