

Correction du Partiel THL

THÉORIE DES LANGAGES

EPITA – Promo 2013
Avec formulaire de QCM, sans documents ni machine

Juin 2011 (1h30)

Correction: Le sujet a été écrit par Akim Demaille et Jonathan Fabrizio.

Barème: Se reporter à la feuille de calcul pour les coefficients des questions.

Bien lire les questions, chaque mot est important.

Répondre sur les formulaires de QCM; aucune réponse manuscrite ne sera corrigée. Renseigner les champs d'identité. Il y a exactement une et une seule réponse juste pour ces questions. Si plusieurs réponses sont valides, sélectionner la plus restrictive. Par exemple s'il est demandé si 0 est *nul*, *non nul*, *positif*, ou *négatif*, cocher *nul* qui est plus restrictif que *positif* et *négatif*, tous deux vrais. Répondre incorrectement est plus pénalisé que de ne pas répondre.

1 Incontournables

Chaque erreur ou non réponse aux trois questions suivantes retire 1/6 de la note finale. Avoir tout faux divise donc la note par 2.

Q.1 Si A, B et N sont des langages rationnels, alors le langage $ANBN$ est rationnel.

✓ vrai ✗ faux

Correction: La concaténation est une des opérations rationnelles : *par définition* elle « préserve » la rationalité.

Q.2 Soit L_1 et L_2 deux langages rationnels. Si $L_1 \subset L \subset L_2$ alors L est rationnel.

✗ vrai ✓ faux

Correction: Tout langage L vérifie $\emptyset \subset L \subset \Sigma^*$, même ceux qui ne sont pas rationnels.

Q.3 Si un langage L peut être défini par une grammaire linéaire à gauche, alors il est reconnaissable par un automate à états fini déterministe.

✓ vrai ✗ faux

Correction: Les grammaires linéaires à gauche (et les grammaires linéaires à droite) définissent les langages rationnels.

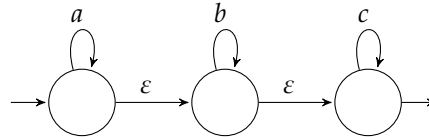
2 Contrôle

Q.4 L'expression rationnelle étendue $[-+]*[0-9A-E]^+([-+/*] [-+]*[0-9A-E]^+)^*$, n'engendre pas :

✗ ++00+++11 ✗ +-11-+22*+23*-1024DD
✓ -DEAD+BEEF ✗ A1A+B2B-C3C*D4D

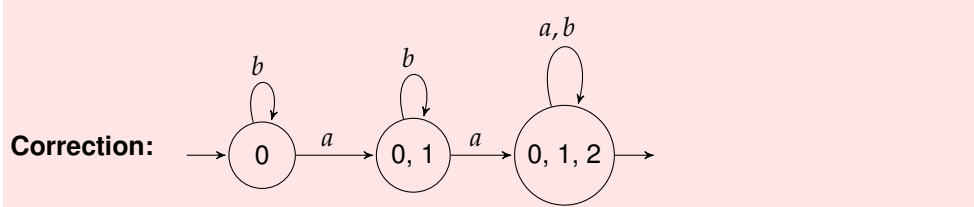
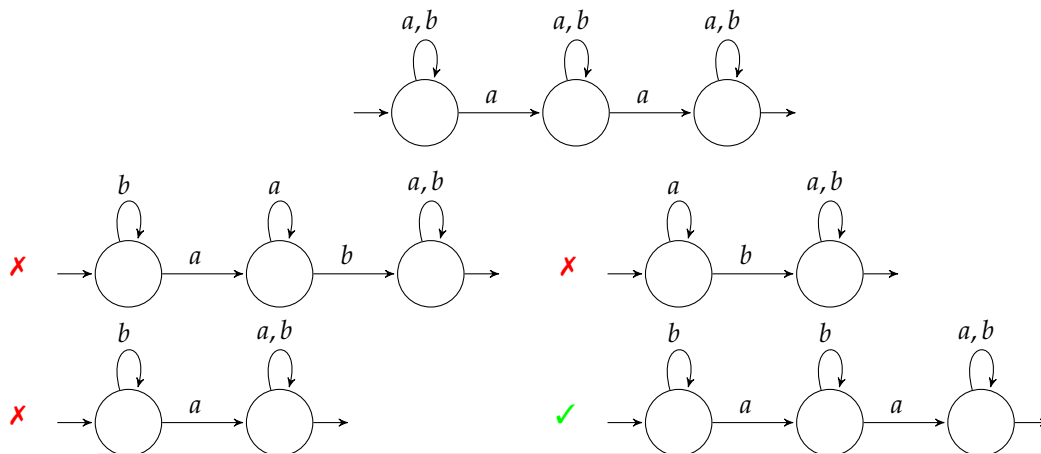
Correction: Il n'est pas possible d'avoir un F.

