

THL — Théorie des Langages

EPITA — Promo 2010 — **Sans document ni machine**

Janvier 2009 (1h30)

Bien lire les questions, chaque mot est important. Répondre sur les formulaires de QCM ; aucune réponse manuscrite ne sera corrigée.

Chaque question a une et une seule réponse attendue. En particulier, lorsque plusieurs réponses sont possibles, prendre la plus restrictive. Par exemple s'il est demandé si 0 est *nul*, *non nul*, *positif*, ou *négatif*, sélectionner *nul* qui est plus restrictif que *positif* et *négatif*, tous deux vrais.

1 Incontournables

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

2 Culture Générale

- Q.4 Combien existe-t-il de sous-ensembles d'un ensemble de taille n ?
 a. $n - 2$ b. $n!$ c. $n(n + 1)/2$ d. n^2 e. 2^n
 Q.5 Combien existe-t-il de mots de n lettres écrits dans un alphabet de m symboles ?
 a. $n - m$ b. $n!$ c. $n(n + m)/m$ d. n^m e. m^n
 Q.6 Combien de valeurs différentes peut coder un octet ?
 a. 8 b. 64 c. 255 d. 256 e. 1024
 Q.7 À quel linguiste les informaticiens doivent-ils le défrichage d'une partie importante de la théorie des langages ?
 a. Frank de Remer c. Donald Knuth e. Mike Lesk
 b. Noam Chomsky d. Blaise Pascal

3 Automates

- Q.8 Quelle séquence permet de calculer un automate déterministe à partir d'une expression rationnelle ?
 a. Norton, Brzozowski et McCluskey, Kleene
 b. Thompson, déterminisation, Brzozowski et McCluskey
 c. Thompson, ε -élimination arrière, déterminisation
 d. Norton, déterminisation, ε -élimination avant
 e. Thompson déterminisé

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.

Q.9 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.

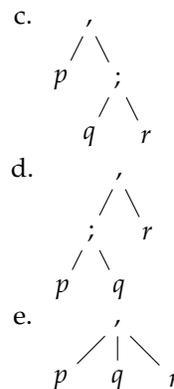
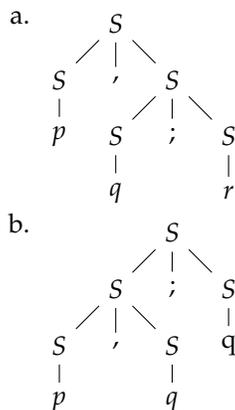
a. $S \rightarrow S \text{' ; ' } S \mid S \text{' , ' } S$
 $\quad \mid S \text{' & ' } S \mid S \text{' | ' } S$
 $\quad \mid \text{' { ' } S \text{' } ' } \mid p$

c. $S \rightarrow \text{' ; ' } S \mid \text{' , ' } S$
 $\quad \mid \text{' & ' } S \mid \text{' | ' } S$
 $\quad \mid \text{' { ' } S \text{' } ' } \mid p$

b. $S \rightarrow S \text{' ; ' } \mid S \text{' , ' }$
 $\quad \mid S \text{' & ' } \mid S \text{' | ' }$
 $\quad \mid \text{' { ' } S \text{' } ' } \mid p$

d. $S \rightarrow \text{' { ' } S \text{' ; ' } S \text{' } ' }$
 $\quad \mid \text{' { ' } S \text{' , ' } S \text{' } ' }$
 $\quad \mid \text{' { ' } S \text{' & ' } S \text{' } ' }$
 $\quad \mid \text{' { ' } S \text{' | ' } S \text{' } ' }$
 $\quad \mid \text{' { ' } p \text{' } ' }$

Q.10 Désigner les arbres de dérivation et leur arbre de syntaxe abstraite de 'p, q; r' correspondant à deux opérateurs associatifs à droite.



Q.11 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', quelles associativités sont naturelles pour ',' et ';' ?

- a. ',' et ';' associatifs à gauche
- b. ',' associatif à gauche et ';' à droite
- c. ',' et ';' associatifs à droite
- d. ',' associatif à droite et ';' à gauche
- e. ',' et ';' associatifs à gauche et à droite

Q.12 Étant donnée leur sémantique, discuter l'associativité de '&' et de '|':

- a. '&' et '|' et associatifs à gauche
- b. '&' associatif à gauche et '|' à droite
- c. '&' et '|' associatifs à droite
- d. '&' associatif à droite et '|' à gauche
- e. '&' et '|' associatifs (à gauche et à droite)

Q.13 Pour rester semblable au modèle du C, on donne une priorité supérieure à '&' sur '|'. Pour aider LL, quelle associativité leur donner ?

- a. associativité gauche et droite
- b. associativité gauche
- c. associativité droite
- d. non-associatif

Q.14 Sachant que ',' et ';' sont les moins prioritaires, donner une grammaire naturelle (pas nécessairement LL) non ambiguë de U.

a. $S \rightarrow T \mid T ';' S \mid T ',' S$
 $T \rightarrow F \mid F '|' T$
 $F \rightarrow P \mid P '&' F$
 $P \rightarrow p \mid '{' S '}'$

$P \rightarrow p \mid '{' S '}'$

b. $S \rightarrow T \mid S ';' T \mid S ',' T$
 $T \rightarrow F \mid T '|' F$
 $F \rightarrow P \mid F '&' P$
 $P \rightarrow p \mid '{' S '}'$

d. $P \rightarrow p \mid '{' S '}'$
 $F \rightarrow P \mid P '&' F$
 $T \rightarrow F \mid F '|' T$
 $S \rightarrow T \mid T ';' S \mid T ',' S$

c. $S \rightarrow T \mid S ';' T \mid S ',' T$
 $T \rightarrow F \mid T '|' F \mid T '&' F$

e. $S \rightarrow T \mid S O T$
 $O \rightarrow '|' \mid '&' \mid ';' \mid ','$
 $T \rightarrow p \mid '{' S '}'$

Q.15 Cette grammaire est-elle LL(1) ?

