

# Correction du QCM CMP

EPITA – Promo 2008

Octobre 2005

Attention, dans ces qcms il y a toujours une et une seule réponse valable. En particulier, lorsque plusieurs réponses sont possibles, prendre la plus restrictive.

# Chapter 1

## CMP1 — Front End

### 1.1 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
- ×

### 1.2 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 désambiguer 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 désambiguer 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
- × faire de l'imagerie médicale
- × éliminer les récursions terminales
- faire de l'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
- ×

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 :**

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

### 1.3 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
- × Atlantic Standard Time
- × 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 2

## CMP2 — Back end

### 2.1 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
- ×

### 2.2 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
- ×



## 2.3 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
- ×

## 2.4 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
- ×

## 2.5 Allocation des registres

On peut toujours colorer un graphe en quatre couleurs ?

**Réponses possibles :**

- × vrai
- faux
- ×
- ×
- ×

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 3

# 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 démuné
- × 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
- ×