

Correction du Partiel CCMP1

CONSTRUCTION DES COMPILATEURS 1

EPITA – Promo 2013 – Sans documents ni machine

Janvier 2011 (1h30)

Correction: Le sujet et sa correction ont été écrits par Akim Demaille.

Bien lire les questions, chaque mot est important. Répondre sur les formulaires de QCM ; aucune réponse manuscrite ne sera corrigée. Renseigner les champs d'identité. Il peut y avoir plusieurs réponses justes pour ces questions.

1 Incontournables

Chaque erreur aux trois questions suivantes retire 1/6 de la note finale. Avoir tout faux divise donc la note par 2.

Q.1 Qu'est Yacc ?

- un générateur de scanners
- un générateur de parsers
- un générateur de grammaires
- un générateur de visiteurs

Q.2 La surcharge de fonctions en C++ est un mécanisme qui dépend des types à l'exécution.

- vrai
- faux

Correction: C'est faux : c'est un mécanisme statique, qui dépend des types *statique* des arguments effectifs, résolu à la compilation.

Q.3 Toute grammaire qui présente des conflits dans un parser LR(1) est une grammaire ambiguë.

- vrai
- faux

Correction: Non. Question posée en partiel de THL.

2 Contrôle

Q.4 Combien existe-t-il de nombres octaux à quatre chiffres ?

- 12^2
- 12^4
- 2^{12}
- 8^{12}

Correction: Cette question avait été posée dans le partiel de THL.
 $8^4 = (2^3)^4 = 2^{3 \times 4} = 2^{12} = 4096.$

Q.5 Quelles phases appartiennent à la partie frontale d'un compilateur ?

- parser
- allocation des registres
- typage
- sélection des instructions
- édition des liens

Q.6 À quoi s'intéresse la partie centrale d'un compilateur ?

- analyse de l'AST
- optimisations indépendantes des langages source et cible
- allocation des registres
- typage des tokens
- édition des liens

Q.7 Quelles phases appartiennent à la partie terminale d'un compilateur ?

- parser
- allocation des registres
- typage
- sélection des instructions
- édition des liens

Q.8 Les visiteurs

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

Q.9 Les multiméthodes permettent

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

3 Exercice

On considère l'ajout en Tiger de l'opérateur '??' tel qu'on peut le rencontrer en C# : 'a ?? b' s'évalue en a si a n'est pas « faux », et b sinon. Dans un premier temps, on ne s'intéressera qu'au cas où a et b sont des expressions entières.

Correction: La rédaction du sujet était malheureuse. Par exemple ici, au lieu de « des expressions entières », il était écrit « des entiers ». Tel que rédigé, vous étiez autorisés à considérer que les expressions n'avaient pas d'effet de bord, or, le but de cet exercice était précisément de prêter attention aux effets de bord. La notation a donc été faite en tenant compte des erreurs dans la rédaction originale du sujet.

Q.10 Quelles règles ajouter au scanner Flex ?

- ```
%x SC_AFTER_QMARK
%%
"?" BEGIN SC_AFTER_QMARK;
<SC_AFTER_QMARK>"?" return TOK_QUESTION_QUESTION;
```
- ```
"??" return TOK_QUESTION; return TOK_QUESTION;
```
- ```
"??" { yy1val->text = strdup (yytext);
return TOK_OPERATOR; }
```
- ```
"??" return TOK_QUESTION_QUESTION;
```

Correction: La réponse reposant sur yytext n'est pas absurde en elle-même. C'est couplée avec le parser qu'elle devient mauvaise, puisqu'elle ne donne pas de moyen de contrôler l'associativité/priorité de cet opérateur.

Q.11 Quelles règles ajouter au parser Bison ?

- ```
exp ::= exp "??" exp.
```
- ```
exp ::= exp "?" "?" exp.
```
- ```
exp: exp "??" exp;
```
- ```
exp: exp "??" exp %left;
```
- ```
exp: exp "??" exp %right;
```

Q.12 Combien existe-t-il d'arbres de dérivation pour 'a ?? b ?? c' ?

- 0                       1                       2                       3                       4

Q.13 Étant donnée sa sémantique, cet opérateur doit être rendu :

- associatif à droite                       non-associatif                       idempotent  
 associatif à gauche                       commutatif

Q.14 On le rendra associatif à droite, et moins prioritaire que les opérateurs logiques '|' et '&' (ce dernier étant prioritaire sur le précédent). Quelles directives Bison utiliser ?

- ```
%right "??"  
%left "|"
```

```
%left "|"
```

```
%right "??"  
%left "|"
```

```
%right "??"  
%left "&"
```

```
%left "&"
```

```
%left "|"
```

```
%right "??"  
%left "&"
```

Q.15 Comment désucre 'a ?? b' pour les entiers ?

- ```
if a then a else b
```
- ```
let  
  var a := a  
  var b := b  
in  
  if a <> 0 then a else b  
end
```
- ```
let
 var b := b
 var a := a
in
 if a <> 0 then a else b
end
```
- ```
let  
  var test := a <> 0  
  var true := a  
  var false := b  
in  
  if test then true else false  
end
```
- ```
let
 var a := a
in
 if a <> 0 then a else b
end
```

**Correction:** Il fallait remarquer que dans le cas où  $a$  et  $b$  sont des expressions de type entier, alors elles peuvent quand même être des fonctions avec des effets de bord, auquel cas la seule bonne réponse est la (e). Toutes les autres, ou bien peuvent évaluer  $a$  deux fois, ou bien évaluent  $b$  même quand il ne le faut pas, voire évaluent même  $b$  avant  $a$ .