Q.5 L'automate suivant est :



- déterministe à états spontanés
- non-déterministe à états spontanés
- déterministe à transitions spontanées
- non-déterministe à transitions spontanées
- indéterministe à états et transitions spontanés

Q.6 Déterminiser l'automate suivant.



Q.7 Quelle séquence permet de calculer un automate déterministe à partir d'une expression rationnelle ?

- Norton, Brzozowski et McCluskey, Kleene
- Thompson, ε-élimination arrière, déterminisation
- Thompson, déterminisation, Brzozowski et McCluskey
- Thompson déterminisé
- Norton, déterminisation, ε-élimination avant

Q.8 La grammaire suivante est . .

$$S \rightarrow aZb \mid ab$$

$$Z \rightarrow aSb$$

- linéaire à gauche
- linéaire au centre
- rationnelle
- hors contexte

Q.9 Le langage engendré par la grammaire précédente est

- indéfini
- rationnel
- hors catégorie
- hors contexte

Q.10 La grammaire suivante est . .

$$S \rightarrow SpS \mid n$$

- Linéaire à droite
- Linéaire à gauche
- Rationnelle
- Ambiguë

Q.11 Si une grammaire est LL(1), alors

- elle n'est pas rationnelle
- elle est rationnelle
- elle n'est pas ambiguë
- elle est ambiguë

Q.12 Si une grammaire hors contexte est non ambiguë

- elle est LL(k)
- elle est LL(1)
- elle n'est pas nécessairement LL
- elle produit nécessairement des conflits dans un parseur LL

Q.13 Un parseur LL(k) est un parseur :

- left-left
- ambigu
- top-down
- bottom-up

Q.14 Un parseur LL(k) . . .

- privilégie l'opération de *shift* lors d'un conflit *shift/reduce*
- fait une lecture en une passe de gauche à droite, avec k symboles de regard avant
- fait k lectures de gauche à droite
- est équivalent à un automate à états fini

Q.15 Une grammaire LL(1) . . .

- ne permet pas de faire de la reprise sur erreur
- permet facilement l'écriture d'un parseur à la main
- engendre un langage rationnel
- est intrinsèquement plus coûteuse à analyser qu'une grammaire LL(2)

Q.16 Il existe des grammaires SLR(1) qui ne sont pas . . .

- LR(2)
- LR(1)
- LALR(1)
- LR(0)

Q.17 Yacc génère un parseur

- LL
- Look Ahead Left-to-right, Rightmost-derivation
- GLR
- LALLR

Q.18 GLR permet l'analyse

- de toutes les grammaires hors contexte, même ambiguës
- de toutes les grammaires hors contexte non ambiguës
- de toutes les grammaires LR(1), même ambiguës
- de toutes les grammaires LR

Q.19 Dans une analyse classique en utilisant Yacc et Lex :

- on appelle `yylex` plusieurs fois, puis `yyparse` une fois
- on appelle `yyparse` plusieurs fois, elle appelle `yylex` chaque fois
- on appelle `yyparse` une fois, elle appelle `yylex` plusieurs fois
- on appelle `yyparse(yylex())` plusieurs fois

Q.20 Soit la grammaire suivante¹ traitée par Yacc :

```
exp: "a" | "a";
```

- Elle n'a pas de conflit.
- Elle présente un conflit *shift/reduce*.
- Elle présente un conflit *reduce/reduce*.
- Elle présente un conflit *shift/shift*.
- Elle présente deux conflits *shift/reduce*.

Q.21 Soit la grammaire des expressions booléennes suivante. Combien de conflits présente le parseur LALR correspondant.

1. Dans cette grammaire Yacc/Bison et les suivantes, les `%%` ne sont pas écrits.

Q.28 L'enseignant

- a. N'est pas pédagogue
- b. Parle à des étudiants qui sont au dessus de mon niveau
- c. Me parle
- d. Se répète vraiment trop
- e. Se contente de trop simple et devrait pousser le niveau vers le haut