

Grammaires algébriques

Licence info S5
TD COMPIL – 2011-2012

Exercice 1 : Des a et des b

On considère la grammaire algébrique $G = (V_T, V_N, A, P)$ avec $V_T = \{a, b\}$, $V_N = \{X\}$ et $P = \{X \rightarrow aXb \mid XX \mid bXa \mid \epsilon\}$.

Q 1.1 : Donner tous les mots de $L(G)$ de longueur 0, 2, 4. □

Q 1.2 : Quel est le langage engendré par G ? □

Q 1.3 : Soit le mot $w = aabbbaab$. Donner pour w , une dérivation droite, une dérivation gauche et un arbre syntaxique. □

Q 1.4 : G est-elle ambiguë? Justifier. □

Exercice 2 : Encore des a et des b

Pour chacun des langages ci-dessous, donner une grammaire algébrique qui l'engendre :

- $\{a^n b^n \mid n \geq 0\}$;
- langage des palindromes sur $\{a, b\}$ (mots qui se lisent de la même manière de gauche à droite et de droite à gauche) ;
- $\{a^p b^n c^n a^p \mid n, p \geq 1\}$;
- $\{a^n b^p \mid n \geq p \geq 0\}$;
- langage des mots bien parenthésés par a et b , contenant par exemple ϵ , $aabbab$, $aabb$.

Exercice 3 : Un langage de commandes

Lors du TD1 on a réfléchi à un analyseur lexical pour le langage de commandes suivant :

- une *commande* est composée d'un nom de commande, suivi d'une liste optionnelle d'arguments, suivie d'une liste facultative d'options ;
- une *liste d'arguments* est une suite d'arguments ;
- une *liste d'options* est une suite non vide d'options encadrée par [et], à l'intérieur de laquelle les options sont séparées par , ;
- une *option* est un caractère précédé d'un tiret ;
- un *argument* est un identificateur, de même qu'un *nom de commande*.

Par exemple, `macom arg1 arg2 [-a, -b]` est une commande, ainsi que `macom [-f]` et `macom`.

Q 3.1 : Donner une grammaire algébrique qui décrit ce langage de commande. Préciser quels sont ses terminaux, ses non-terminaux et son axiome. □

Q 3.2 : Le langage décrit est-il régulier? Justifier. □

On envisage d'enrichir la syntaxe d'une commande pour ses options : une option est soit un caractère comme précédemment soit une liste d'options. Par exemple `macom [-a, [-b, [-c]], [[-d]]]` est maintenant une commande correcte (comment ça il n'est pas réaliste ce langage de commandes?).

Q 3.3 : Modifier la grammaire de la question 1 pour prendre en compte ces modifications. □

Q 3.4 : Le langage décrit (après modification) est-il régulier? Justifier. □

Exercice 4 : Description de volumes

On s'intéresse à la description textuelle de volumes contenant une arborescence de répertoires et fichiers. La description commence par le mot-clé **contentFor** suivi du nom du volume (une suite de caractères). Ensuite on trouve la liste des fichiers et/ou répertoires contenus dans le volume. Un fichier est représenté par son nom. La description d'un répertoire commence par un crochet ouvrant suivi du nom du répertoire suivi du caractère : suivi de la description du contenu du répertoire, suivie d'un crochet fermant. Le contenu du répertoire est décrit par la liste de ses éléments. Un répertoire contient des fichiers et/ou d'autres répertoires. Les noms de répertoire et de fichier sont des suites de caractères (des identificateurs à la Java). Un volume et un répertoire peuvent être vides de contenu. Voila un exemple de description :

```
contentFor USBkey
README [machin : ] toto
[Desktop : [tpCompil : Makefile tp.tex tp.dvi] photo.jpg ]
dump
```

Q 4.1 : Donner les productions d'une grammaire qui engendre l'ensemble des descriptions de volume. Entourer l'axiome et souligner les terminaux. ☐

Q 4.2 : Le langage engendré par cette grammaire est-il régulier ? Justifier en au plus 2 lignes. ☐

Exercice 5 : Expressions régulières

On souhaite décrire les expressions régulières parenthésées sur $\{a,b\}$ ne contenant pas le mot vide. Le choix entre deux expressions sera dénotée par $+$ (pour ne pas confondre avec le choix $|$ des grammaires algébriques). La concaténation est implicite (pas de symbole de concaténation).

Q 5.1 : Écrire une grammaire à opérateur G (intuitive, et par là sans doute ambiguë) décrivant ces expressions régulières, en explicitant l'ensemble des terminaux et l'ensemble des non-terminaux. ☐

Q 5.2 : Justifier l'ambiguïté de cette grammaire. ☐

Q 5.3 : Donner un arbre syntaxique pour le mot $a+b*a$ et le mot $(a+b)*(ab)$. Ces mots sont-ils ambigus ? Justifier. ☐

Q 5.4 : Sachant que l'opérateur $*$ est prioritaire sur la concaténation, elle même prioritaire sur l'opérateur $+$, et que les opérateurs binaires sont associatifs à gauche, donner une grammaire non ambiguë G' équivalente à celle de la question 1. ☐

Q 5.5 : Donner à nouveau un arbre syntaxique pour le mot $a+b*a$, cette fois pour G' . ☐

Exercice 6 : Expressions arithmétiques post-fixées

Soit la grammaire $G = (V_T, V_N, E, P)$ avec $V_N = \{E\}$, $V_T = \{+, -, *, /, i\}$ et $P = \{E \rightarrow E E + \mid E E * \mid E E - \mid E E / \mid i\}$

Q 6.1 : Donner un arbre syntaxique pour les mots $ii+i*$ et $iii++$. ☐

Q 6.2 : Donner un mot dont l'interprétation infixée est $(i+i)+i$, ainsi qu'un arbre syntaxique pour ce mot. ☐

