Corrigé Épreuve de contrôle de Mathématiques Discrètes (consolidation juin 2006)

Question de cours :

1°) Donnez la définition d'une grammaire algébrique et la signification des signes utilisés dans cette définition. Réponse en moins de 10 lignes.

Une grammaire algébrique est un quadruplet $G = (N, T, \rightarrow, X)$ tel que N et T sont deux vocabulaires disjoints, le premier est le vocabulaires auxiliaire et le second le vocabulaires terminal. La flèche $\rightarrow$ est une relation entre N et $(N \cup T)^*$ telle que pour chaque A de N, il n'y ait qu'un nombre fini de α tel que $A \rightarrow \alpha$.. Enfin, le symbole X est un élément particulier de N appelé axiome.

2°) Donnez la définition d'un automate fini indéterministe et la signification des signes utilisés dans cette définition. Réponse en moins de 10 lignes.

Un automate fini indéterministe est un quadruplet A = (S, s_0, P, S') tel que S est un ensemble fini appelé ensemble de états de l'automate, s_0 est un état particulier appelé "état initial", le symbole P représente un sous-ensemble de $S \times V$ Y $\{\wedge\} \times S$ où V est le vocabulaire sur lequel travaille l'automate et S' est un sous-ensemble de S appelé ensemble des états de satisfaction. Un élément (s, x, s') est dans l'ensemble P si et seulement si l'automate étant dans l'état s et lisant x sur sa bande d'entrée (c'est-à-dire ne lisant rien si $x = \wedge$), l'automate est susceptible de passer dans l'état s'.

3°) Dans un algorithme d'analyse syntaxique ascendante, dans un état de l'automate SLR(1) d'une grammaire G, on a les deux items $A \rightarrow \alpha.a\beta$ et $B \rightarrow \alpha'.$. Quel type de conflit potentiel y a-t-il dans la ligne de la table action correspondant à cet état ? (Réponse en trois mots). Comment fera-t-on pour tenter de résoudre ce conflit et quand sera-t-il résolu ?

Il y a un potentiel conflit lecture-réduction. On calcule les voisins de B ; si a n'est pas dans les voisins de B, alors le conflit est levé. Si le caractère lu en avance est a, on effectue une lecture, si le caractère lu en avance est un voisin de B, on effectue la réduction correspondant à la règle $B \rightarrow \alpha'$. Dans les autres cas, on passe en erreur.

Test d'algèbre de Boole, manipulation d'expressions arithmétiques et logique :

1°) Veuillez donner le résultat de l'évaluation de la fonction booléenne suivante pour les données ci-dessous.

Fonction booléenne : $f(x, y, z, t) = [(y \Rightarrow z) \wedge (t \text{ NOR } x)] \vee [(y \Leftrightarrow x) \wedge (z \Leftarrow t)]$

Données : (0, 0, 1, 0) (0, 1, 1, 1) (1, 1, 0, 1).

1°) On trouve facilement que

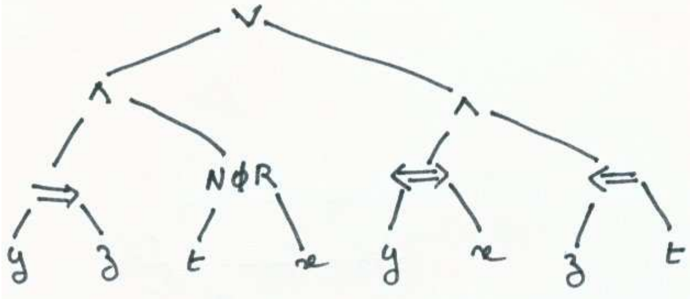
$f(0, 0, 1, 0) = [(0 \Rightarrow 1) \wedge (0 \text{ NOR } 0)] \vee [(0 \Leftrightarrow 0) \wedge (1 \Leftarrow 0)] = 1$

$f(0, 1, 1, 1) = [(1 \Rightarrow 1) \wedge (1 \text{ NOR } 0)] \vee [(1 \Leftrightarrow 0) \wedge (1 \Leftarrow 1)] = 0$

$f(1, 1, 0, 1) = [(1 \Rightarrow 0) \wedge (1 \text{ NOR } 1)] \vee [(1 \Leftrightarrow 1) \wedge (0 \Leftarrow 1)] = 0$

2°) On considère la fonction booléenne de la question précédente. Veuillez en écrire l'arbre abstrait, l'écriture préfixée et l'écriture postfixée.

