

Analyse Syntaxique – DS1

Octobre 2015

Exercice 1. On considère l'automate $M = (A, Q, q_0, F, \delta)$ où $A = \{a, b, c, d\}$, $Q = \{0, 1, 2, 3, 4\}$, $q_0 = 0$, $F = \{2, 4\}$ et δ est déterminée par la table de transition suivante :

δ	a	b	c	d	ε
0	{1}	{1}	\emptyset	\emptyset	{1}
1	\emptyset	\emptyset	{2}	\emptyset	\emptyset
2	\emptyset	\emptyset	{2}	{3, 4}	{4}
3	\emptyset	\emptyset	{4}	\emptyset	\emptyset
4	\emptyset	\emptyset	{4}	\emptyset	\emptyset

Question 1. Dessinez le graphe de cet automate.

Question 2. Déterminez cet automate et dessiner le graphe de l'automate déterministe obtenu.

Question 3. Proposez une expression rationnelle pour le langage reconnu par cet automate.

Exercice 2. Soit la grammaire algébrique $G = (\Sigma, V, E, P)$ avec $\Sigma = \{\&\&, |, id, ?, : \}$, $V = \{E, L, C\}$ et $P = \{E \rightarrow L \mid C, L \rightarrow E \&\& E \mid E | | E \mid id, C \rightarrow L ? E : E\}$.

Question 4. Donnez un arbre de dérivation pour le mot $id ? id \&\& id : id$.

Question 5. Donnez un arbre de dérivation pour le mot $id ? id \&\& id ? id : id : id$. Cette grammaire est-elle ambiguë ?

Exercice 3 (Les nombres en PYTHON3). En PYTHON3, le type `int` permet de représenter les entiers. Un *littéral* (i.e. une constante) `int` peut être donné en base 10 (littéral décimal), en base 2 (littéral binaire), en base 8 (littéral octal) ou en base 16 (hexadécimal) et suit les règles suivantes :

- un `int` peut commencer par un `-` ou un `+`;
- un décimal ne peut pas commencer par `0` (même après un éventuel `-` ou `+`) sauf s'il représente `0`;
- un décimal n'utilise que les chiffres `0, 1, 2, 3, 4, 5, 6, 7, 8` et `9`;
- un binaire commence (après un éventuel `-` ou `+`) par `0b`;

- un binaire n'utilise que les chiffres 0 et 1 ;
- un octal commence (après un éventuel - ou +) par 0o ;
- un octal n'utilise que les chiffres 0, 1, 2, 3, 4, 5, 6 et 7 ;
- un hexadécimal commence (après un éventuel - ou +) par 0x ;
- un hexadécimal n'utilise que les chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9, ainsi que les lettres A, B, C, D, E et F (les lettres peuvent indifféremment être en capitales ou en minuscules).

Voici des exemples de littéraux `int`.

```
0 +42 0x101010 0o52 0x2a -0xfF
```

Question 6. *Donnez une expression rationnelle pour représenter les littéraux décimaux de PYTHON3.*

Question 7. *Donnez une expression rationnelle pour représenter les littéraux `int` de PYTHON3.*

Question 8. *Donnez un automate à états finis reconnaissant les littéraux `int` de PYTHON3.*

Question 9. *Le langage des littéraux `int` de PYTHON3 est-il rationnel ? reconnaissable ? régulier ? algébrique ? Justifiez.*

En PYTHON3, une variable peut avoir pour nom n'importe quelle combinaison de lettres latines (a, b, ..., z) en capitales ou en minuscules, de chiffres ou de `_`, mais ce nom ne doit pas commencer par un chiffre.

Question 10. *Donner une expression rationnelle pour représenter les noms de variables de PYTHON3.*

Une expression arithmétique en PYTHON3 est constituée de nombres littéraux, de noms de variables et d'opérateurs binaires ou unaires. Elle doit être bien parenthésée. Ces opérateurs sont, par ordre de priorité croissant :

- l'addition et la soustraction `+` et `-` ;
- la multiplication `*` et les divisions `/`, `//` et `%` ;
- le passage à l'opposé `-` (qui est un opérateur unaire) ;
- l'exponentiation `**`.

Question 11. *Donnez une grammaire algébrique pour représenter les expressions arithmétiques de PYTHON3. Vous devez utiliser les mots littéraux suivants : {nombre, variable, +, -, *, **, /, //, %, (,) }. Vous n'avez pas à prendre en compte la priorité des opérateurs.*

Question 12. *Donnez une suite de dérivations permettant d'obtenir l'expression arithmétique suivante (dans laquelle `-6.022e23` a été reconnu comme un nombre).*

```
4/3 * pi * r**2 + --6.022e23
```

(Indication : vous devez obtenir `nombre/nombre*variable*variable**nombre+-nombre`.) *Desinez l'arbre de dérivation correspondant.*

Question 13. *Votre grammaire est-elle ambiguë ? Justifiez.*