



Q.5 Soit les langages rationnels suivants :  $A$ ,  $B$  et  $N$ . Le langage  $ANBN$  :

- a. est un langage hors-contexte ;
- b. est reconnaissable par un automate à états fini déterministe ;
- c. n'est évidemment pas rationnel ;
- d. nécessiterait une mémoire infinie pour le reconnaître.

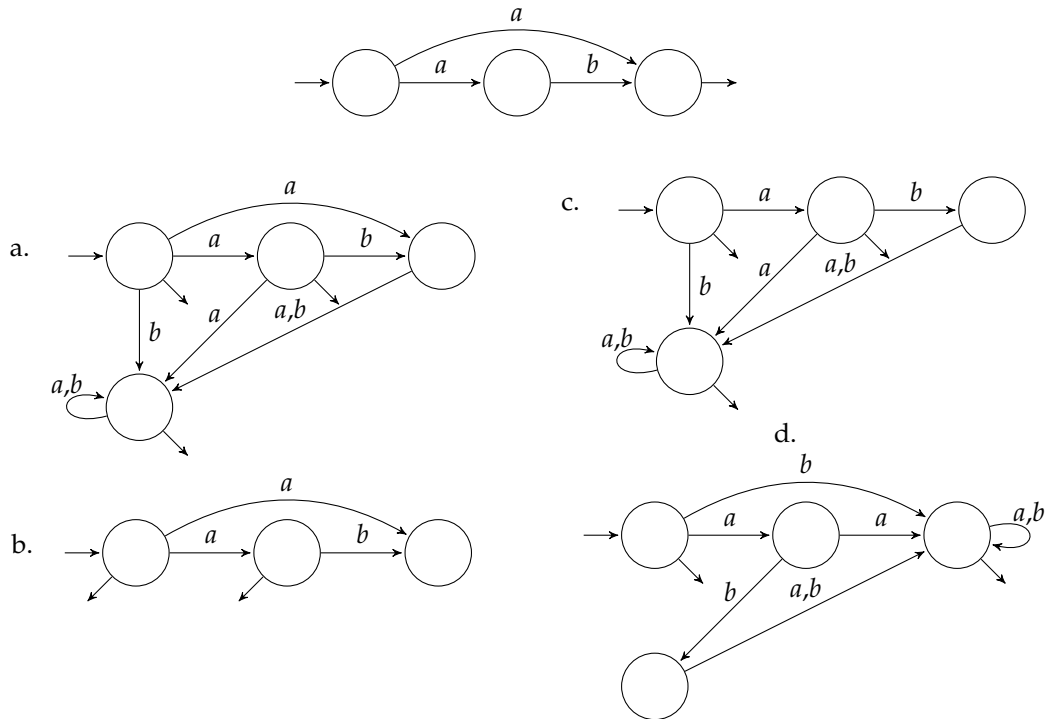
Q.6 Soit une expression rationnelle  $\alpha$  et un automate  $A$ . Est-il possible de déterminer s'ils correspondent au même langage ?

- a. c'est parfois possible ;
- b. c'est possible en temps fini ;
- c. c'est possible en temps infini ;
- d. c'est impossible.

Q.7 J'ai une expression rationnelle assez longue  $\alpha$ . J'ai construit, à l'aide de l'algorithme de Thompson un automate  $A$  reconnaissant  $\alpha$ . Combien d'états a-t-il ?

- a. 27
- b. 37
- c. 72
- d. 73

Q.8 Quel automate reconnaît le langage complémentaire du langage reconnu par l'automate suivant (sur l'alphabet  $\Sigma = \{a, b\}$ ) :



Q.9 Quelle est la classe de la grammaire suivante ?

$$\begin{aligned}A &\rightarrow aABC \mid abC \\CB &\rightarrow BC \\bB &\rightarrow bb \\bC &\rightarrow bc \\cC &\rightarrow cc\end{aligned}$$

- a. Rationnelle (Type 3)
- b. Hors contexte (Type 2)
- c. Sensible au contexte (Type 1)
- d. Monotone (Type 1)

Q.10 Quelle est la classe de la grammaire suivante ?

$$S \rightarrow aSb \mid c$$

- a. Rationnelle
- b. Hors contexte
- c. Sensible au contexte
- d. Monotone

Q.11 Si le parseur LALR(1) associé à une grammaire présente des conflits

- a. Il n'existe pas de parseur pour cette grammaire ;
- b. Il peut exister un parseur LR(0) pour cette grammaire ;
- c. Il peut exister un parseur LLR(1) pour cette grammaire ;
- d. Il peut exister un parseur LR(1) pour cette grammaire ;
- e. Il peut exister un parseur SLR(1) pour cette grammaire.

Q.12 LL(k) signifie

- a. lecture en deux passes de gauche à droite, avec  $k$  symboles de regard avant ;
- b. lecture en deux passes de gauche à droite, avec une pile limitée à  $k$  symboles ;
- c. lecture en une passe de gauche à droite, avec  $k$  symboles de regard avant ;
- d. lecture en une passe de gauche à droite, avec une pile limitée à  $k$  symboles.

