UNIVERSITÉ de LORRAINE
TELECOM Nancy

TELECOM Nancy	Durée du sujet : 2h
Mathématiques Appliquées pour l'Informatique	$1^{ère}$ année formation par apprentissage
Date : 19 décembre 2017	**Corrigé**
Horaire : 14h à 16h	

Exercice 1 (Théorie des langages : automates et langages réguliers)

1. Nous exprimons le système associé à l'automate :
$$\begin{cases} (1)\ L_p = aL_q + bL_p \\ (2)\ L_q = aL_q + bL_r \\ (3)\ L_r = aL_q + bL_r + \varepsilon \quad (car\ L_r\ est\ terminal) \end{cases}$$
Comme l'état p est le seul état initial de l'automate $\mathcal{A}_1$, le langage $L(\mathcal{A}_1)$, reconnu par $\mathcal{A}_1$ est le langage L_p.
On applique le lemme de Arden aux équations (1) et (2), avec des solutions uniques car $\varepsilon \notin \{a\}$ et $\varepsilon \notin \{b\}$.
$$\begin{cases} (1)\ L_p = b^*aL_q \\ (2)\ L_q = a^*bL_r \\ (3)\ L_r = aL_q + bL_r + \varepsilon \end{cases}$$
On substitue dans (1) et (3), la valeur de L_q prise dans (2).
$$\begin{cases} (1)\ L_p = b^*aa^*bL_r \\ (2)\ L_q = a^*bL_r \\ (3)\ L_r = aa^*bL_r + bL_r + \varepsilon = (aa^*b + b)L_r + \varepsilon \end{cases}$$
On applique enfin le lemme d'Arden à l'équation (3) (cas $\varepsilon \notin (aa^*b + b)$).
$$\begin{cases} (1)\ L_p = b^*aa^*bL_r \\ (2)\ L_q = a^*bL_r \\ (3)\ L_r = (aa^*b + b)^*\varepsilon = (aa^*b + b)^* \end{cases}$$
On substitue la valeur de L_r prise dans (3), dans l'équation (1) et (2).
$$\begin{cases} (1)\ L_p = b^*aa^*b(aa^*b + b)^* \\ (2)\ L_q = a^*b(aa^*b + b)^* \\ (3)\ L_r = (aa^*b + b)^*\varepsilon = (aa^*b + b)^* \end{cases}$$
$b^*aa^*b(aa^*b + b)^*$ est une expression rationnelle dénotant le langage $L(\mathcal{A}_1)$.
2. Les éléments permettant d'affirmer que $\mathcal{A}_2$ est indéterministe sont les suivants :
— une transition sur ε : $(3,\ \varepsilon,\ 0)$
— les transitions suivantes :
  — $(1,\ a,\ 1)$ et $(1,\ a,\ 2)$
Déterminisation.
L'état initial de l'automate déterministe obtenu est l'état constitué de l'ensemble des états atteignables à partir des états initiaux de l'automate indéterministe sans consommer de lettres de l'alphabet $\{a,\ b\}$ (c'est-à-dire les états initiaux de $\mathcal{A}_2$ et ceux qui peuvent être atteints à partir de $\mathcal{A}_2$ par des $\varepsilon-$ *transitions*). Ici l'état initial est $\{0\}$.
Construction de la table intermédiaire :

	a	b
0	$\{1\}$	$\{2\}$
1	$\{1,\ 2\}$	$\{0,\ 3\}$
2	$\{0\}$	$\{0,\ 3\}$
3	$\{1,\ 2\}$	$\{2\}$

Déroulement de l'algorithme : détermination de δ la fonction de transitions de l'automate déterministe. On exécute l'algorithme en partant de l'état initial de l'automate déterministe, c'est-à-dire $\{0,\ 1\}$ et en utilisant la table intermédiaire. Chaque nouvel état généré est mis en entrée (dans la colonne δ de la table).

δ	a	b
$\{0\}$	$\{\mathbf{1}\}$	$\{\mathbf{2}\}$
$\{1\}$	$\{\mathbf{1},\ \mathbf{2}\}$	$\{\mathbf{0},\ \mathbf{3}\}$
$\{2\}$	$\{0\}$	$\{0,\ 3\}$
$\{1,\ 2\}$	$\{\mathbf{0},\ \mathbf{1},\ \mathbf{2}\}$	$\{0,\ 3\}$
$\{0,\ 3\}$	$\{1,\ 2\}$	$\{2\}$
$\{0,\ 1,\ 2\}$	$\{0,\ 1,\ 2\}$	$\{\mathbf{0},\ \mathbf{2},\ \mathbf{3}\}$
$\{0,\ 2,\ 3\}$	$\{0,\ 1,\ 2\}$	$\{0,\ 2,\ 3\}$