Si cependant  $a$  et  $b$  sont des littéraux entiers, toutes les réponses sont justes. Par conséquent dans la notation, cette question a un coefficient nul.

Q.16 Quelles sont les règles de typage de 'a ?? b' ?

- $a$  entier, retourne entier  
  $a$  et  $b$  entiers, retourne entier  
  $a$  entier ou nul, et  $b$  entier non nul, retourne non nul

Q.17 Étant donné ce désucre (question (15)) et ces règles de typage (question (16)), il n'est pas nécessaire de faire un contrôle de type spécifique pour '??'

- vrai                                               faux

Q.18 Quels modules du compilateur Tiger faut-il modifier pour supporter l'opérateur '??' sur les entiers ?

- scanner                                               liaison de noms                                               code intermédiaire  
 parser                                               typage

Q.19 On désire à présent que cet opérateur fonctionne également avec les chaînes de caractères : 'a ?? b' retourne  $b$  si  $a$  est la chaîne vide,  $a$  sinon. Quelles sont les règles de typage de '??' ?

- prend deux chaînes ou deux entiers, et renvoie une chaîne

- ✗ prend deux chaînes ou deux entiers, et renvoie une entier
- ✓ prend deux chaînes et renvoie une chaîne ou bien prend deux entiers et renvoie un entier

Q.20 Comment désucre 'a ?? b' si a et b sont des expressions de type chaîne de caractères ?

|   |                                                                                       |   |                                                                                                                        |
|---|---------------------------------------------------------------------------------------|---|------------------------------------------------------------------------------------------------------------------------|
| ✗ | <pre>if size(a) &lt;&gt; 0 then a else b</pre>                                        | ✗ | <pre>let   var test := size(a) &lt;&gt; 0   var true := a   var false := b in   if test then true else false end</pre> |
| ✗ | <pre>let   var a := a   var b := b in   if size(a) &lt;&gt; 0 then a else b end</pre> | ✓ | <pre>let   var a := a in   if size(a) then a else b end</pre>                                                          |
| ✗ | <pre>let   var b := b   var a := a in   if size(a) &lt;&gt; 0 then a else b end</pre> |   |                                                                                                                        |

**Correction:** À nouveau la rédaction originale pouvait laisser penser qu'il s'agissait de chaînes de caractères littérales, auquel cas toutes les réponses étaient juste. Question annulée dans la notation.

Q.21 Quels sont les modules concernés si l'on désire introduire cet opérateur sur les entiers et les chaînes de caractères ?

- ✓ scanner
- ✓ parser
- ✗ liaison de noms
- ✓ typage

**Correction:** Il faut désormais connaître le type des arguments pour résoudre l'appel, comme c'est déjà le cas par exemple pour '<' ou encore =. Il faut donc retarder le désucre à après le typage.

Q.22 On désire également pouvoir appliquer cet opérateur sur les enregistrements (les *records* de Tiger). Quelles sont les règles de typage générales de 'a ?? b' ?

- ✗ a et b doivent être simultanément des entiers, des chaînes ou des enregistrements, et c'est le type du résultat ;
- ✗ a et b doivent être du même type, et c'est le type du résultat ;
- ✓ a et b doivent être du même type (entiers, ou chaînes, ou mêmes types d'enregistrements), et c'est le type du résultat.

Q.23 Il convient de ne pas confondre une expression qui vaut 'nil' (par exemple une variable déclarée avec un type enregistrement spécifique, mais initialisée à 'nil') d'une expression de type Nil (comme 'nil' elle-même). On souhaite supporter 'nil' également. Comment traduire 'a ?? b' quand a est du type Nil ?

|   |                                                                      |   |                   |
|---|----------------------------------------------------------------------|---|-------------------|
| ✗ | <pre>let   var a := a in   if a &lt;&gt; nil then a else b end</pre> | ✗ | <pre>b</pre>      |
|   |                                                                      | ✓ | <pre>(a; b)</pre> |

## 4 À propos de ce cours

Nous nous engageons à corriger les copies sans tenir compte des renseignements ci-dessous. Ils ne sont pas anonymes de façon à pouvoir rechercher des corrélations avec les notes. 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.24 Prises de notes

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

Q.25 Travail personnel

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

Q.26 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.27 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.28 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