

Semântica Formal N

Trabalho Prático 2

2014/2

1 Regras

- O trabalho deve ser feito em grupos de 3 componentes.
- O grupo deverá, ao final do semestre, apresentar o trabalho para a turma. A apresentação deve ser feita por todos os integrantes do grupo e deve ser encarada como uma prova oral na qual serão feitas perguntas sobre os algoritmos e sua respectiva implementação.
- A linguagem a ser usada para implementação deve ser OCaml (www.ocaml.org).
- O grupo deverá enviar o código-fonte OCAML *via Moodle* através de link específico. As datas para a entrega e apresentação serão divulgadas via Moodle.
- Este trabalho valerá 100% da nota relativa a trabalhos práticos no semestre. A parte da linguagem relativa a L1 (sem extensões) vale exatamente 20% da nota deste trabalho.

2 Descrição do trabalho

O trabalho consiste de implementar um *interpretador e verificador de tipos* para uma extensão da linguagem L1 com **letrec** (a linguagem implementada no trabalho prático 1). Essa extensão contempla duas novas características:

- mecanismo de exceções
- listas como estruturas de dados

A sintaxe abstrata dessa nova linguagem compreende as seguintes novas construções:

$e ::=$...	
	nil	(lista vazia)
	$e_1 :: e_2$	(prefixa elemento e_1 à lista e_2)
	hd e	(primeiro elemento da lista e (<i>head</i>))
	tl e	(restante da lista e (<i>tail</i>))
	raise	(lançamento de exceção)
	try e_1 with e_2	(tratamento de exceção)

Note que a verificação de tipos é mais complexa do que em L1 tradicional. Por exemplo, considerem qual seria o tipo dos seguintes programas:

1. `raise`
2. `nil`
3. `(hd raise) :: nil`
4. `nil :: nil`

A referência para o verificador de tipos do trabalho é o capítulo 22 do livro *Types and Programming Languages* de Benjamin Pierce (TPL) . O algoritmo para geração de constraints está na solução do exercício 22.3.10 e o algoritmo para unificação de tipos está na solução do exercício 22.4.6.