L'automate déterministe obtenu est $\mathcal{A}_{d2}$ tel que

$$\mathcal{A}_{d2} = (\{a,\ b\}, Q,\ \{0\},\ \delta,\ \{\{0,\ 2,\ 3\},\ \{1,\ 2\},\ \{0,\ 1,\ 2\},\ \{2\}\})$$

où δ est la fonction de transition définie dans la table ci-dessus, et

$$Q = \ \{\{0,\ 2,\ 3\},\ \{1,\ 2\},\ \{0,\ 1,\ 2\},\ \{0,\ 3\},\ \{2\},\ \{0\},\ \{1\}\}$$

3. Minimisation. Déterminons les états accessibles à partir de l'état initial 0, en utilisant l'algorithme vu en TD, on obtient dans l'ordre : $0,\ 2,\ 4,\ 3,\ 5$, donc 1 et 6 sont des états inaccessibles.
   On exécute l'algorithme :

2	1			
3	0	0		
4	1	1	0	
5	0	0		0
	0	2	3	4

   Donc $3 \sim 5$.
   Les quatre classes obtenues sont les quatre états de l'automate obtenu, $\mathcal{A}_m$, qui est formellement défini comme suit :

$$\mathcal{A}_m = (\{a,\ b\},\ \{\{0\},\ \{2\},\ \{3,\ 5\},\ \{4\}\},\ \{0\},\ \delta_m,\ \{\{3,\ 5\}\})$$

   où δ_m est la table de transition suivante :

δ_m	$\{0\}$	$\{2\}$	$\{3, 5\}$	$\{4\}$
a	$\{4\}$	$\{4\}$	$\{4\}$	$\{3, 5\}$
b	$\{2\}$	$\{3, 5\}$	$\{2\}$	$\{2\}$

Exercice 2 (Analyse syntaxique descendante)

1. Soit la grammaire $G_1 = (\{A, B, C, D, E, F\}, \{a,\ b,\ c\}, \rightarrow,\ A)$ dont les règles sont :

$$\begin{cases} A \rightarrow\ BC \\ B \rightarrow\ aBcD \mid b \\ C \rightarrow\ bC \mid \varepsilon \end{cases} \qquad \begin{cases} D \rightarrow\ aE \mid F \\ E \rightarrow\ aE \mid \varepsilon \\ F \rightarrow\ cFb \mid \varepsilon \end{cases}$$

   $P_\varepsilon = \{C,\ D,\ E,\ F\}$ à cause des règles $C \rightarrow \varepsilon$, $E \rightarrow \varepsilon$, $F \rightarrow \varepsilon$ et $D \rightarrow F$.

   On calcule les premiers et les suivants grâce aux algorithmes vu en cours, ils sont définis dans le tableau suivant :

	A	B	C	D	E	F
Premier	$\{a,\ b\}$	$\{a,\ b\}$	$\{b\}$	$\{a,\ c\}$	$\{a\}$	$\{c\}$
Suivant	$\{\$\}$	$\{b,\ c,\ \$\}$	$\{\$\}$	$\{b,\ c,\ \$\}$	$\{b,\ c,\ \$\}$	$\{b,\ c,\ \$\}$

   Les symboles directeurs des règles se calculent directement en utilisant la définition, ils sont donnés comme suit :

$$SD(A \rightarrow\ BC) = \{a,\ b\}$$

$$\left.\begin{array}{l} SD(B \rightarrow\ aBcD) = \{a\} \\ SD(B \rightarrow\ b) = \{b\} \end{array}\right\} \text{L'intersection des deux ensembles est vide.}$$

$$\left.\begin{array}{l} SD(C \to \ \varepsilon) = \{\$\} \\ SD(C \to \ bC) = \{b\} \end{array}\right\}$$ L'intersection des deux ensembles est vide.

$$\left.\begin{array}{l} SD(D \to \ F) = \{c,\ b,\ \$\} \\ SD(D \to \ aE) = \{a\} \end{array}\right\}$$ L'intersection des deux ensembles est vide.

$$\left.\begin{array}{l} SD(E \to \ \varepsilon) = \{b,\ c,\ \$\} \\ SD(E \to \ aE) = \{a\} \end{array}\right\}$$ L'intersection des deux ensembles est vide.

$$\left.\begin{array}{l} SD(F \to \ \varepsilon) = \{b,\ c,\ \$\} \\ SD(F \to \ cFb) = \{c\} \end{array}\right\}$$ L'intersection des deux ensembles est non vide.

En conclusion il existe deux règles de la forme $A \to \alpha$ et $A \to \beta$ où $\alpha \neq \beta$ pour lesquelles on a $SD(A \to \alpha) \cap SD(A \to \beta) \neq \emptyset$, la grammaire G_1 n'est donc pas LL(1).

