

Compilation de Programmes (CS444, 4A IR&C) – 2021-22

Contrôle Continu (Durée 20 min)

Notez vos NOM et Prénom ici :

NOM (majuscules) Prénom (minuscule)

.....

Toute communication (orale, téléphonique, par messagerie, etc.) avec les autres étudiants est interdite. Aucun document autorisé.

- Pour la partie QCM, plusieurs réponses peuvent être valides à chaque question, on souhaite avoir **toutes** les réponses valides. Chaque question admet au moins une réponse valide et au moins une réponse incorrecte. Il n'y aura pas de point négatif (une mauvaise réponse vaut au pire 0 points).
- Vous pouvez écrire au crayon de papier pour répondre à la question de typage.

Question 1 ♣ (1 point) Pour qu'un programme MiniC (comme dans le cours) soit bien typé :

- ☒ il faut que toutes les expressions et les instructions du programme soient bien typées
- ☒ il faut que toutes les feuilles et noeuds de l'AST soient annotées par des types valides
- ☐ il faut que toutes les variables soient initialisées
- ☐ il faut qu'il n'y ait pas de boucle infinie

Question 2 ♣ (1 point) La sémantique d'un programme :

- ☐ est une représentation du programme sous forme d'arbre
- ☐ est toujours définie et documentée proprement
- ☐ est l'ensemble des éléments lexicaux (lexèmes) produits par le lexeur
- ☒ est une façon d'expliquer l'exécution du programme

Question 3 ♣ (1 point) Lorsqu'on utilise les règles de typage, notées $\frac{P}{Q}$:

- ☐ P est la conclusion et Q l'hypothèse / la prémisse de la règle
- ☒ P est l'hypothèse / la prémisse, Q la conclusion de la règle
- ☐ Il suffit de vérifier une hypothèse parmi l'ensemble d'hypothèses de la prémisse
- ☒ Il faut que toutes les hypothèses soient vérifiées (dans la prémisse)

Question 4 ♣ (1 point) Sont des langages interprétés

- ☒ Javascript
- ☐ C++
- ☒ Python
- ☐ Java

Question 5 ♣ (1 point) Dans un compilateur C, le typage :

- ☒ Permet d'éliminer les programmes que l'on ne désire pas compiler.
- ☐ Permet de s'assurer de l'absence d'erreur à l'exécution.
- ☐ S'effectue après la phase de génération de code.
- ☐ Peut être statique ou dynamique.
- ☒ Permet d'implémenter le domaine d'application de certaines opérations de calcul, définies dans la spécification du langage.

Question 6 ♣ (1 point) Quelles sont les étapes du compilateur qui font partie du *front-end* ?

- ☒ Le typage
- ☒ L'analyse lexicale
- ☐ La génération de code
- ☒ L'analyse syntaxique
- ☐ L'allocation des temporaires en mémoire

Question 7 ♣ (1 point) Lorsque l'on attribue une grammaire :

- Question 4 (4 points) : Lorsque l'on utilise une grammaire :
- ☐ on permet de reconnaître plus de programmes
 - ☒ on limite généralement le nombre de programmes reconnus
 - ☐ on décore les non terminaux et les terminaux par des types
 - ☒ on donne un moyen de calculer par induction de l'information sur les éléments syntaxiques du programme

Toutes les solutions données ne sont que des éléments de correction qu'il faudrait rédiger mieux.

Pour l'exercice rédigé ci-dessous, on rappelle les règles de typage de MiniC du cours et du TD.

$$\begin{array}{l}
[cte] : \frac{c \in \mathbb{Z}}{\Gamma \vdash c : \mathbf{int}} \quad [env] : \frac{\Gamma(x) = t \quad t \in \{\mathbf{int}, \mathbf{bool}\}}{\Gamma \vdash x : t} \quad [seq] : \frac{\Gamma \vdash S_1 : \mathbf{void} \quad \Gamma \vdash S_2 : \mathbf{void}}{\Gamma \vdash S_1; S_2 : \mathbf{void}} \\
[:=] : \frac{\Gamma \vdash e : t \quad \Gamma \vdash x : t \quad t \in \{\mathbf{int}, \mathbf{bool}\}}{\Gamma \vdash x = e : \mathbf{void}} \quad [while] : \frac{\Gamma \vdash b : \mathbf{bool} \quad \Gamma \vdash S : \mathbf{void}}{\Gamma \vdash \mathbf{while}(b)\{S\} : \mathbf{void}} \\
[if] : \frac{\Gamma \vdash b : \mathbf{bool} \quad \Gamma \vdash S_1 : \mathbf{void} \quad \Gamma \vdash S_2 : \mathbf{void}}{\Gamma \vdash \mathbf{if } b \mathbf{ then } S_1 \mathbf{ else } S_2 : \mathbf{void}} \quad [skip] : \frac{}{\Gamma \vdash \mathbf{skip} : \mathbf{void}} \\
[+] : \frac{\Gamma \vdash e_1 : \mathbf{int} \quad \Gamma \vdash e_2 : \mathbf{int}}{\Gamma \vdash e_1 + e_2 : \mathbf{int}} \quad [<] : \frac{\Gamma \vdash e_1 : \mathbf{int} \quad \Gamma \vdash e_2 : \mathbf{int}}{\Gamma \vdash e_1 < e_2 : \mathbf{bool}}
\end{array}$$

