

Correction du QCM THL & CCMP

EPITA

Abstract

Ce document veut rassembler toutes les questions à choix multiples que nous utilisons, de façon à servir de banque de questions. Merci d'y consigner toutes les questions que vous rédiger, de les classer, pour qu'elles puissent servir à d'autres occasions.

Attention, dans ces questions il y a toujours une et une seule réponse valable. 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 *négalif*, sélectionner *nul* qui est plus restrictif que *positif* et *négalif*, tous deux vrais.

Chapter 1

THLR

1.1 Langages rationnels

Le langage a^n est

Réponses possibles :

- ☐ fini
- ☒ rationnel
- ☐ non reconnaissable par automate fini
- ☐ vide

Le langage $a^n b^n$ pour $n < 42^{51} - 1$ est . .

Réponses possibles :

- ☐ infini
- ☒ rationnel
- ☐ non reconnaissable par automate fini
- ☐ vide

Le langage $(ab)^n$ est . .

Réponses possibles :

- ☐ fini
- ☒ rationnel
- ☐ non reconnaissable par automate fini
- ☐ vide

Le langage $a^n b^m$, où n, m parcourent les entiers naturels, est. . .

Réponses possibles :

- ☐ fini
- ☐ rationnel
- ☐ non reconnaissable par automate fini
- ☐ vide

Le langage $a^n b^n$ est. . .

Réponses possibles :

- ☐ fini
- ☐ rationnel
- ☐ non reconnaissable par automate fini
- ☐ vide

Le langage des nombres binaires premiers compris entre 0 et $2^{2^2} - 1$ est. . .

Réponses possibles :

- ☐ rationnel
- ☐ non reconnaissable par un automate fini déterministe
- ☐ non reconnaissable par un automate fini nondéterministe
- ☐ non reconnaissable par un automate fini à transitions spontanées

Les logins de votre promo constituent un langage. . .

Réponses possibles :

- ☐ rationnel
- ☐ non reconnaissable par un automate fini déterministe
- ☐ non reconnaissable par un automate fini nondéterministe
- ☐ non reconnaissable par un automate fini à transitions spontanées

Un langage quelconque. . .

Réponses possibles :

- ☐ est toujours inclus (\subset) dans un langage rationnel
- ☐ est toujours récursif
- ☐ peut avoir une intersection non vide avec son complémentaire
- ☐ peut ne pas être inclus dans un langage défini par une expression rationnelle

1.2 Expressions rationnelles

Il est possible de tester si une expression rationnelle engendre un langage vide.

Réponses possibles :

- Toujours vrai
- × Souvent vrai
- × Souvent faux
- × Toujours faux

Il est possible de tester si une expression rationnelle engendre un langage infini.

Réponses possibles :

- Toujours vrai
- × Souvent vrai
- × Souvent faux
- × Toujours faux

L'expression rationnelle étendue $[-+]?[0-9]^+, [0-9]^*$ n'engendre pas:

Réponses possibles :

- 42
- × 42,
- × 42,4
- × 42,42

L'expression rationnelle étendue $[-+]?[0-9]^+ + (, [0-9]^+)(e[-+]?[0-9]^+)$ n'engendre pas:

Réponses possibles :

- × $42e42$
- $42, e42$
- × $42, 4e42$
- × $42, 42e42$

L'expression rationnelle étendue $[a - zA - Z][a - zA - Z0 - 9]^*$ n'engendre pas:

Réponses possibles :

- `__STDC__`
- × `main`
- × `eval_expr`
- × `exit_42`

L'expression rationnelle étendue `"([a - zA - Z]|\\) + "` engendre :

Réponses possibles :

- × `""`
- × `"\"`
- `"\\\\"`
- × `"\n"` où `\n` représente le caractère « retour à la ligne »

1.3 Automates

L'automate de Thompson de l'expression rationnelle $(ab)^*c$

Réponses possibles :

- × ne contient pas de boucle
- × n'a aucune transition spontanée
- a 8, 10, ou 12 états
- × est déterministe

Quelle est l'écriture la plus raisonnable ?

Réponses possibles :

- × machine à état fini
- × machine à état finis
- machine à états finie
- × machine à états finis

Un automate fini déterministe. . .

Réponses possibles :

- × n'est pas nondéterministe
- × n'est pas à transitions spontanées
- n'a pas plusieurs états initiaux
- × n'a pas plusieurs états finaux

Soit une expression rationnelle α et un automate A . Il est possible de déterminer s'ils correspondent au même langage :

Réponses possibles :

- × vrai en temps constant
- vrai en temps fini
- × faux en temps fini
- × faux en temps infini

Il est possible de tester si un automate est déterministe.

