Ensimag 1ère année — Mai 2019

Théorie des Langages 2

Durée : 3h. — Documents : tous documents autorisés.

Les 4 exercices sont indépendants 2 à 2. Barème indicatif sur 21 points.

Exercice 1. (3 points) Soit G la BNF suivante d'axiome $\mathbf{S}$ et de terminaux `if`, `a`, `c`, `else`, `fi` :

(1) $\mathbf{S} ::= \mathbf{I}\ \mathbf{S}$	(3) $\mathbf{I} ::= \texttt{a}$	(5) $\mathbf{O} ::= \texttt{else}\ \mathbf{S}\ \texttt{fi}$
(2) $\quad\mid\ \epsilon$	(4) $\quad\mid\ \texttt{if c}\ \mathbf{S}\ \mathbf{O}$	(6) $\quad\mid\ \epsilon$

▷ **Question 1. (2 points)** Calculer les suivants LL(1) de G. On commencera par donner le système d'équations des suivants puis on calculera sa plus petite solution.

▷ **Question 2. (1 point)** La BNF G est-elle LL(1) ? Est-elle ambiguë ? Justifier.

Exercice 2. (6 points) On considère la BNF attribuée suivante, sur le vocabulaire terminal $\{\texttt{a}, \texttt{b}, \texttt{x}\}$, d'axiome $\mathbf{S}{\uparrow}r$ et avec deux autres non-terminaux $\mathbf{U}{\uparrow}r$ et $\mathbf{O}{\downarrow}m{\downarrow}k{\uparrow}r$ pour r, m, k des entiers de $\mathbb{N}$:

$$
\begin{array}{llll}
(1) & \mathbf{S}{\uparrow}r & ::= & \mathbf{U}{\uparrow}r_1\ \mathbf{O}{\downarrow}r_1{\downarrow}1{\uparrow}r \\
(2) & & \mid & \epsilon \qquad\qquad r := 0 \\
(3) & \mathbf{U}{\uparrow}r & ::= & \texttt{a}\ \mathbf{S}{\uparrow}r\ \texttt{b} \\
(4) & & \mid & \texttt{x} \qquad\qquad r := 1 \\
(5) & \mathbf{O}{\downarrow}m{\downarrow}k{\uparrow}r & ::= & \mathbf{U}{\uparrow}r_1\ \mathbf{O}{\downarrow}\max(m, r_1){\downarrow}(k+1){\uparrow}r \\
(6) & & \mid & \epsilon \qquad\qquad r := \max(m, k)
\end{array}
$$

▷ **Question 1. (2 points)** Calculer les directeurs LL(1) de cette BNF. Est-elle LL(1) ?

▷ **Question 2. (1 point)** Dessiner l'arbre d'analyse du mot "`xaxxabxbaxb`" et le décorer avec la propagation d'attributs.

On définit ci-dessous un type `token` où chaque terminal est représenté par un token de même nom, la sentinelle de fin étant appelée `END`. On suppose aussi définie une procédure `parse_token` qui met à jour la variable `current` comme dans le TP2 (`current` devant être initialisé au lancement de l'analyse par un appel à l'analyseur lexical `next`). Enfin, on suppose définie une fonction `max`.

```
typedef enum { a, b, x, END } token;
token current;
token next();

void parse_token(token expected);
int max(int n1, int n2) { return (n1<n2?n2:n1); }
```

▷ **Question 3. (3 points)** Écrire le code C d'une fonction "**int** `parse()`" qui implémente l'analyseur spécifié par la BNF attribuée. On pourra introduire des fonctions auxiliaires.

Exercice 3. (6 points) On considère un langage d'expressions arithmétiques qui est une variante de celui du TP. On réutilise les terminaux du TP suivants : `INT`, `OPAR`, `CPAR` et `MINUS`. On considère également un terminal additionnel `POWER` qui correspond au nouvel opérateur "`^`". Le langage d'expression est défini par l'équation $\mathbf{exp}{\uparrow}r$ avec $r \in \mathbb{R}$ suivante :

$$
\begin{array}{lcll}
\mathbf{exp}{\uparrow}r & ::= & \texttt{INT}{\uparrow}n & r := n \\
& | & \texttt{OPAR}\ \mathbf{exp}{\uparrow}r\ \texttt{CPAR} & \\
& | & \texttt{MINUS}\ \mathbf{exp}{\uparrow}r_0 & r := -r_0 \\
& | & \mathbf{exp}{\uparrow}r_1\ \texttt{MINUS}\ \mathbf{exp}{\uparrow}r_2 & r := r_1 - r_2 \\
& | & \mathbf{exp}{\uparrow}r_1\ \texttt{POWER}\ \mathbf{exp}{\uparrow}r_2 & r := r_1^{r_2}
\end{array}
$$

