Exame Linguagens de Programação II

19 Janeiro de 2007

Notas: O exame é com consulta e tem a duração de 2h30m.
Todas as alíneas valem 2 valores

1. Considere a linguagem de programação LambdaValue, cuja sintaxe abstracta é definida pela gramática seguinte:

$$G \quad ::= \quad Num \mid Id \mid (\texttt{fun}\ Id \Rightarrow G) \quad G \bullet G$$

As únicas construções da linguagem são o identificador (Id), os números inteiros (Num), a abstracção funcional ($\texttt{fun}\ Id \rightarrow G$), e a aplicação funcional $G \bullet G$: esta linguagem corresponde à sublinguagem da linguagem CALCF estudada na disciplina e conhecida mais vulgarmente como "cálculo lambda".

Usando a interpretação usual da aplicação de funções a argumentos conhecida como passagem de valor (call-by-value), determine o valor de cada uma das expressões seguintes se a sua avaliação terminar, ou escreva "não termina", caso contrário. Justifique a sua resposta, apresentando os passos intermédios da avaliação.

(a) $(\texttt{fun}\ x \Rightarrow x) \bullet 3$

(b) $((\texttt{fun}\ f \rightarrow (\texttt{fun}\ y \Rightarrow y \bullet f)) \bullet (\texttt{fun}\ x \Rightarrow x)) \bullet (\texttt{fun}\ y \Rightarrow y)$

(c) $(\texttt{fun}\ x \Rightarrow x \bullet x) \bullet (\texttt{fun}\ x \Rightarrow x \bullet x)$

2. Indique como representaria a sintaxe abstracta da linguagem LambdaValue usando um conjunto de classes Java. Para cada classe que identificar indique quais os seus atributos (variáveis de instância) e o construtor.

3. Considere que vai ter que especificar e programar um interpretador para a linguagem Lambda baseado num ambiente, e numa representação dos valores funcionais usando fechos.

Indique como representaria em Java o ambiente, assim como os resultados da linguagem LambdaValue (note que os resultados são os números inteiros e os fechos).

4. Indique como especificaria em cada uma das classes identificadas na alínea 2 um método com a declaração seguinte (assuma definida a classe ou interface `Environment`, definida da forma usual).

```
IValue evaluate(Environment env)
```

de modo a definir um interpretador completo para a linguagem LambdaValue.

5. Imagine agora uma extensão da linguagem LambdaValue com os operadores `delay` e `force`, definidos de seguida. A linguagem resultante, denominada LambdaForce, tem a sintaxe abstracta seguinte:

$$G ::= Num \mid Id \mid (\texttt{fun}\ Id \Rightarrow G) \mid G \bullet G \mid \texttt{delay}(G) \mid \texttt{force}(G)$$

Intuitivamente, a ideia é que $\texttt{delay}(G)$ não avalia logo o valor da expressão G, mas produz um valor especial chamado "suspensão". A única operação que pode ser aplicada a uma suspensão é a operação `force`, que avalia a expressão suspensa dentro da suspensão e produz o seu valor. A semântica destas construções segue uma disciplina estática de resolução de nomes.

Note que a avaliação de uma expressão da forma $\texttt{delay}(G)$ termina sempre! Por outro lado, a operação `force` só pode ser aplicada a uma suspensão, caso contrário ocorrerá um erro de execução. Por exemplo, o valor de $\texttt{force}(\texttt{delay}(G))$ será sempre o mesmo valor de G. Por outro lado, a tentativa de avaliar $\texttt{force}(5)$ será um erro, pois 5 não tem por valor uma suspensão.

Estenda o interpretador definido acima de modo a cobrir as novas construções `delay` e `force`. Para isso, defina as duas novas classes na sintaxe abstracta, e programe nelas o método `evaluate`. Note que não terá que introduzir no interpretador mais classes que as aqui mencionadas.

6. Recorde o sistema de tipos para a linguagem funcional tipificada estudada nas aulas, e estenda-o de modo a produzir um sistema de tipos para a linguagem LambdaForce. Considere a seguinte estrutura para os tipos:

$$T ::= \texttt{Int} \mid T \rightarrow T \mid \texttt{Frozen}(T)$$

e defina o sistema de tipos usando um conjunto de regras de inferência de tipos. O tipo $\texttt{Frozen}(T)$ deve tipificar as suspensões que quando activadas produzem valores de tipo T.

7. Que erros de execução são evitados pelo sistema de tipos que propôs na alínea anterior?

8. Usando o sistema de tipos que especificou na alínea anterior, apresente com detalhe a derivação do tipo da expressão seguinte:

$$(\texttt{fun}\ x : \texttt{Int} \Rightarrow \texttt{fun}\ f : \texttt{Frozen}(\texttt{Int} \rightarrow \texttt{Int}) \Rightarrow \texttt{delay}(\texttt{force}(f) \bullet x))$$

9. Suponha que dispõe de um compilador para a linguagem LambdaValue (parecido com o efectuado nas aulas práticas) e pretende agora produzir um compilador para .NET CLR para a linguagem LambdaForce. Indique de forma suficientemente detalhada o modo como procederia para implementar o compilador.

10. Indique o código que o seu compilador geraria para cada uma das expressões seguintes:

$$\texttt{delay}(5)$$

$$(\texttt{fun}\ x \Rightarrow \texttt{force}(x))$$