Q.13 Dans une analyse classique en utilisant Lex et Yacc:

- a. on appelle la fonction `yyparse` une fois, elle appelle la fonction `yylex` plusieurs fois;
- b. on appelle la fonction `yyparse(yylex())` plusieurs fois;
- c. on appelle la fonction `yylex` plusieurs fois, puis la fonction `yyparse` une fois;
- d. on appelle la fonction `yyparse` plusieurs fois, elle appelle la fonction `yylex` chaque fois.

### 3 Dessine-moi un programme

Considérez la grammaire suivante, qui s’inspire d’un sous-ensemble du langage Logo.

```

<instr> ::= "forward" <exp>
        | "left" <exp>
        | "right" <exp>

        # Loop.
        | "repeat" <exp> "[" <instrs> "]"

        # Subprogram definition.
        | "to" "id" <args> <instrs> "end"

        # Subprogram call.
        | "id"

<exp>   ::= "num"
        | ":" "id"

<args>  ::= <args> <arg>
        |

```

Dans cette grammaire, le terminal **"num"** désigne un entier littéral et le terminal **"id"** un identifiant (qui suit les mêmes règles lexicales que les identifiants du langage Tiger).

Le langage Logo permet (entre autres) de dessiner sur un écran à l’aide d’une “tortue” que l’on manipule avec des instructions comme **forward 10** (avance de 10 unités en traçant un trait) ou **right 30** (tourne à droite de 30 degrés).

Q.14 La grammaire donnée ne décrit qu’une seule instruction, mais on voudrait pouvoir en avoir plusieurs (mais au moins une). Quelles règles est-il préférable d’ajouter si l’on veut utiliser un parseur LR(1) :

- |    |  |    |  |
|----|--|----|--|
| a. | <code>&lt;instrs&gt; ::= &lt;instr&gt;   &lt;instrs&gt; &lt;instr&gt;</code> | c. | <code>&lt;instrs&gt; ::= &lt;instr&gt;   &lt;instr&gt; &lt;instrs&gt;</code> |
| b. | <code>&lt;instrs&gt; ::=   &lt;instrs&gt; &lt;instr&gt;</code>               | d. | <code>&lt;instrs&gt; ::=   &lt;instr&gt; &lt;instrs&gt;</code>               |

Q.15 Si l’on voulait faire la même chose avec un parseur LL(1), on aurait écrit :

- |    |  |    |  |
|----|--|----|--|
| a. | <code>&lt;instrs&gt; ::= &lt;instr&gt;   &lt;instrs&gt; &lt;instr&gt;</code> | c. | <code>&lt;instrs&gt; ::= &lt;instr&gt;   &lt;instr&gt; &lt;instrs&gt;</code> |
| b. | <code>&lt;instrs&gt; ::=   &lt;instrs&gt; &lt;instr&gt;</code>               | d. | <code>&lt;instrs&gt; ::=   &lt;instr&gt; &lt;instrs&gt;</code>               |

## Arithmétique

On souhaite donner la possibilité à l'utilisateur de saisir des expressions arithmétiques afin de donner plus de flexibilité. Ainsi nous modifions les règles  $\langle exp \rangle$  :

```

 $\langle exp \rangle ::= \langle exp \rangle \text{ "+" } \langle exp \rangle \mid \langle exp \rangle \text{ "-" } \langle exp \rangle \mid \langle exp \rangle \text{ "*" } \langle exp \rangle \mid \langle exp \rangle \text{ "/" } \langle exp \rangle$ 
           | ":" "id"
           | "num"

```

Malheureusement, ces règles introduisent des ambiguïtés. Elles ne codent pas l'associativité gauche comme on le souhaiterait ni la priorité de "\*" et "/" sur "+" et "-".

1. Donner deux arbres de **dérivation** de l'expression :  
:step \* 12 + :iteration
2. Proposer une grammaire non ambiguë équivalente à la grammaire de  $\langle exp \rangle$  pour résoudre les conflits et pour respecter l'associativité gauche ainsi que la priorité des opérateurs. Profitez-en pour ajouter les parenthèses.
3. Avec Bison, quelles directives ajouteriez-vous, sans changer les règles de  $\langle exp \rangle$ , pour respecter l'associativité et les priorités.
4. L'utilisation des nouvelles règles de  $\langle exp \rangle$  dans la grammaire initiale introduit-elle des conflits ? Si oui, lesquels ?
5. Compléter, jusqu'à l'acceptation, la séquence de décalages/réductions suivante :

```

      ⊢                forward : step * 12 + : iteration ⊢
s ⊢ "forward"          : step * 12 + : iteration ⊢
s ⊢ "forward" ":"      step * 12 + : iteration ⊢
s ⊢ "forward" ":" "id"  * 12 + : iteration ⊢
r ⊢ "forward" exp      * 12 + : iteration ⊢

```

## 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.16 Prises de notes

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

### Q.17 Travail personnel

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

### Q.18 Ce cours

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

### Q.19 Ce cours

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

### Q.20 L'enseignant

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