Arbre abstrait :



Écriture préfixée :
$\vee \wedge \Rightarrow y\ z$ NOR $t\ x \wedge \Leftrightarrow y\ x \Leftarrow z\ t$

Écriture postfixée :
$y\ z \Rightarrow t\ x$ NOR $\wedge\ y\ x \Leftrightarrow z\ t \Leftarrow \wedge \vee$

3°) En utilisant les calculs faits à la question 1°) et en considérant cette fonction comme l'écriture d'une formule du calcul des propositions, dites si l'on peut répondre aux questions suivantes :
- Est-ce que cette formule est une tautologie ? Dites pourquoi.
- Est-ce que cette formule est contradictoire ? Dites pourquoi.

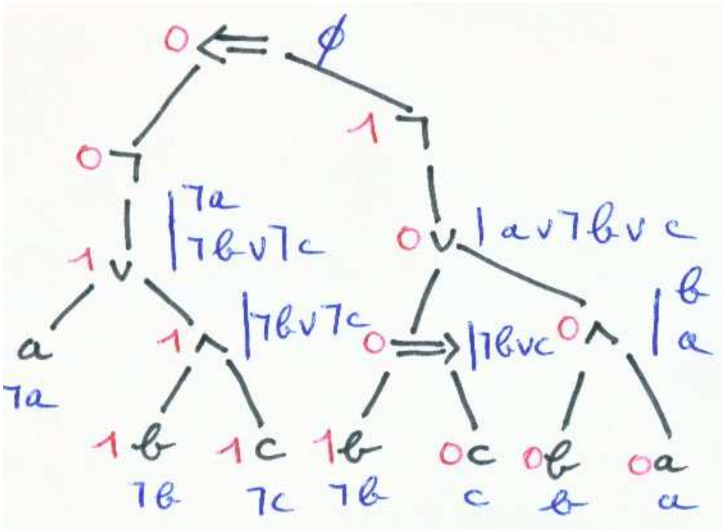
3°) Comme $f(1, 1, 0, 1) = [(1 \Rightarrow 0) \wedge (1 \text{ NOR } 1)] \vee [(1 \Leftrightarrow 1) \wedge (0 \Leftarrow 1)] = 0$, la formule n'est pas toujours vraie. Ce n'est donc pas une tautologie (formule toujours vraie).
$f(0, 1, 0, 0) = [(0 \Rightarrow 1) \wedge (0 \text{ NOR } 0)] \vee [(0 \Leftrightarrow 0) \wedge (1 \Leftarrow 0)] = 1$, la formule n'est pas toujours fausse et donc elle n'est pas contradictoire (formule toujours fausse).

Test de logique des propositions :

Montrez que la formule α suivante est une tautologie en utilisant la méthode travaillant sur l'arbre abstrait et décorant cet arbre avec diverses valeurs (nombres de négations, ensembles de clauses).
NB : seules les solutions utilisant la méthode imposée seront acceptées même si d'autres méthodes donnent le même résultat éventuellement plus vite.

$$\alpha = \neg[a \vee (b \wedge c)] \Leftarrow \neg[(b \Rightarrow c) \vee (b \wedge a)]$$

Calcul des clauses associées à la formule α ci-dessus :



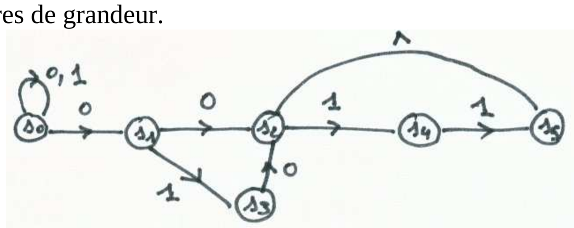
Test de théorie des langages (automate fini) :

1°) Trouvez un automate fini indéterministe qui reconnaît le langage L suivant :

$$L = \{0, 1\}^* \{00, 010\} \{11\}^+.$$

Indication : l'auteur de l'énoncé a résolu le problème avec un automate indéterministe à 7 états et qui est rendu déterministe avec aussi 7 états. Vous devriez avoir les mêmes ordres de grandeur.