Réponses possibles :

- Toujours vrai
- × Souvent vrai
- × Souvent faux
- × Toujours faux

Est-il possible de tester si un automate nondéterministe reconnaît un langage non vide ?

Réponses possibles :

- Oui, toujours.
- × Souvent.
- × Rarement.
- × Jamais.

Existe-t-il un algorithme testant si un automate déterministe reconnaît un langage non vide ?

Réponses possibles :

- Oui.
- × Seulement si le langage n'est pas rationnel.
- × Non.
- × Cette question n'a pas de sens.

Chapter 2

THL

2.1 Hiérarchie de Chomsky

2.1.1 Grammaires

Quelle est la classe de la grammaire suivante ?

$$\begin{aligned}P &\rightarrow P \text{ inst } ' ;' \\P &\rightarrow \text{ inst } ' ;'\end{aligned}$$

Réponses possibles :

- Rationnelle (Type 3)
- × Hors contexte (Type 2)
- × Sensible au contexte (Type 1)
- × Monotone (Type 1)

Quelle est la classe de la grammaire suivante ?

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

Réponses possibles :

- × Rationnelle (Type 3)
- × Hors contexte (Type 2)
- × Sensible au contexte (Type 1)
- Monotone (Type 1)

Quelle est la classe de la grammaire suivante ?

$$\begin{aligned} S &\rightarrow abc \\ S &\rightarrow aSQ \\ bQc &\rightarrow bbcc \\ cQ &\rightarrow Qc \end{aligned}$$

Réponses possibles :

- × Rationnelle
- × Hors contexte
- × Sensible au contexte
- Monotone

Quelle est la classe de la grammaire suivante ?

$$\begin{aligned} S &\rightarrow abC \\ S &\rightarrow aSQ \\ bQC &\rightarrow bbCC \\ CQ &\rightarrow CX \\ CX &\rightarrow QX \\ QX &\rightarrow QC \\ C &\rightarrow c \end{aligned}$$

Réponses possibles :

- × Rationnelle
- × Hors contexte
- Sensible au contexte
- × Monotone

Quelle est la classe de la grammaire suivante ?

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow c \end{aligned}$$

Réponses possibles :

- × Rationnelle
- Hors contexte
- × Sensible au contexte
- × Monotone

Quelle est la classe de la grammaire suivante ?

$$S \rightarrow aS$$

$$S \rightarrow Sb$$

$$S \rightarrow c$$

Réponses possibles :

- ☐ Rationnelle
- ☒ Hors contexte
- ☐ Sensible au contexte
- ☐ Monotone

Quelle est la classe de la grammaire suivante ?

$$S \rightarrow SaS$$

$$S \rightarrow c$$

Réponses possibles :

- ☐ Rationnelle
- ☒ Hors contexte
- ☐ Sensible au contexte
- ☐ Monotone

Quelle est la classe de la grammaire suivante ?

$$S \rightarrow Sac$$

$$S \rightarrow c$$

Réponses possibles :

- ☒ Rationnelle
- ☐ Hors contexte
- ☐ Sensible au contexte
- ☐ Monotone

2.1.2 Machines abstraites

Un transducteur est

Réponses possibles :

- × un élément de transitor
- une machine ayant une entrée et une sortie
- × un automate fini avec des transductions spontanées
- × un automate infini

Une machine de Turing nondéterministe

Réponses possibles :

- × ne sait pas ce qu'elle fait
- est sûrement plus efficace qu'une machine de Turing déterministe
- × permet d'aboutir à une réponse là où les machines déterministes échouent
- × gèrent les ensembles flous

2.1.3 Divers

Un langage quelconque. . .

Réponses possibles :

- est toujours inclus dans un langage rationnel
- × est toujours inclus dans un langage hors-contexte
- × est toujours inclus dans un langage sensible au contexte
- × est toujours inclus dans un langage de récursivement énumérable

Il existe un formalisme qui permette une description finie de tout langage.

Réponses possibles :

- × Oui.
- × Ça dépend du formalisme.
- × Ça dépend de l'alphabet.
- Non.

L'équation $P \subset NP$ signifie

Réponses possibles :

- × un problème de résolution d'équations polynomiales est plus facile qu'un problème de résolution d'équations exponentielles
- on ne perd pas de performances en ayant plus de CPU
- × les problèmes solubles dans un polynôme précipitent dans une solution non polynomiale
- × un problème solvable par une machine de Turing à une bande P est solvable par une machine de Turing ayant en plus une bande N .

Soit L_r est un langage rationnel. Si $L \subset L_r$, alors...