On désambiguïse cette BNF avec les 3 niveaux de priorités suivantes (pour respecter les conventions mathématiques usuelles) :

- **niveau 2** (le moins prioritaire) associatif à gauche, contenant l'opérateur `MINUS` binaire ;
- **niveau 1** (intermédiaire) associatif à droite, contenant le `MINUS` unaire et `POWER` ;
- **niveau 0** (le plus prioritaire) contenant `INT` et les parenthèses.

▷ **Question 1. (2 points)** Dessiner l'arbre d'analyse (respectant les priorités) de chacun des deux mots suivants : "`1 - 2 ^ 3 - 4 ^ 5`" et "`- 2 ^ - 3 ^ - 4 - 5`"
On ne demande pas de dessiner la propagation d'attributs.

▷ **Question 2. (4 points)** Donner une (E)BNF attribuée LL(1) équivalente à cette BNF avec priorités. Comme vu en TD et en cours, on commencera par encoder les priorités dans la BNF attribuée, puis on la transformera en BNF ou EBNF LL(1) au choix. Justifier le caractère LL(1).

Exercice 4. (6 points) Dans cet exercice, on considère les Machines de Turing (MT) travaillant sur un certain vocabulaire V d'entrée/sortie fixé, et on identifie V^* à $\mathbb{N}$. Une telle MT représente donc une fonction partielle (calculable) de $\mathbb{N} \to \mathbb{N}$.
Soit $< ., . >$ une fonction totale bijective calculable sur $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$; soit un codage des MT comme des entiers de $\mathbb{N}$ et une MT universelle $\mathbf{U}$ qui, sur une entrée $< m, n >$, calcule le résultat éventuel (dans $\mathbb{N}$) de la MT de code m sur l'entrée n. On note "$\mathbf{U}(< m, n >)$" ce processus de calcul.

▷ **Question 1. (2 points)** Chacune des fonctions totales suivantes de $\mathbb{N} \to \mathbb{N}$ est-elle calculable ? Justifier la réponse (en s'appuyant sur les cours et les TDs).

1. Les fonctions $f_{<m,n>} : i \mapsto \begin{cases} 0 \text{ si } \mathbf{U}(< m, n >) \text{ termine en au plus } i \text{ pas d'exécution} \\ 1 \text{ sinon} \end{cases}$
   pour m et n de $\mathbb{N}$ fixés.
2. La fonction $h : < m, n > \mapsto \begin{cases} 0 \text{ si } \mathbf{U}(< m, n >) \text{ termine} \\ 1 \text{ sinon} \end{cases}$

▷ **Question 2. (4 points)** Soit g la fonction (partielle) de $\mathbb{N} \to \mathbb{N}$ telle que
$$\begin{cases} \mathrm{dom}(g) = \{\ m \in \mathbb{N} \mid \exists n \in \mathbb{N}, \mathbf{U}(< m, n >) \text{ termine } \} \\ \forall m \in \mathrm{dom}(g),\quad g(m) = \min(\{\ r \in \mathbb{N} \mid \exists n \in \mathbb{N}, \mathbf{U}(< m, n >) \text{ termine sur le résultat } r\ \}) \end{cases}$$

1. Justifier qu'une telle fonction g existe.
2. Soit le polynôme $P(X) = X^2 - 6X + 14$. Montrer que $\forall n \in \mathbb{N}, P(n) \in \mathbb{N}$.
   Soit m le code dans $\mathbb{N}$ d'une MT calculant P. Montrer $m \in \mathrm{dom}(g)$ et calculer $g(m)$.
3. La fonction g est-elle calculable ? Justifier.
   Indication : on essayera d'appliquer g sur le code des fonctions calculables de la question 1.