

Correction du Partiel THL

THÉORIE DES LANGAGES

EPITA – Promo 2013 – **Sans documents ni machine**

Décembre 2010 (1h30)

Correction: Le sujet et sa correction ont été écrits par Akim Demaille et Jonathan Fabrizio.

Barème: Toutes les questions sont notées sur 4, et c'est ainsi qu'il faut lire les barèmes qui suivent. Se reporter à la feuille de calcul pour voir les coefficients donnés aux questions.

Best-of: Le best-of est tiré des copies des étudiants. Les erreurs, en particulier d'orthographe, sont les leurs ! Les commentaires du correcteur sont notés [*ainsi*].

Bien lire les questions, chaque mot est important. Écrire court, juste, et bien. Une argumentation informelle mais convaincante est souvent suffisante.

Répondre aux questions à choix multiples (numérotées Q.1, Q.2 etc.) sur les formulaires de QCM; aucune réponse manuscrite ne sera corrigée. Renseigner les champs d'identité. Sauf mention contraire, 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.

1 Incontournables

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

Q.1 L'intersection de deux langages rationnels est rationnelle.

✓ vrai

✗ faux

Correction: Oui. Le model-checking repose sur cette propriété.

Q.2 Toute partie d'un langage rationnel est rationnelle.

✗ vrai

✓ faux

Correction: Bien sûr que non. Tout langage (comme $a^n b^n$ bien connu pour ne pas être rationnel) est partie de Σ^* , qui est rationnel.

Q.3 Toute grammaire qui présente des conflits dans un parseur LR(1) est une grammaire ambiguë.

✗ vrai

✓ faux

Correction: Non. Comme on le voit d'ailleurs dans l'exercice 3.

2 Contrôle

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

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

Correction: Il n'est pas possible d'avoir une étoile après le moins.

Q.5 Qui est l'inventeur de LL(1) et LR(1) ?

- ✗ Frank DeRemer ✗ Noam Chomsky ✓ Donald Knuth ✗ Stephen Kleene

Correction: Donald Knuth est bien l'inventeur d'une part de la famille LR, et d'autre part des appellations LR et LL. En revanche, contrairement à ce qui vous a été dit en cours, il n'est pas l'inventeur de LL.

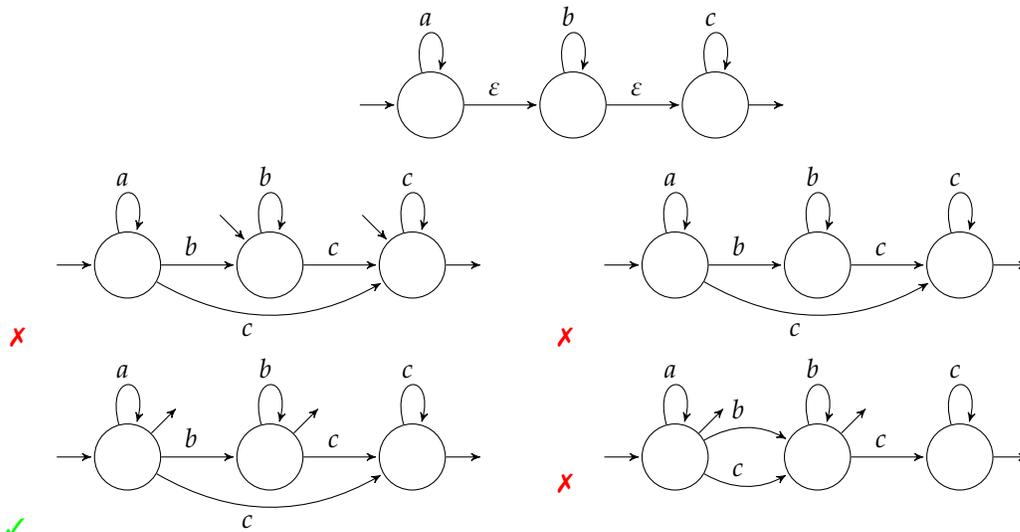
Q.6 Combien existe-t-il de nombres octaux à quatre chiffres ?

- ✗ 12^2 ✗ 12^4 ✓ 2^{12} ✗ 8^{12}

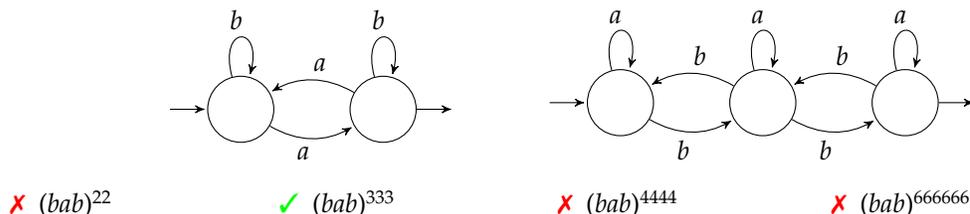
Correction: $8^4 = (2^3)^4 = 2^{3 \times 4} = 2^{12} = 4096$.