- a. oui
- b. non : elle est ambiguë
- c. non : les FIRST des parties droites des règles de S sont identiques
- d. non : les FOLLOW des parties gauches des règles de S sont identiques
- e. non : NULLABLE n'est pas vide

Q.16 Est-elle LL(2) ?

- a. oui
- b. non : elle est ambiguë
- c. non : les T pouvant être arbitrairement longs, les terminaux qui distinguent les règles de S peuvent être arbitrairement
- d. non : les FOLLOW(2) des parties gauches des règles de S sont identiques
- e. non : NULLABLE n'est pas vide

Q.17 Quelle est l'implémentation conventionnelle d'un parseur prédictif récursif descendant de LL(1) pour la règle suivante d'une grammaire généralisée ?

$S \rightarrow T ((';' \mid ',') T)^*$

a.

```
parse_S ()
{
  res = parse_T();
  switch (la)
  {
    case ';':
      eat(';'); res = make_semicolon(res, parse_T()); break;
    case ',':
      eat(','); res = make_comma(res, parse_T()); break;
  }
  return res;
}
```

b.

```
parse_S ()
{
  while (la == ';' || la == ',')
  switch (la)
  {
    case ';':
      eat(';'); res = make_semicolon(parse_T(), parse_T()); break;
    case ',':

```

```

        eat(','); res = make_comma(parse_T(), parse_T()); break;
    }
    return res;
}

```

c. parse_S ()

```

{
    res = parse_T();
    while (la == ';' || la == ',')
        switch (la)
        {
            case ';':
                eat(';'); res = make_semicolon(res, parse_T()); break;
            case ',':
                eat(','); res = make_comma(res, parse_T()); break;
        }
    return res;
}

```

d. parse_S ()

```

{
    do {
        res = parse_T();
        switch (la)
        {
            case ';':
                eat(';'); res = make_semicolon(res, parse_T()); break;
            case ',':
                eat(','); res = make_comma(res, parse_T()); break;
        }
    } while (la == ';' || la == ',');
    return res;
}

```

Q.18 Quelles directives d'associativité/priorité prendre pour que la grammaire Bison suivante analyse correctement le langage U ?

```

%%
p: p ',' p
  | p '|' p
  | p '&' p
  | p ';' p
  | '{' p '}'
  | 'p';

```

a. %right "," ";" %left "|" "&"

b. %left "," %left ";" %left "|" %left "&"

c. %right "," ";" %left "|" %left "&"

d. %right "," %right ";" %left "|" %left "&"

e. %left "&" %left "|" %right "," ";"

Q.19 Quelle est la séquence de décalages/réductions pour 'p & p | p' ?

a.

```

┌           p & p | p ⊢
s ⊢ "p"      & p | p ⊢
r ⊢ p        & p | p ⊢
s ⊢ p "&"     p | p ⊢
s ⊢ p "&" "p" | p ⊢
r ⊢ p        | p ⊢
s ⊢ p "|"    p ⊢
s ⊢ p "|" "p" ⊢
r ⊢ p        ⊢
s ⊢ p ⊢
accept
    
```

b.

```

┌           p & p | p ⊢
s ⊢ "p"      & p | p ⊢
r ⊢ p        & p | p ⊢
s ⊢ p "&"     p | p ⊢
s ⊢ p "&" "p" | p ⊢
r ⊢ p "&" p   | p ⊢
r ⊢ p        | p ⊢
s ⊢ p "|"    p ⊢
s ⊢ p "|" "p" ⊢
r ⊢ p "|" p  ⊢
r ⊢ p        ⊢
s ⊢ p ⊢
accept
    
```

c.

```

┌           p & p | p ⊢
s ⊢ "p"      & p | p ⊢
r ⊢ p        & p | p ⊢
s ⊢ p "&"     p | p ⊢
s ⊢ p "&" "p" | p ⊢
s ⊢ p "&" "p" "|" p ⊢
s ⊢ p "&" "p" "|" "p" ⊢
r ⊢ p "&" "p" "|" p ⊢
r ⊢ p "&" p   ⊢
r ⊢ p        ⊢
s ⊢ p ⊢
accept
    
```

d.

```

┌           p & p | p ⊢
s ⊢ "p"      & p | p ⊢
r ⊢ p        & p | p ⊢
s ⊢ p "&"     p | p ⊢
s ⊢ p "&" "p" | p ⊢
r ⊢ p "&" p   | p ⊢
s ⊢ p "&" p "|" p ⊢
s ⊢ p "&" p "|" "p" ⊢
r ⊢ p "&" p "|" p ⊢
r ⊢ p "&" p   ⊢
r ⊢ p        ⊢
s ⊢ p ⊢
accept
    
```

e.

```

┌           p & p | p ⊢
s ⊢ "p"      & p | p ⊢
r ⊢ p        & p | p ⊢
s ⊢ p "&"     p | p ⊢
s ⊢ p "&" "p" | p ⊢
r ⊢ p "&" p   | p ⊢
s ⊢ p "&" p "|" p ⊢
s ⊢ p "&" p "|" "p" ⊢
r ⊢ p "&" p "|" p ⊢
r ⊢ p "&" p   ⊢
r ⊢ p        ⊢
s ⊢ p ⊢
accept
    
```

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.

Q.20 Travail personnel

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

Q.21 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.22 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.23 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