Réponses possibles :

- × L est rationnel
- × L est hors-contexte
- × L est sensible au contexte
- L peut ne pas être définissable par une grammaire

2.2 Grammaires Hors Contexte

2.2.1 Rationnelle vs. ambiguë

Quelle propriété cette grammaire vérifie ?

$$\begin{aligned} S &\rightarrow Sac \\ S &\rightarrow c \end{aligned}$$

Réponses possibles :

- Linéaire à gauche
- × Linéaire à droite
- × Hors contexte
- × Ambiguë

Quelle propriété cette grammaire vérifie ?

$$S \rightarrow aSc$$

$$S \rightarrow c$$

Réponses possibles :

- × Linéaire à gauche
- × Linéaire à droite
- Hors contexte
- × Ambiguë

Quelle propriété cette grammaire vérifie ?

$$S \rightarrow SpS$$

$$S \rightarrow n$$

Réponses possibles :

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

2.2.2 Propriétés

Parmi les propriétés suivantes, laquelle est vérifiée pour toute grammaire hors contexte ambiguë ?

Réponses possibles :

- Elle produit un langage non vide
- × Elle produit un langage rationnel
- × Elle produit un langage infini
- × Elle produit un langage non rationnel

Une grammaire hors contexte est ambiguë ssi il existe

Réponses possibles :

- un mot ayant deux arbres de dérivation.
- × un mot ayant une dérivation droite et une dérivation gauche.
- × une dérivation gauche (ou droite) ayant deux arbres de dérivation.
- × un automate nondéterministe qui reconnaisse ses arbres de dérivation.

2.3 Analyse Syntaxique

Si une grammaire hors contexte est non ambiguë

Réponses possibles :

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

2.3.1 LL(1)

Si une grammaire hors contexte est LL(1), alors

Réponses possibles :

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

Si une grammaire hors contexte est non ambiguë

Réponses possibles :

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

LL(k) signifie

Réponses possibles :

- × lecture en deux passes de gauche à droite, avec k symboles de regard avant
- × lecture en deux passes de gauche à droite, avec une pile limitée à k symboles
- lecture en une passe de gauche à droite, avec k symboles de regard avant
- × lecture en une passe de gauche à droite, avec une pile limitée à k symboles

2.3.2 LALR

Si un parseur LALR(1) a des conflits, alors sa grammaire
--

Réponses possibles :

- × est ambiguë
- × n'est pas LR(1)
- n'est pas LR(0)
- × n'est pas déterministe

Chapter 3

CMP1 — Front End

3.1 Généralités

Un compilateur large

Réponses possibles :

- × fait passer toute la chaîne de traitement à chaque ligne de programme l'une après l'autre
- × compile les langages des main-frames
- fait passer toutes les étapes au programme entier les unes après les autres
- × ne peut pas détecter les fonctions inutilisées

3.2 Analyse

Lex/Flex sont des

Réponses possibles :

- générateurs de scanners
- × générateurs de parsers
- × parseurs
- × scanners

Yacc/Bison sont des

Réponses possibles :

- × générateurs de scanners
- générateurs de parsers
- × parseurs
- × scanners

Comment désambigüiser pour Yacc/Bison le morceau d'arithmétique suivant:

```
exp: exp '+' exp | exp '-' exp | NUM;
```

Réponses possibles :

- %left '+' '-'
- × %left '+' %left '-'
- × %left '-' %left '+'
- × %left '+' %left '-' %nonassoc NUM

Comment désambigüiser pour Yacc/Bison le morceau d'arithmétique suivant:

```
[language=Bison] exp: exp '*' exp | exp '+' exp | NUM;
```

Réponses possibles :

- × %left '+' '*'
- %left '+' %left '*'
- × %left '*' %left '+'
- × %left '*' %left '+' %nonassoc NUM

Le métier d'un parser est de

Réponses possibles :

- × segmenter un flux de caractères en un flux de tokens
- × s'assurer que les types sont bien utilisés
- × éliminer les récursions terminales
- faire de l'analyse syntaxique

Les "start conditions" de Lex/Flex (%s et %x) permettent

Réponses possibles :

- × le choix du parseur à utiliser
- × de déterminer quand l'analyse lexicale doit commencer
- de supporter différents contextes lexicaux
- × la conversion des chaînes de chiffres en la valeur qu'elles représentent

Yacc repose sur l'algorithme

Réponses possibles :

- × LL(k)
- × YACC(1)
- × LR(k)
- LALR(1)

3.3 AST

La syntaxe concrète est

Réponses possibles :

- × une grammaire de Backus en forme de Naur partagée
- l'interface homme machine d'un langage de programmation
- × une représentation des programmes à partir d'objets
- × une méthode de modélisation pragmatique

Que signifie AST ?

