

# THL – Théorie des Langages

EPITA – Promo 2010 – Documents autorisés

Janvier 2008(1h30)

Écrire court, juste, et bien. Une argumentation informelle mais convaincante est souvent suffisante.

## 1 Incontournables

Les questions suivantes sont fondamentales. Une pénalité sur la note finale sera appliquée pour les erreurs. Répondez sur les feuilles de QCM qui vous sont remises.

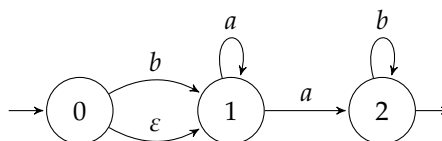
1. Tout automate non-déterministe n'est pas déterministe. a. vrai/b. faux ?
2. Le langage engendré par  $A \rightarrow B \mid A \rightarrow \times \mid B \rightarrow A$  est rationnel. a. vrai/b. faux ?
3. Toute partie d'un langage rationnel est rationnelle. a. vrai/b. faux ?

## 2 Culture Générale

1. Combien existe-t-il de sous-ensembles d'un ensemble de taille  $n$  ?
2. Combien existe-t-il de mots de  $n$  lettres écrits dans un alphabet de  $m$  symboles ?
3. Combien de valeurs différentes peut coder un octet ?
4. À quel linguiste les informaticiens doivent-ils le défrichage d'une partie importante de la théorie des langages formels ?

## 3 Automates

Déterminer rigoureusement l'automate suivant.



## 4 Parsage LL et LR

Soit  $U$  un langage pour la programmation de robots offrant des primitives de composition de processus séquentielle et parallèle :

```
p | q # Exécuter p et immédiatement à la fin d'icelui, exécuter q.
p & q # Démarrer p et q de façon synchrone.
p ; q # Exécuter p puis q après la fin de p (possiblement bien plus tard).
p , q # Lancer p puis sans attendre sa fin, exécuter q (peut-être plus tard).
```

Les accolades, '{' et '}', groupent les processus.

1. Écrire la grammaire naïve de ce langage en utilisant 'p' pour désigner les processus élémentaires. On prendra garde d'utiliser des conventions typographiques permettant de distinguer les symboles du langage U de ceux du formalisme des grammaires.
2. Dessiner les arbres de dérivation et leur arbre de syntaxe abstraite de 'p, q ; r' correspondant à deux opérateurs associatifs à gauche. Faire de même en associativité à droite.
3. La sémantique de 'p ; q' est la même qu'en shell. Celle de 'p, q' est comparable à celle de 'p&q' en shell avec l'importante différence que dans '{p, q} ; r', 'r' attendra la fin de 'p' et de 'q' pour commencer.  
Tous deux, ' ; ' et ' , ' ; ', ont même priorité. Étant donnée la sémantique voulue et pour que ' ; ' se comporte comme '&' en shell dans une phrase comme 'p, q ; r', quelle associativité faut-il prendre pour ' ; ' et ' , ' ; ' ?
4. Étant donnée leur sémantique, discuter l'associativité et la commutativité de '&' puis celles de '|'.
5. Pour rester semblable au modèle du C, on donne une priorité supérieure à '&' sur '|'. Pour aider LL, quelle associativité leur donner ?
6. Sachant que ' ; ' et ' , ' ; ' sont les moins prioritaires, donner une grammaire non ambiguë de U.
7. Pourquoi cette grammaire n'est pas LL(1) ?
8. Est-elle LL(2) ?
9. Généraliser cette grammaire en utilisant des opérateurs rationnels (EBNF).
10. Écrire la routine de parsing d'une implémentation conventionnelle de LL(1) en C (parseur prédictif récuratif descendant) pour l'axiome (le symbole de tête) de cette grammaire EBNF. Pour ce faire, utiliser les déclarations suivantes :

```
/* The type of the processus trees. */
typedef ... proc_t;
/* The connectives: ';', ',', '&', '|'. */
typedef enum { conn_semicolon, conn_comma, conn_and, conn_pipe } conn_t;

/* Return a processus which composes lhs and rhs with conn. */
proc_t* proc_new (conn_t conn, proc_t* lhs, proc_t* rhs);

/* The lookahead. */
token_t la;

/* Check that the lookahead is equal to 't', and then advance.
   Otherwise, report an error, and throw tokens until 't' (or
   end of file) is found. */
void eat (token_t t);
```

On prendra garde à :

- retourner la valeur de l'expression,
- reporter les erreurs à l'utilisateur,
- soumettre du code *lisible*,
- ne pas se perdre dans les détails, rester abstrait (e.g., on peut utiliser `afficher (la)` sans en fournir d'implémentation).

11. On souhaite à présent étudier la possibilité d'une implémentation en Bison de la grammaire précédente. Traduire la grammaire de la question 6 en une grammaire Bison exploitant les primitives d'associativité et de priorité.
12. Entre règles récursives à droite et à gauche, lesquelles préfèrent les parsers LR, et pourquoi.
13. Compléter la séquence de décalages/réductions de la phrase suivante.

```
      ⊢          p, p, p ⊢
s ⊢ 'p'      , p, p ⊢
```

14. On souhaite également utiliser le langage U avec un interprète de commande interactif, comme pour un shell Unix. Et, toujours comme pour un shell, on souhaite que 'p,' lance 'p' en « tâche de fond ».

Expliquer pourquoi cette implémentation en Bison ne convient pas.

## 5 À propos de ce cours

Bien entendu je m'engage à ne pas tenir compte des renseignements ci-dessous pour noter votre copie. Ils ne sont pas anonymes, car je suis curieux de confronter vos réponses à votre note. En échange, quelques points seront attribués pour avoir répondu. Merci d'avance.

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

4. Travail personnel
  - a Rien
  - b Bachotage récent
  - c Relu les notes entre chaque cours
  - d Fait les annales
  - e Lu d'autres sources
5. 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
6. 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
7. 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