Best-of: Dire que « 0 », « 1 », ..., « 777 » ont des nombres octaux à quatre chiffres me gêne un peu (heureusement que « $2^{12} - 2^9$ » n'est pas dans les réponses). En revanche, dire qu'ils peuvent être représentés avec « 1000 », ..., « 7777 » sur 12 bits me paraît correct. [Bien sûr que « 0 », « 1 », ..., « 777 » ne sont pas des nombres octaux de quatre chiffres ; en revanche « 0000 », « 0001 », ..., « 0777 » le sont.]

Q.7 Quel est le résultat d'une ϵ -clôture arrière sur l'automate suivant ?

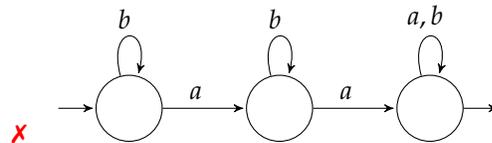
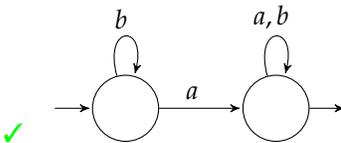
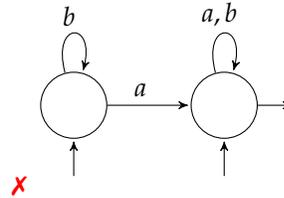
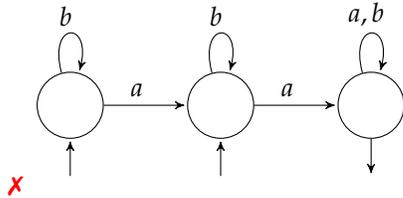
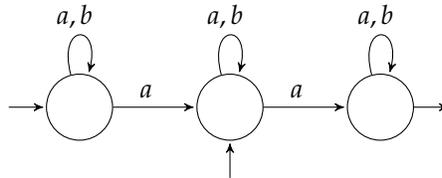


Q.8 Quel mot est reconnu par l'automate produit des deux automates suivants ?

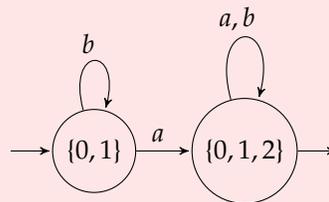


Correction: L'automate produit calcule l'intersection des langages. Le premier automate veut un nombre impair de a , le second un nombre pair non nul de b . Donc, évidemment, (b) .

Q.9 Déterminer l'automate suivant.



Correction: Ne pas oublier qu'il faut commencer par un état initial qui représente tous les états initiaux du nondéterministe.



Best-of: Rien ne vaut de bons schémas. [Si, répondre sur le formulaire de qcm. Ça évite aussi les fautes d'orthographe.]

Q.10 Quelle définition de l'arithmétique traditionnelle est bien adaptée à l'écriture d'un parseur LL(1)? (Ci-dessous les parenthèses représentent les caractères eux-mêmes; les accolades appartiennent à la notation, et signifient "zéro ou plusieurs fois", i.e., $\{e\} = e^*$).

- | | |
|--|--|
| $\begin{aligned} E &\rightarrow E + E \mid E - E \\ \times &\quad \mid E \times E \mid E / E \\ &\quad \mid (E) \mid n \\ E &\rightarrow T + E \mid T - E \mid T \\ \times & T \rightarrow F \times T \mid F / T \mid F \\ & F \rightarrow (E) \mid n \end{aligned}$ | $\begin{aligned} E &\rightarrow E + T \mid E - T \mid T \\ \times & T \rightarrow T \times F \mid T / F \mid F \\ & F \rightarrow (E) \mid n \\ E &\rightarrow T \{+ T \mid - T\} \\ \checkmark & T \rightarrow F \{\times F \mid / F\} \\ & F \rightarrow (E) \mid n \end{aligned}$ |
|--|--|

Correction: (a) est ambiguë, (b) n'est pas LL parce que (par exemple) les FIRST des trois règles de E sont égaux, pareil pour (c) qui de surcroît présente des récursions gauches.