1°) Voilà un automate possible ; ce n'est pas le seul, mais il semble bien adapté et lisible. Il ne comporte d'ailleurs que 6 états et pas 7 comme dans un premier essai maladroit. Le seul état de satisfaction est s_5.



2°) Appliquez la méthode qui permet de trouver un automate fini déterministe équivalent à celui de la question 1°).

2°) On construit déjà la table intermédiaire qui ne pose pas de problème particulier, puis la table de l'automate déterministe qui simule l'automate indéterministe. Dans le second tableau, il y a les états sous la forme de sous-ensembles des états et avec des nouveaux noms (q_i)

	0	1
s_0	$s_0\ s_1$	s_0
s_1	s_2	s_3
s_2	-	s_4
s_3	s_2	-
s_4	-	$s_5\ s_2$
s_5	-	s_4

Tableau intermédiaire

		0	0	1	1
s_0	q_0	$s_0\ s_1$	q_0	s_0	q_1
$s_0\ s_1$	q_1	$s_0\ s_1\ s_2$	q_2	$s_0\ s_3$	q_3
$s_0\ s_1\ s_2$	q_2	$s_0\ s_1\ s_2$	q_0	$s_0\ s_3\ s_4$	q_3
$s_0\ s_3$	q_3	$s_0\ s_1\ s_2$	q_4	s_0	q_3
$s_0\ s_3\ s_4$	q_4	$s_0\ s_1\ s_2$	$\boldsymbol{q_5}$	$s_0\ s_2\ \boldsymbol{s_5}$	q_3
$s_0\ s_2\ \boldsymbol{s_5}$	$\boldsymbol{q_5}$	$s_0\ s_1$	q_6	$s_0\ s_4$	q_1
$s_0\ s_4$	q_6	$s_0\ s_1$	$\boldsymbol{q_5}$	$s_0\ s_2\ \boldsymbol{s_5}$	q_1

Automate déterministe. L'état initial est q_0. Le seul état de satisfaction est $\boldsymbol{q_5}$ et est en gras.

Un calcul rapide montre que cet automate est minimal et ne peut pas être amélioré.

3°) Veuillez trouver l'automate fini déterministe équivalent à l'automate suivant :

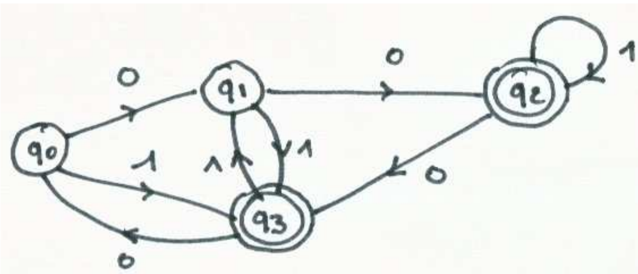
état	0	1
s_0	s_1	s_5
s_1	s_4	s_6
s_2	s_8	s_3

état	0	1
s_3	s_7	s_1
s_4	s_3	s_4
s_5	s_2	s_8

état	0	1
s_6	s_7	s_1
s_7	s_1	s_6
s_8	s_4	s_3

L'état initial est évidemment s_0. Les états de satisfaction sont s_3, s_4, s_5 et s_6.
NB : La première itération devrait vous suffire pour trouver les états de l'automate minimum.
Tracez le graphe du nouvel automate obtenu.

3°) On initialise l'algorithme avec deux paquets d'états (ceux de non-satisfaction et ceux de satisfaction), c'est-à-dire : (s_0, s_1, s_2, s_7, s_8) et (s_3, s_4, s_5, s_6).
La première itération fait 4 paquets qui sont : (s_0, s_2, s_7), (s_1, s_8), (s_3) et (s_4, s_5, s_6) . Un rapide calcul montre que l'itération suivante ne recoupe pas de paquet et donc le résultat est obtenu. On appelle respectivement ces paquets q_0, q_1, q_2 et q_3. Les deux états q_2 et q_3 sont de satisfaction car formés uniquement d'états de satisfaction et l'état initial est q_0 car il contient l'ancien état initial s_0.
Le dessin de cet automate est :



Test d'analyse syntaxique (analyse syntaxique descendante et ascendante) :

1°) On considère la grammaire algébrique suivante (les règles sont numérotées). On souhaite faire une analyse syntaxique descendante de cette grammaire.