Question 8 (3 points)

```
if (x + y < 3) then  x = x + 2; else skip;
```

Utilisez les règles de typage et un arbre de preuve pour montrer que le programme ci-dessus est bien typé sous l'environnement $[x \mapsto \text{int}, y \mapsto \text{int}]$.

0	1	2	3	4	5
---	---	---	---	---	---

Réservé[illegible]

CORRECTED

10 Total : 10

Question 7 ♣ (1 point) Lorsque l'on attribue une grammaire :

- ☐ on permet de reconnaître plus de programmes
- ☒ on limite généralement le nombre de programmes reconnus
- ☐ on décore les non terminaux et les terminaux par des types
- ☒ on donne un moyen de calculer par induction de l'information sur les éléments syntaxiques du programme

Toutes les solutions données ne sont que des éléments de correction qu'il faudrait rédiger mieux.

Pour l'exercice rédigé ci-dessous, on rappelle les règles de typage de MiniC du cours et du TD.

$[cte] : \frac{c \in \mathbb{Z}}{\Gamma \vdash c : \text{int}}$	$[env] : \frac{\Gamma(x) = t \quad t \in \{\text{int}, \text{bool}\}}{\Gamma \vdash x : t}$	$[seq] : \frac{\Gamma \vdash S_1 : \text{void} \quad \Gamma \vdash S_2 : \text{void}}{\Gamma \vdash S_1; S_2 : \text{void}}$
$[:=] : \frac{\Gamma \vdash e : t \quad \Gamma \vdash x : t \quad t \in \{\text{int}, \text{bool}\}}{\Gamma \vdash x = e : \text{void}}$	$[while] : \frac{\Gamma \vdash b : \text{bool} \quad \Gamma \vdash S : \text{void}}{\Gamma \vdash \text{while}(b)\{S\} : \text{void}}$	
$[if] : \frac{\Gamma \vdash b : \text{bool} \quad \Gamma \vdash S_1 : \text{void} \quad \Gamma \vdash S_2 : \text{void}}{\Gamma \vdash \text{if } b \text{ then } S_1 \text{ else } S_2 : \text{void}}$	$[skip] : \frac{}{\Gamma \vdash \text{skip} : \text{void}}$	
$[+] : \frac{\Gamma \vdash e_1 : \text{int} \quad \Gamma \vdash e_2 : \text{int}}{\Gamma \vdash e_1 + e_2 : \text{int}}$	$[<] : \frac{\Gamma \vdash e_1 : \text{int} \quad \Gamma \vdash e_2 : \text{int}}{\Gamma \vdash e_1 < e_2 : \text{bool}}$	

Question 8 (3 points)

P

if (x + y < 3) then x = x + 2; else skip;

Utilisez les règles de typage et un arbre de preuve pour montrer que le programme ci-dessus est bien typé sous l'environnement $[x \mapsto \text{int}, y \mapsto \text{int}]$.

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 Réserve

Soit P le programme

toutes les feuilles sont des axiomes, donc le programme est bien typé

$\Gamma(y) = \text{int}$ $\Gamma(y) = \text{int}$ $\Gamma(x) = \text{int}$ $\Gamma(z) = \text{int}$ $\Gamma(y) = \text{int}$

$\Gamma \vdash x : \text{int}$ $\Gamma \vdash y : \text{int}$ $\Gamma \vdash x + 2 : \text{int}$ $\Gamma \vdash x : \text{int}$

$[+]$ $\Gamma \vdash x + y : \text{int}$ $\Gamma \vdash 3 : \text{int}$

$[<]$ $\Gamma \vdash x + y < 3 : \text{bool}$ $\Gamma \vdash x = x + 2 : \text{void}$ $\Gamma \vdash \text{skip} : \text{void}$

$[if]$ $\Gamma \vdash P : \text{void}$

$\exists z \in \mathbb{Z}$ $[cte]$

$(skip)$