Q.11 Pour toute grammaire linéaire à gauche,

- \times il n'existe pas nécessairement d'automate qui reconnaisse son langage
- \checkmark il existe un automate déterministe qui reconnaisse son langage
- \times il existe un automate non-déterministe qui reconnaisse son langage
- \times il existe un automate non-déterministe à transition spontanée qui reconnaisse son langage

Correction: Une grammaire linéaire à gauche engendre un langage rationnel, pour lequel il existe un automate qui le reconnaisse. On peut le demander déterministe (éventuellement au prix d'une élimination des transitions spontanées, puis d'une déterminisation), si bien que (b) est la bonne réponse, puisqu'elle est plus restrictive que (c) et (d), vraie également.

Q.12 Dans une analyse classique en utilisant Yacc et Lex :

- \checkmark on appelle yyparse une fois, elle appelle yy1ex plusieurs fois ;
- \times on appelle yyparse(yy1ex()) plusieurs fois ;
- \times on appelle yy1ex plusieurs fois, puis yyparse une fois ;
- \times on appelle yyparse plusieurs fois, elle appelle yy1ex chaque fois.

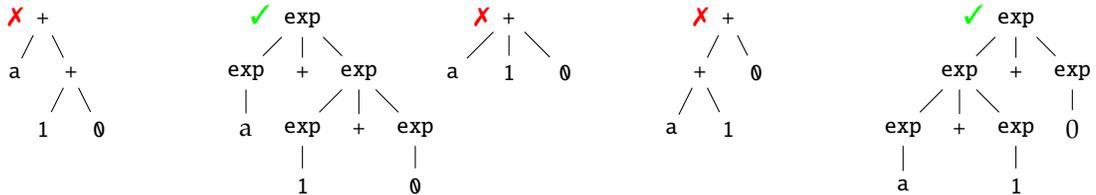
3 Parsage LALR(1)

On étudie la possibilité d'une implémentation en Bison d'un parseur pour la grammaire des expressions rationnelles suivante.

```
%%
exp:
  "0" // Empty language.
| "1" // Empty word.
| "a" // For each letter "a" of the alphabet.
| exp "+" exp // Alternation.
| exp "." exp // Concatentation.
| exp "*" exp // Kleene star.
| "(" exp ")" // Grouping.
;
```

« a » représente une lettre de l'alphabet (c'est un terminal). Lorsque l'on parle des opérateurs (infixes) binaires, on se réfère à « + » et « . ».

Q.13 Montrer deux arbres de dérivation de a + 1 + 0.



Correction: L'énoncé comportait une erreur (un 1 à la place du 0 dans (e)), si bien que seront considérées correctes les réponses (b) et (be).

Q.14 Cette grammaire est :

- ambiguë
- infinie
- rationnelle
- LL(2)

Correction: La question précédente rendait impardonnable toute erreur à celle-ci.

Q.15 Les opérateurs binaires sont pris associatifs à gauche. Les priorités des opérateurs sont, dans l'ordre croissant, $e + f$, puis $e.f$, puis e^* .

Quelle forme parenthésée correspond à $a * + a * + a \cdot a^*$?

- $((a^*) + ((a^*) + (a \cdot (a^*))))$
- $((((a^*) + (a^*)) + (a \cdot (a^*))))$
- $((((a^*) + a)^*) + (a \cdot (a^*)))$
- $((((a^*) + (a^*)) + ((a \cdot a)^*))$

Q.16 Quelles directives implantent ces contraintes pour les opérateurs binaires ?

<p><input checked="" type="checkbox"/> <code>%left "."</code> <code>%left "+"</code></p> <p><input checked="" type="checkbox"/> <code>%left "." "+"</code></p> <p><input checked="" type="checkbox"/> <code>%right "+" "."</code></p>	<p><input checked="" type="checkbox"/> <code>%prec "+" < "."</code></p> <p><input checked="" type="checkbox"/> <code>%left "+"</code> <code>%left "."</code></p>
---	---

Les opérateurs postfixes sont inhabituels. Étudions les conflits qu'ils engendrent afin de pouvoir les résoudre correctement.

- Montrer que la règle de l'opérateur postfixe est impliquée dans deux conflits shift/reduce avec les autres opérateurs en exhibant la situation dans laquelle se trouve alors le parseur.

Correction: Dans la situation suivante, deux calculs sont possibles :

```
⊢ exp "+" exp      "*" ⊢
```

La réduction donne priorité à +, le décalage à l'étoile. Il en est de même dans le cas · versus l'étoile.