Q 6.3 : Cette grammaire est-elle ambiguë ? ☐

Exercice 7 : Conditionnelles

On considère la grammaire des conditionnelles suivantes, dont l'ensemble des terminaux est `{if, else, statNoIf}` :

$stat \rightarrow statNoIf \mid if\ stat \mid if\ stat\ else\ stat$

Q 7.1 : Montrer que cette grammaire est ambiguë. □

Q 7.2 : Donner une grammaire non ambiguë équivalente. □

Pour les curieux, consulter la spécification du langage Java, paragraphe 14.5 et 14.9.

http://java.sun.com/docs/books/jls/second_edition/html/statements.doc.html#101241

Exercice 8 : Expressions régulières

On reprend les expressions régulières de l'exercice 5, et on considère la grammaire d'axiome E :

$E \rightarrow choix$

$choix \rightarrow choix + concat \mid concat$

$concat \rightarrow etoile concat \mid etoile$

$etoile \rightarrow X * \mid X$

$X \rightarrow a \mid b \mid (E)$

Q 8.1 : Cette grammaire n'engendre pas les expressions régulières habituelles. Dites pourquoi et justifier par des exemples. □

Q 8.2 : Que se passe-t-il si on remplace $etoile \rightarrow X *$ par $etoile \rightarrow E *$? □

Exercice 9 : Propriétés de clôture des langages algébriques

Q 9.1 : Montrer que l'union / le produit de 2 langages algébrique est un langage algébrique, de même que l'étoile d'un langage algébrique.

Idée : construire une grammaire algébrique à partir de grammaires algébriques qui engendrent les langages opérands. □

Q 9.2 : Montrer que les langages algébriques ne sont fermés ni par intersection ni par complémentaire.

Idées :

- Pour l'intersection chercher à obtenir le langage contre-exemple $\{a^n b^n c^n \mid n \geq 0\}$ à partir de deux langages algébriques qui n'imposent pas l'égalité du nombre de a , b et c , mais seulement de deux d'entre eux, du genre $\{a^n b^n c^p \mid n \geq 0, p \geq 0\}$.
- Pour le complémentaire : partir des propriétés de clôture déjà prouvées et raisonner par l'absurde. □

Exercice 10 : Recettes de cuisine

J'aimerais stocker sur ordinateur mes recettes de cuisine préférées. Voilà un exemple de recette :

Riz carottes lentilles

Recette végétarienne facile et pas chère, pratique quand le réfrigérateur est vide

Nombre de part : 2

Temps de préparation : 10mn

Temps de cuisson : 20mn

*Coût : £ Difficulté : **

Ingrédients : 2 carottes, 2 oignons, 1/2 tasse riz, 1/2 tasse lentilles, sel, poivre, curry, huile d'olive, algues

Émincer les oignons et les faire revenir dans un peu d'huile d'olive. [...] Ajouter les lentilles, attendre 12 minutes, saler en fin de cuisson.

Astuces : délicieux avec de la sauce soja

Servir avec : de l'eau ?

Remarques : si pas d'algues, rallonger la cuisson des lentilles

Déjà dans le placard : riz, lentilles, huile, curry, sel, poivre, algues

J'ai écrit la grammaire algébrique suivante :

- les terminaux sont : { TEXTE, COUT, POURIEN, PASDONNE, DISPENDIEUX, TEMPS, DUREEPREPA, DUREECUISSON, DUREEREPOS, SERVIRAVEC, ASTUCES, REMARQUES, INGREDIENTS, DANSPLACARD, DIFFICULTE, INRATABLE, MOYEN, DIFFICILE, NBPARTS, ENTIER, VIRG } ;
- les non-terminaux sont : { recette, titre, preambule, ingredients, description, postambule, commentaire, nbParts, durees, cout, difficulte, tempsPreparation, tempsCuisson, tempsRepos, lecout, listeIngredients, ingredient, astuces, remarques, dejaServiA, dansMonPlacard, tache, diff } ;
- l'axiome est *recette* ;
- les productions sont les suivantes :

```

recette → titre preambule ingredients description postambule
preambule → commentaire nbParts durees cout difficulte
durees → tempsPreparation tempsCuisson
tempsPreparation → DUREEPREPA TEMPS
tempsCuisson → DUREECUISSON TEMPS | ε
tempsRepos → DUREEREPOS TEMPS | ε
cout → COUT lecout
lecout → POURIEN | PASDONNE | DISPENDIEUX
durees → tempsPreparation tempsCuisson
tempsPreparation → DUREEPREPA TEMPS
tempsCuisson → DUREECUISSON TEMPS
tempsRepos → DUREEREPOS TEMPS
ingredients → INGREDIENTS listeIngredients
listeIngredients → ingredient VIRG listeIngredients
ingredient → TEXTE
postambule → astuces remarques dansMonPlacard
astuces → ASTUCES TEXTE
servirAvec → SERVIRAVEC TEXTE
dansMonPlacard → listeIngredients
description → tache description | tache
tache → TEXTE
difficulte → DIFFICULTE diff
diff → INRATABLE | MOYEN
nbParts → NBPARTS ENTIER

```

Q 10.1 : Que pensez-vous de cette grammaire algébrique ?

□

Exercice 11 : Réduction de grammaire

Q 11.1 : Réduire la grammaire $G = (V_T, V_N, S, P)$ avec $V_T = \{a, b, c\}$, $V_N = \{S, A, B, C, D, E\}$ et P contient les productions :

```

S → AB | aS | b          C → cC | cE | c
A → BD | CD              D → DD | DC
B → bB | b | Db          E → aSa

```

□