

Contrôle continu de compilation

Christian Rinderknecht

Mercredi 8 octobre 2003

Durée : deux heures. Longueur : deux pages. Les documents et les calculatrices ne sont pas autorisés.

1. Quelle est la structure simplifiée d'un compilateur ? Que sont les phases d'analyse et de synthèse ?
2. Qu'est-ce que le typage ? Pourquoi est-ce utile que le typage soit statique ?
3. Qu'est-ce qu'un interprète ? Comparez un interprète à un compilateur.
4. Qu'est-ce que le lexique d'un langage ? Qu'est-ce que la syntaxe concrète d'un langage ? Qu'est-ce que la syntaxe abstraite d'un langage ?
5. Considérez la session suivante :

```
$ javac Toto.java
Toto.java:3: integer number too large: 3000000000000
    int x = 3000000000000;
           ^
1 error
```

En supposant que le compilateur `javac` est bien conçu, à quel moment a-t-il émis ce message ?

6. À quoi sert la sémantique formelle d'un langage de programmation ?
7. Qu'est-ce qu'une sémantique opérationnelle déterministe ? En quoi ce déterminisme est-il pertinent ?
8. En suivant la syntaxe concrète de O'Caml, donnez une syntaxe abstraite et une sémantique opérationnelle pour une calculette comprenant les constantes entières, les booléens, les variables, les quatre opérations arithmétiques, la conditionnelle et la liaison locale — sans tenir compte des erreurs. Commentez vos choix.
9. Établissez formellement la valeur de `let x = 1 in ((let x = x in x) + x)`.
10. Considérez la sémantique opérationnelle de la division qui ne spécifie pas l'ordre d'évaluation des opérandes. Réécrivez la règle pour qu'elle force l'évaluation de son second opérande avant le premier. Pour cela, inspirez-vous de la règle de la liaison locale et faites attention aux captures de variables temporaires.

L'implantation fidèle de cette nouvelle sémantique est maladroite. Proposez une variation simple du code O'Caml pour la division qui répond au problème.

11. Ajoutez à votre sémantique la prise en compte de toutes les erreurs possibles tout en minimisant le nombre de calculs. Commentez vos choix.
12. Répondez formellement aux questions :
 - (a) Qu'est-ce qu'une variable libre dans une expression ?
 - (b) Qu'est-ce qu'une expression close ?
 - (c) Quels sont les variables libres de `let x = 1 in ((let x = 2 in x) + x)` ?
13. Est-ce utile de déterminer les variables libres d'une expression avant son évaluation ?