2. (a) Calcul des symboles directeurs :

i. $$SD(F \to \ CQ) = \{p,\ q,\ r,\ \neg\}$$

ii. $$\left.\begin{array}{l} SD(Q \to \varepsilon) = \{\$\} \\ SD(Q \to \vee CQ) = \{\vee\} \end{array}\right\} \textit{l'intersection des deux ensembles est vide}$$

iii. $$SD(C \to \ LD) = \{p,\ q,\ r,\ \neg\}$$

iv. $$\left.\begin{array}{l} SD(D \to \varepsilon) = \{\$,\ \vee\} \\ SD(D \to \wedge LD) = \{\wedge\} \end{array}\right\} \textit{l'intersection des deux ensembles est vide}$$

v. $$\left.\begin{array}{l} SD(L \to V) = \{p,\ q,\ r\} \\ SD(L \to \neg V) = \{\neg\} \end{array}\right\} \textit{l'intersection des deux ensembles est vide}$$

vi. $$\left.\begin{array}{l} SD(V \to p) = \{p\} \\ SD(V \to q) = \{q\} \\ SD(V \to r) = \{r\} \end{array}\right\} \textit{les intersections des ensembles pris deux à deux sont vides}$$

Les calculs précédents (toutes les instersections considérées sont vides) montrent que la grammaire G_2 est LL(1).

(b) Table d'analyse de la grammaire G_2.

	$\vee$	$\wedge$	$\neg$	p	q	r	$\$$
F			$F \to \ CQ$	$F \to \ CQ$	$F \to \ CQ$	$F \to \ CQ$	
Q	$Q \to \vee CQ$						$Q \to \varepsilon$
C			$C \to LD$	$C \to LD$	$C \to LD$	$C \to LD$	
D	$D \to \varepsilon$	$D \to \wedge LD$					$D \to \varepsilon$
L			$L \to \neg V$	$L \to V$	$L \to V$	$L \to V$	
V				$V \to p$	$V \to q$	$V \to r$	

(c) Analyse du mot $m_1 = \neg p \wedge q$.

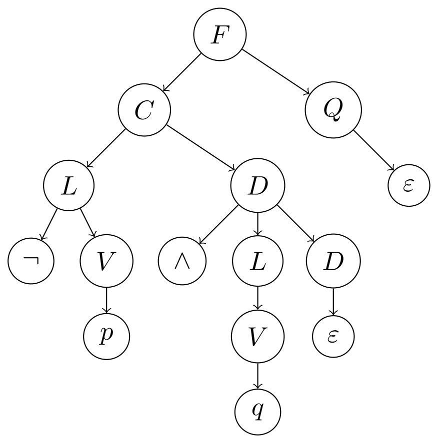
$PILE$	$Entrée$	$Sortie$
$\$F$	$\neg p \wedge q\$$	$F \to CQ$
$\$QC$	$\neg p \wedge q\$$	$C \to LD$
$\$QDL$	$\neg p \wedge q\$$	$L \to \neg V$
$\$QDV\neg$	$\neg p \wedge q\$$	
$\$QDV$	$p \wedge q\$$	$V \to p$
$\$QDp$	$p \wedge q\$$	
$\$QD$	$\wedge q\$$	$D \to \wedge LD$
$\$QDL\wedge$	$\wedge q\$$	
$\$QDL$	$q\$$	$L \to V$
$\$QDV$	$q\$$	$V \to q$
$\$QDq$	$q\$$	
$\$QD$	$\$$	$D \to \varepsilon$
$\$Q$	$\$$	$Q \to \varepsilon$
$\$$	$\$$	*succes*

Le mot m_1 appartient au langage $L(G_2)$, engendré par G_2.

Dérivation à gauche du mot m_1 en partant de l'axiome F :

$F \rightarrowtail CQ \rightarrowtail LDQ \rightarrowtail \neg VDQ \rightarrowtail \neg pDQ \rightarrowtail \neg p \wedge LDQ \rightarrowtail \neg p \wedge VDQ$
$\rightarrowtail \neg p \wedge qDQ \rightarrowtail \neg p \wedge qQ \rightarrowtail \neg p \wedge q$

Arbre syntaxique :



Analyse du mot $m_2 = p \vee \wedge r$.