Axiome : X $N = \{X, A\}$ $T = \{$**(**, **;**, **)**, ***a***$\}$

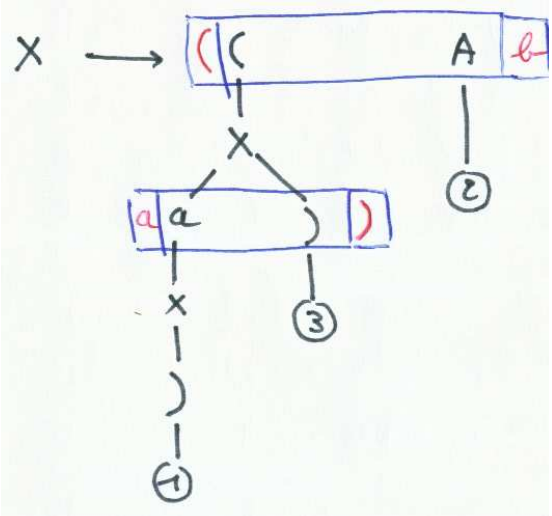
Règles : $X \rightarrow$ **(** X ***a*** X **)** |①| A |②| **(** X **)** |③|

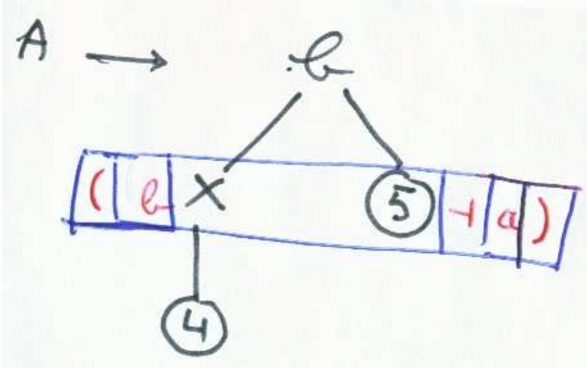
$A \rightarrow$ ***b***X |④| ***b*** |⑤|

a) Mettez cette grammaire sous forme arborescente et désignez par des cadres les endroits qui nécessiteront des tests dans les procédures.

b) Mettez en place les tests nécessaires à une analyse syntaxique descendante en lisant un caractère à l'avance.

1°) a) et b)





c) Écrivez le programme principal et les procédures d'analyse syntaxique, y compris celle associée à l'analyse d'un terminal.

c) Les tests portant en début de chaque procédure sur le booléen *erreur* ont été omis dans un but de plus grande lisibilité.

Programme principal :

Res ← ∧ ; *erreur* ← *faux* ; lire(*y*) ;
Ana-*X* ;
<u>Si</u> *erreur* <u>alors</u> écrire("*le mot n'est pas engendré par la grammaire*")
<u>sinon</u> <u>si</u> $y \neq \dashv$ <u>alors</u> écrire("*le mot n'est pas engendré par la grammaire*")
<u>sinon</u> écrire("*le mot est engendré par la grammaire*", *res*)
<u>fsi</u>
<u>fsi</u>

Procédure Ana(*x* : *T*) ;
<u>si</u> *y* = *x* <u>alors</u> { *res* ← *res*⊕*x* ; lire(*y*) } <u>sinon</u> *erreur* ← *vrai* <u>fsi</u>

Procédure Ana-*X* ;
<u>cas</u> *y* = **(** → Ana(**(**) ; Ana-*X* ; <u>cas</u> *y* = ***a*** → Ana(***a***) ; Ana-*X* ; *res* ← *res*⊕① ;
y = **)** → Ana(**)**) ; *res* ← *res*⊕③ ;
autre cas → *erreur* ← *vrai*
<u>sac</u>
y = ***b*** → Ana-*A* ; *res* ← *res*⊕②
autre cas → *erreur* ← *vrai*
<u>sac</u>

Procédure Ana-*A* ;
Ana(***b***) ;
<u>cas</u> *y* = **(** ***b*** → Ana-*X* ; *res* ← *res*⊕④ ;
y = ***a*** **)** ⊣ → *res* ← *res*⊕⑤
autre cas → *erreur* ← *vrai*
<u>sac</u>

