Compilation - TD 01/02

Stefano Guerrini - Info 3, Sup Galilée, Paris 13

Année 2010/11 - 17/09/2010

Exercice 1. Le but est de lire et vérifier des expressions booléennes construites en utilisant
- les opérateur `&` (la conjonction "et"), `+` (la disjonction "ou") et `~` (la négation "non")
- et qui peuvent contenir des variables et les constant `vrai` et `faux`.

Exemple : `(x & vrai) + ~y & ~(faux + ~alfa)`

1. Donner les unité lexicales qui on trouve dans ces expressions booléen.
2. Donner les modèles de ces expressions booléen
3. Donner la suite d'unités lexicales et de lexèmes qui correspond à l'expression de l'exemple.
4. Donner la structure récursive qui définie les expressions.
5. A partir de la définition récursive donnée au point précédent, dessiner l'arbre associé à l'expression de l'exemple.

Exercice 2. Donnez une expression régulière qui correspond à chacun des langages suivants (définis sur l'alphabet $\Sigma = \{0, 1\}$) :
1. Toutes les chaînes qui se terminent par 00.
2. Toutes les chaînes dont le 10ème symbole, compté à partir de la fin de la chaîne, est un 1.
3. Toutes les chaînes dans lesquelles chaque paire de 0 apparaît devant une paire de 1.
4. Toutes les chaînes ne contenant pas 101.
5. Tous les nombres binaires divisibles par 4.

Définition 1 (Automate fini)**.** Un automate fini est un tuple $A = (Q, \Sigma, \delta, q_0, F)$ tel que :
- Q est une ensemble fini d'états,
- Σ est un alphabet fini, et on note $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$,
- $\delta : Q \times \Sigma_\epsilon \to 2^Q$ est la fonction de transition,
- $q_0 \in Q$, est l'état initial,
- $F \subseteq Q$ est l'ensemble des états accepteurs (ou terminaux).

Définition 2 (chemin)**.** Un chemin σ de longueur $k \geq 1$ dans un automate fini A est une séquence de transitions $\sigma = t_1 t_2 \ldots t_k$, $k \geq 1$ telle que pour tout $1 \leq i \leq k$ on a $t_i = (s_i, a_i, s_{i+1})$ et $s_{i+1} \in \delta(s_i)$.

s_1 est l'origine de σ et s_{k+1} le but de σ.

On écrit souvent un chemin σ sous la forme

$$\sigma = s_1 \xrightarrow{a_1} s2 \xrightarrow{a_3} s_3 \ldots s_k \xrightarrow{a_k} s_{k+1}$$

ou, si $w = a_1 a_2 \ldots a_k$ (qu'on appelle la trace de σ)

$$\sigma = s_1 \xrightarrow{w} s_{k+1}$$

Définition 3 (langage accepté)**.** Un mot $w \in \Sigma$ est accepté par A à partir de q si $q \xrightarrow{w} q'$ et $q' \in F$

Un mot est accepté par A s'il est accepté à partir de q_0.

Le langage $L(q, A)$ accepté par A à partir de $q \in Q$, est l'ensemble des mots acceptés par A à partir de q :

$$L(q, A) = \{w \in \Sigma^* \mid \exists q' \in F \text{ tel que } q \xrightarrow{w} q'\}$$

Le langage accepté par A est $L(q_0, A)$ que l'on note $L(A)$.

Exercice 3. Pour chacune des expression régulières de l'exercice 2, donner une automate fini qui l'accepte.

Exercice 4. Une liste de plaques d'immatriculation françaises et métropolitaines séparées par des virgules et terminée par un point est donnée en entrée d'un programme sous forme d'une chaîne de caractère. Exemple de chaîne :

`684AAA44,4581ZD44,500BAC75,7584VR44,125VF85,5457QD44.`

1. Donnez une expression régulière représentant les chaînes possibles.
2. Dessinez un automate acceptant le langage décrit par cette expression régulière.

Remarque Structure des plaques :

- 1ère partie : un à quatre chiffres ;
- 2ème partie : une à trois lettres ;
- 3ème partie : deux chiffres. Pour la Corse, 2A et 2B (il reste cependant toujours quelques plaques 20).

Exercice 5.

1. Démontrer que tout langage régulière sont acceptés par u automate fini.
2. Démontrer que tout automate fini accepte un langage régulière.