Best-of:

- [...] Le shift est cette fois en conflit avec * la règle 5 : $exp \rightarrow exp".exp$.
* ** un reduce utilisant. . .
[Puis, en note de bas de page :]
** Est-ce que l'opérateur "*" (astérisque — note de bas de page) est postfixe à "avec" ou infixe ?
- [Je mets ici ensemble les réponses de l'élève aux questions 1, 2 et 3.]
 - (a) Dans le cas de l'analyse de l'expression $a + -1 + 0$
 - (b) Dans le premier il faut faire un shift, et dans la deuxième cas un reduce.
 - (c) `%expect 2`

Barème:

2 Si une seule des deux situations

2. Dire dans chacun de ces deux cas qui du shift ou du reduce doit l'emporter.

Correction: Dans les deux cas il faut donc choisir le décalage, ce qui se fait simplement en donnant à l'étoile une priorité supérieure aux deux autres.

Best-of:

- Le shift l'emporte parce que les gentils gagnent toujours ! [Ah bon !?!?!?]

Barème:

2 Si une seule des deux situations

3. En déduire les directives Bison de priorité/associativité résolvant correctement tous les conflits.

Correction:

```
%left "+"
%left "."
%nonassoc "*" // Ou mieux encore, %precedence.
```

Best-of:

- ? Fonce tout droit visage pâle.
- Comme j'ai vue dans les annales que les bisons avaient été exterminés par les cowboys, je ne les aient pas cherchés.

```
%left "*" "."
%left "*" "+"
```

- `%isover90000 "*" "`

Barème: Pas de sanction pour les fautes d'orthographe sur `%nonassoc` par exemple.

1 si c'est correct, mais à l'envers.

3 si c'est correct, mais `%left` ou `%right` a été utilisé pour *. C'est une question de goût, certes, mais le bon goût compte dans la programmation.

4. Compléter, jusqu'à l'acceptation, la séquence de décalages/réductions suivante.

```

┆          0 + e . a * ┆
s ┆ "0"      + e . a * ┆
r ┆ exp      + e . a * ┆
s ┆ exp "+"  e . a * ┆
s ┆ exp "+" "e" . a * ┆
    
```

Correction:

```

┆          0 + e . a * ┆
s ┆ "0"      + e . a * ┆
r ┆ exp      + e . a * ┆
s ┆ exp "+"  e . a * ┆
s ┆ exp "+" "e" . a * ┆
r ┆ exp "+" exp . a * ┆
s ┆ exp "+" exp "." a * ┆
s ┆ exp "+" exp "." "a" * ┆
r ┆ exp "+" exp "." exp * ┆
s ┆ exp "+" exp "." exp "*" ┆
r ┆ exp "+" exp "." exp ┆
r ┆ exp "+" exp ┆
r ┆ exp ┆
s ┆ exp ┆
accept
    
```

Barème: Pas de pénalité s'il manque accept. Pas de sanction si shift/reduce n'est pas écrit.
 0 si des règles qui n'existent pas sont réduites. Beaucoup d'entre vous font par exemple

```

s ┆exp "+" "e" . a * ┆
r ┆exp . a * ┆
    
```

(sans parler de l'erreur de priorité ici).
 2 si c'est incorrect (erreur de priorité), mais décalages et réductions corrects.

4 À propos de ce cours

Nous nous engageons à ne pas tenir compte des renseignements ci-dessous pour noter votre copie. Ils ne sont pas anonymes, car nous sommes curieux de confronter vos réponses à votre note. En échange, quelques points seront attribués pour avoir répondu. Merci d'avance.

Répondez sur les formulaires de QCM qui vous sont remis. Vous pouvez cocher plusieurs réponses par question.

Q.17 Prises de notes

- a. Aucune
- b. Sur papier
- c. Sur ordinateur à clavier
- d. Sur ardoise
- e. Sur le journal du jour

Q.18 Travail personnel

- a. Rien
- b. Bachotage récent
- c. Relu les notes entre chaque cours
- d. Fait les annales
- e. Lu d'autres sources

Q.19 Ce cours

- a. Est incompréhensible et j'ai rapidement abandonné
- b. Est difficile à suivre mais j'essaie
- c. Est facile à suivre une fois qu'on a compris le truc
- d. Est trop élémentaire

Q.20 Ce cours

- a. Ne m'a donné aucune satisfaction
- b. N'a aucun intérêt dans ma formation
- c. Est une agréable curiosité
- d. Est nécessaire mais pas intéressant
- e. Je le recommande

Q.21 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

Best-of: L'absence de réponse à la question 21 n'est pas le résultat d'une négligence, mais d'un manque de choix dans les réponses.
 Je n'ai pas beaucoup travaillé alors que j'ai apprécié ce cours.
 C'est incohérent.
 C'est stupide.
 C'est l'Ing1.