Licence ST Informatique – S4 – 2008/2009

Automates et langages

mars 2009

Résumé du cours 4 : Langages rationnels

1 Généralités sur les langages

1.1 Mot

Définition 1 :

- Un *alphabet* est un ensemble fini de symboles appelés *lettres*.
- Un *mot* sur un alphabet X est une suite finie de lettres de X. Une suite vide de lettres est appelée le *mot vide*, celui-ci est noté : ε.
- La *longueur* du mot m, notée $|m|$ est le nombre de lettres de m.

Définition 2 : Soient m_1 et m_2 deux mots sur l'alphabet X. Si m_1 est la suite de lettres $a_1 \dots a_n$ et m_2 est la suite de lettres $b_1 \dots b_p$ alors le mot $m_1.m_2$ est la suite $a_1 \dots a_n b_1 \dots b_p$. L'opérateur . est appelé opérateur de *concaténation*.

- La concaténation est associative : $(m_1.m_2).m_3 = m_1.(m_2.m_3)$
- ε est l'élément neutre :$\varepsilon.m = m.\varepsilon = m$.
- La concaténation n'est pas commutative : en général, $m_1.m_2 \neq m_2.m_1$.

Définition 3 : Le mot v est *sous-mot* du mot u si $u = a_1 \dots a_n$, $v = a_{i_1} \dots a_{i_k}$, avec $\{i_1, \dots, i_k\} \subseteq \{1, \dots n\}$ et $i_1 < \dots < i_k$.

Définition 4 : v est un *facteur* de u s'il existe des mots u_1 et u_2 tels que $u = u_1.v.u_2$.

- Si $u_1 = \varepsilon$ alors v est un *facteur gauche* ou *préfixe* de u.
- Si $u_2 = \varepsilon$ alors v est un *facteur droit* ou *suffixe* de u.
- Si $v \neq u$ et $v \neq \varepsilon$, alors v est un *facteur propre* de u.

1.2 Langage

Définition 5 : Un *langage* sur un alphabet X est un ensemble de mots sur X.

Exemple 1 *On choisit l'alphabet* $\{0, 1\}$.

- $L_1 = \{00, 01001, 110\}$ *est un langage fini.*
- $L_2 = \{m \mid m$ *représente un multiple de 3 en base 2*$\}$ *est un langage infini.*

On peut étendre l'opérateur de concaténation aux langages :
Définition 6 : Si L_1 et L_2 sont des langages sur l'alphabet X,

$$L_1.L_2 = \{m_1.m_2 \mid m_1 \in L_1 \text{ et } m_2 \in L_2\}$$

Exemple 2 *$L_1 = \{ab, a, \varepsilon\}$ et $L_2 = \{abc, aabb, bc, c\}$.*
$L_1.L_2 = \{ababc, abaabb, abbc, abc, aabc, aaabb, ac, aabb, bc, c\}$

Pour terminer, on peut concaténer un langage avec lui-même, et itérer cette concaténation :
Définition 7 : Si L est un langage sur l'alphabet X,

- $L^0 = \{\varepsilon\}$
- pour $(i > 0)$, $L^i = L^{i-1}.L$

$$L^* = \bigcup_{i \geq 0} L^i$$

L'opérateur $*$ est appelé *étoile de Kleene*.

Notation : on note L^+ le langage $L^*.L$, soit $\bigcup_{i>0} L^i$.

Exemple 3 *Si $L = \{ab, a, \varepsilon\}$, alors $L^0 = \{\varepsilon\}$, $L^1 = L$, $L^2 = \{abab, aba, ab, aab, aa, a, \varepsilon\}$, ...*

2 Les langages rationnels

Ils sont définis à l'aide d'expressions dites *rationnelles*. Voici une définition syntaxique des expressions rationnelles :
Définition 8 : Soit X un alphabet.

- $\emptyset$ est une expression rationnelle,
- a est une expression rationnelle, pour tout $a \in X$
- ε est une expression rationnelle,
- Si e est une expression rationnelle alors (e) et e^* sont des expressions rationnelles,
- si e_1 et e_2 sont des expressions rationnelles, alors $e_1 + e_2$, $e_1.e_2$ sont des expressions rationnelles.

On définit une priorité des opérateurs : * est prioritaire sur . lui-même prioritaire sur +
La sémantique d'une expression rationnelle est un langage :
Définition 9 : Soit X un alphabet.

- $\emptyset$ représente le langage vide,
- a représente le langage $\{a\}$,
- ε représente le langage $\{\varepsilon\}$,
- Si e est une expression rationnelle qui représente le langage L alors e^* représente L^*,

- si e_1 et e_2 sont des expressions rationnelles représentant respectivement les langages L_1 et L_2, alors $e_1 + e_2$ représente $L_1 \cup L_2$ et $e_1.e_2$ représente $L_1.L_2$

Exemple 4 $(a + b + c)^*$ *représente l'ensemble de tous les mots sur l'alphabet* $X = \{a, b, c\}$. *On le note aussi* X^*.
$((a + b).(a + b))^*$ *représente l'ensemble des mots sur l'alphabet* $\{a, b\}$ *qui sont de longueur paire.*

Définition 10 : Un langage est *rationnel* s'il existe une expression rationnelle qui le représente.

2.1 Applications

- Les expressions régulières Unix : Elles ne sont pas plus expressives que les expressions rationnelles (régulier est synonyme de rationnel) mais leur syntaxe est étendue, de façon à rendre les expressions plus simples à écrire. Le premier TP permet de les utiliser avec l'outil `egrep` de recherche de motif. Les expressions régulières sont utilisées dans de nombreux outils, éditeurs de texte, ou langages de programmation.
- Les DTD (Document type definition) : pour chaque type d'élément d'un document XML, on donne une expression rationnelle qui définit le contenu de l'élément.