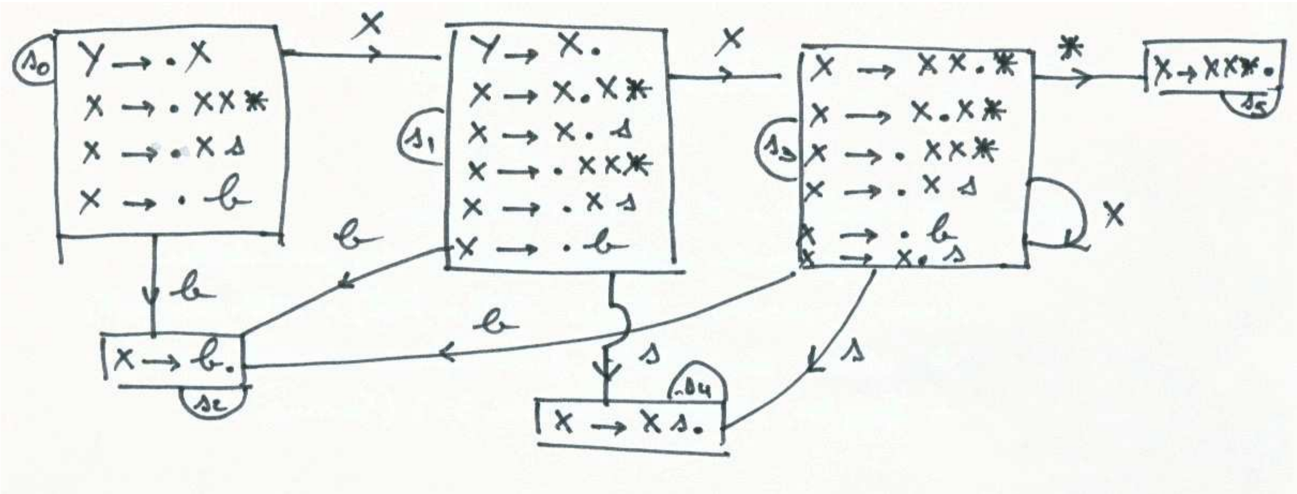
2°) On considère la grammaire algébrique suivante (les règles sont numérotées). On souhaite faire une analyse syntaxique ascendante de cette grammaire par la méthode SLR(1).
Axiome : *X* Vocabulaire auxiliaire : *N* = {*X*} Vocabulaire terminal *T* = {×, ***b***, ***s***}
Règles : *X* → *X X* × | ① | *X* ***s*** | ② | ***b*** | ③ |
a) Il faut avant toute chose ajouter une règle à cette grammaire. Laquelle ?

b) Quel est l'état initial de l'automate SLR qui contrôlera l'analyse ?
c) Construisez cet automate et définissez la table action.

a) Il faut ajouter la règle $Y \to X$ où le nonterminal Y devient le nouvel axiome de la grammaire (augmentation de la grammaire fournie ; opération faite dans le but de mieux gérer l'arrêt sur *succès* du programme d'analyse).

b) L'état initial est $s_0 = \{Y \to \mathbf{.}\, X,\ X \to \mathbf{.}\, X\, X \times,\ X \to \mathbf{.}\, X\, \boldsymbol{s},\ X \to \mathbf{.}\, \boldsymbol{b}\}$

c) L'automate est fourni par la figure suivante :



La table d'automate et d'action est la suivante :

	X	$\times$		$\boldsymbol{s}$		$\boldsymbol{b}$		$\dashv$
	état	*état*	*action*	*état*	*action*	*état*	*action*	*action*
s_0	s_1	-		-		s_2	Lecture	-
s_1	s_3	-		s_4	Lecture	s_2	Lecture	***succès***
s_2	-	-	Réduction ③	-	Réduction ③	-	Réduction ③	Réduction ③
s_3	s_3	s_5	Lecture	s_4	Lecture	s_2	Lecture	-
s_4	-	-	Réduction ②	-	Réduction ②	-	Réduction ②	Réduction ②
s_5	-	-	Réduction ①	-	Réduction ①	-	Réduction ①	Réduction ①

Seul l'état s_1 peut conduire à un conflit lecture réduction. On cherche les voisins de Y et l'on trouve uniquement le marqueur de fin $\dashv$. Cela rend cette ligne de la table d'action déterministe.

Les autres états sont soit purement de lecture (s_1 et s_3) soit purement de réduction (s_2, s_4 et s_5). Les réductions se font sur les voisins de X qui sont $\times$, $\boldsymbol{s}$, $\boldsymbol{b}$ et $\dashv$.

FIN DU CORRIGÉ