$PILE$	$Entrée$	$Sortie$
$\$F$	$p \vee \wedge r\$$	$F \to CQ$
$\$QC$	$p \vee \wedge r\$$	$C \to LD$
$\$QDL$	$p \vee \wedge r\$$	$L \to V$
$\$QDV$	$p \vee \wedge r\$$	$V \to p$
$\$QDp$	$p \vee \wedge r\$$	
$\$QD$	$\vee \wedge r\$$	$D \to \varepsilon$
$\$Q$	$\vee \wedge r\$$	$Q \to \vee CQ$
$\$QC\vee$	$\vee \wedge r\$$	
$\$QC$	$\wedge r\$$	*erreur*

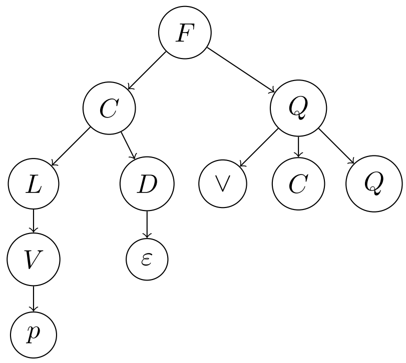
On a ici une erreur, car dans la table d'analyse la case correspondant au couple c(C, $\wedge$) est vide.

Le mot m_2 n'appartient pas au langage $L(G_2)$ engendré par G_2.

Début de la dérivation à gauche du mot m_2 en partant de l'axiome F :

$F \rightarrowtail CQ \rightarrowtail LDQ \rightarrowtail VDQ \rightarrowtail pDQ \rightarrowtail pQ \rightarrowtail p \vee CQ \rightarrowtail erreur$

Arbre syntaxique incomplet :



Exercice 3 (Logique des propositions)

1. Les hypothèses f_i $(1 \leq i \leq 5)$ et la conclusion γ s'écrivent de la façon suivante :
   — $f_1 \ : \ (c \wedge v) \Rightarrow e$
   — $f_2 \ : \ \neg c \Rightarrow f$
   — $f_3 \ : \ \neg v \Rightarrow m$
   — $f_4 \ : \ \neg e$
   — $f_5 \ : \ s \Rightarrow (\neg f \wedge \neg m)$
   — $\gamma \ : \ \neg s$
2. Mise sous forme clausale des formules f_1, f_2, f_3, f_4 et f_5. On utilise l'équivalence $a \Rightarrow b \ \equiv \ \neg a \vee b$, les lois de de Morgan, la distributivité de $\vee$ par rapport à $\wedge$ et l'équivalence $\neg\neg a \equiv a$.
   — $f_1 = (c \wedge v) \Rightarrow e = \neg(c \wedge v) \vee e = \neg c \vee \neg v \vee e$, donc $C(f_1) = \{\neg c \vee \neg v \vee e\}$
   — $f_2 = \neg c \Rightarrow f = \neg(\neg c) \vee f = c \vee f$, donc $C(f_2) = \{c \vee f\}$.
   — $f_3 = \neg v \Rightarrow m = \neg(\neg v) \vee m = v \vee m$, d'où $C(f_3) = \{v \vee m\}$.
   — $C(f_4) = \{\neg e\}$.
   — $f_5 = s \Rightarrow (\neg f \wedge \neg m) = \neg s \vee (\neg f \wedge \neg m) = (\neg s \vee \neg f) \wedge (\neg s \vee \neg m)$, d'où $C(f_5) = \{\neg s \vee \neg f, \neg s \vee \neg m\}$.
   On note
   $c_1 = \neg c \vee \neg v \vee e$,
   $c_2 = c \vee f$,
   $c_3 = v \vee m$,
   $c_4 = \neg e$,
   $c_5 = \neg s \vee \neg f$,
   $c_6 = \neg s \vee \neg m$
   La valuation δ, telle que

δ	c	e	f	m	s	v
	1	0	1	1	0	0

   est un modèle de l'ensemble des clauses $\{c_1,\ c_2,\ c_3,\ c_4,\ c_5,\ c_6\}$, l'ensemble des formules $\{f_1,\ f_2,\ f_3,\ f_4,\ f_5\}$ est donc non contradictoire.
3. On a $C(\neg\gamma) = \{s\}$, notons $c_7 = s$. On considère les clauses, c_i, pour $1 \leq i \leq 7$.
   La résolution entre c_1 et c_4 produit : $c_8 \ : \ \neg c \vee \neg v$.
   La résolution entre c_7 et c_5 produit : $c_9 \ : \ \neg f$.
   La résolution entre c_7 et c_6 produit : $c_{10} \ : \ \neg m$.
   La résolution entre c_3 et c_{10} produit : $c_{11} \ : \ v$.
   La résolution entre c_2 et c_9 produit : $c_{12} \ : \ c$.
   La résolution entre c_8 et c_{12} produit : $c_{13} \ : \ \neg v$.
   La résolution entre c_{13} et c_{11} produit : $c_{14} \ : \ \Box$.
   On vient de montrer que $\{c_i,\ 1 \leq i \leq 7\}$ est contradictoire, on a donc $\{c_i,\ 1 \leq i \leq 6\} \models \neg s$, on a donc démontré que Superman n'existe pas.