Réponses possibles :

- × Arbre abstrait de synthèse
- × Arbre de synthèse abstraite
- × Arbre abstrait de syntaxe
- Arbre de syntaxe abstraite

AST signifie

Réponses possibles :

- × Adjunction of Sub Types
- Abstract Syntax Tree
- × for All Such That
- × Absolutely Strict Typing

ASN.1 est

Réponses possibles :

- × un langage de programmation abstrait, simple, normalisé
- × un langage de spécification fonctionnel
- une syntaxe pour décrire des paquets de données structurées
- × le premier outil de calcul électromécanique

Désucreur signifie

Réponses possibles :

- × retirer les commentaires, signes de ponctuation et retour à la ligne
- traduire certaines phrases dans une forme plus primitive
- × convertir une grammaire de SUGAR (SUGAR Unleashes Grammar Attribute Rules) à YACC (Yet Another Compiler Compiler)
- × reconnaître et corriger les erreurs de programmation typiques

La classe Symbol qui gère les identificateurs permet de

Réponses possibles :

- les manipuler aussi efficacement que s'il s'agissait d'entiers
- × de faire de la correction orthographique
- × de faire de la reprise sur erreur
- × de s'affranchir des caractères ASCII

Chapter 4

CMP2 — Back end

4.1 Langage intermédiaire

Quel rôle ne joue pas les langages intermédiaires ?

Réponses possibles :

- résolution de la surcharge
- × factorisation de certaines optimisations
- × décomposition en plusieurs étapes de la traduction
- × indépendance des parties frontales et terminales

4.2 Traduction vers l'assembleur

Que signifie BURG

Réponses possibles :

- × C'est le nom de son inventeur
- × Boolean Unit Rational Grammar
- Bottom Up Rewrite Generator
- × BASIC Users Redmond Group

4.3 Flot de contrôle

Que n'inclue pas le flot de contrôle ?

Réponses possibles :

- les appels de fonctions
- × les branchements conditionnels
- × les branchements inconditionnels
- × le passage à l'instruction suivante

4.4 Vivacité des variables

Quelles sont les variables vivantes au sortir de ce code ?

```
a := 0 L1: b := a + 1 a := b * 2
```

Réponses possibles :

- × a et b
- on ne peut pas le savoir
- × a
- × b

Pour le code d'une fonction, les registres callee-save sont

Réponses possibles :

- × live-in et live-out
- live-in
- × live-out
- × on ne peut pas le savoir

4.5 Allocation des registres

On peut colorer un graphe en quatre couleurs.

Réponses possibles :

- × toujours et même trois suffisent
- × toujours
- parfois
- × jamais

Que signifie le "spilling" ?

Réponses possibles :

- × la fusion deux temporaires
- × l'allocation d'une temporaire dans un registre
- l'allocation d'une temporaire sur la pile
- × la sauvegarde d'un registre callee-save

Que signifie le "coalescing" ?

Réponses possibles :

- la fusion deux temporaires
- × l'allocation d'une temporaire dans un registre
- × l'allocation d'une temporaire sur la pile
- × la sauvegarde d'un registre callee-save

Chapter 5

TYLA

Les multiméthodes permettent

Réponses possibles :

- × aux méthodes de retourner plusieurs résultats
- le polymorphisme dynamique sur les arguments de fonctions
- × aux classes d'avoir plusieurs méthodes de même nom
- × différents paradigmes de programmation

Le patron de conception « Visitor » permet l'utilisation

Réponses possibles :

- × d'itérateurs en profondeur d'abord
- × d'itérateurs en largeur d'abord
- des multiméthodes dans un langage objet qui en est démunie
- × d'accesseurs sur des membres pourtant privés

Les templates de classe du C++

Réponses possibles :

- × sont des collections de templates de fonctions libres
- sont des générateurs de classes
- × sont des classes dont toutes les méthodes sont virtuelles
- × sont des classes dont toutes les méthodes sont virtuelles pures

Un design pattern est

Réponses possibles :

- × un langage de conception universel
- une bonne solution à un problème connu
- × une méthode de conception d'application orientée objet
- × un générateur de classes de conception

La liaison dynamique en C++

Réponses possibles :

- × fait référence aux bibliothèques dynamiques
- × a rapport avec la surcharge d'opérateurs
- a rapport avec "virtual"
- × repose sur "template"

La résolution des appels "virtual" nécessite

Réponses possibles :

- × la connaissance du type des contenants
- × la connaissance du type des classes
- × la connaissance du type des opérateurs
- la connaissance du type des contenus

La résolution de la surcharge nécessite

Réponses possibles :

- la connaissance du type des contenants
- × la connaissance du type des classes
- × la connaissance du type des opérateurs
- × la connaissance du type des contenus

Les visiteurs

Réponses possibles :

- × permettent de parcourir de façon générique les conteneurs
- × sont des fonctions objets
- permettent d'implémenter le "dispatching" une fois pour toute
- × remplacent les accesseurs

Chapter 6

Correction