

DCC888 – Constraint Based Analyses

Nome: _____ Matrícula: _____

1. Este exercício refere-se ao programa abaixo, que foi escrito na linguagem funcional SML:

```
let f = fn x => x + 1
in let g = fn y => y + 2
    in let h = fn z => z + 3
        in (f g) + (f h)
```

Adicione rótulos ao programa, e encontre uma solução válida para a análise de fluxo de controle. Procure garantir que a sua solução satisfaça as restrições que definem a análise de fluxo de controle.

2. Uma solução da análise de fluxo de controle satisfaz uma aplicação de funções $(t_1^{l_1} t_2^{l_2})^l$ se as seguintes regras são verdade:

$$(C, R) \models t_1^{l_1} \wedge (C, R) \models t_2^{l_2} \wedge \\ (\forall (fn\ x \Rightarrow t_0^{l_0}) \in C(l_1) : (C, R) \models t_0^{l_0} \wedge C(l_2) \subseteq R(x) \wedge C(l_0) \subseteq C(l))$$

Porém, essas regras não levam em consideração a ordem em que expressões são avaliadas. Por exemplo, em uma linguagem com passagem de parâmetros por valor, uma expressão como $E_1 E_2$ requer primeiro a avaliação de E_1 antes de procedermos para a avaliação de E_2 . Entretanto, se E_1 não produz nenhum *closure*, então não precisamos avaliar E_2 .

- (a) Como poderíamos modificar a regra acima para avaliar o operando l_2 somente se o operador l_1 produz um *closure*?
- (b) Como poderíamos modificar a regra acima para avaliar o operando l_2 somente se o operador l_1 produz um *closure* $fn\ x \Rightarrow t_0^{l_0}$ em que a variável x apareça em t_0 ?
3. Prove que o algoritmo de resolução de restrições baseado em um grafo de restrições termina. Utilize a técnica da *saturação*. O algoritmo utiliza algumas estruturas de dados para guardar informação. Essas estruturas irão guardar uma quantidade finita de informação durante a execução do algoritmo. Uma vez que informação é adicionada a estrutura de dados, essa nunca mais é removida de lá.
4. Em uma linguagem orientada a objetos, é interessante podermos inferir quais os possíveis implementações de um método m , quando encontramos uma chamada tal qual $o.m(a, b, c)$. Uma forma de resolvermos esse problema é implementarmos uma análise de fluxo de controle. Porém, existem formas mais rápidas (porém menos precisas) de encontrarmos os alvos da chamada. Discuta diferentes maneiras de encontrarmos as possíveis implementações de m que não demandem uma análise de fluxo de controle.