

Resoluções provas antigas

Ap1 2024

Questão 1)

Item A)

respondido na lista (lembrando que as funções estão trocadas).

Item B)

O estilo mais natural e intuitivo é Sum2, que vai somando N ao seu antecessor recursivamente, até chegar em 0, retornando a soma de todos os elementos no intervalo $[0, N]$, muito mais semelhante a uma definição matemática. Já no caso de Sum1, há a utilização de um acumulador S , que guarda os valores das somas e é retornado quando N chega a 0, sendo um artifício para tornar o programa mais eficiente com a chamada recursiva pela cauda, que permite o compilador tratar as chamadas recursivas de forma semelhante a laços de repetição, sem aumentar o tamanho da pilha de recursão.

Questão 2)

Itens A) e B)

Uma referência externa diz respeito às variáveis usadas em um procedimento, mas que foram definidas fora de seu escopo. Quando o procedimento é mapeado no ambiente, seu valor, no store, é salvo em um par que contém tanto o código do procedimento quanto seu ambiente contextual (CE), que guarda todos os identificadores livres do procedimento que foram definidos fora de seu escopo. Esse par procedimento-CE é o chamado closure, ou valor procedimento, e é fundamental para programas que trabalham com escopo estático.

```

local SpecialMax Max3 Max5 in
  SpecialMax = proc {$ Value ?SMax}
    SMax = fun {$ X}
      if X>Value then
        X
      else
        Value
      end
    end
  end
local A B in
  A = 3
  B = 5
  {SpecialMax A Max3}
  {SpecialMax B Max5}
end
end

```

Máquina abstrata:

Início

$((\langle s \rangle, \emptyset), \emptyset) \Rightarrow$

$((\langle \text{SpecialMax} = \text{proc}\{\$ \text{Value } ?\text{SMax}\} \langle s1 \rangle \text{ end} \rangle \langle s2 \rangle,$
 $\{\text{SpecialMax} \rightarrow \text{sm}, \text{Max3} \rightarrow \text{m3}, \text{Max5} \rightarrow \text{m5}, \text{Value} \rightarrow \text{v},$
 $\text{SMax} \rightarrow \text{m}\}), \{\text{sm} = (\text{proc } \{\$ \text{Value } ?\text{SMax}\} \langle \dots \rangle \text{ end},$
 $\{\emptyset\}), \text{m3}, \text{m5}, \text{v}, \text{m}\}) \Rightarrow$

Executando $\langle s2 \rangle$ (chamadas do procedimento)

$((\langle \{\text{SpecialMax A Max3}\} \rangle, \{\text{SpecialMax} \rightarrow \text{sm}, \text{Max3} \rightarrow \text{m},$
 $\text{A} \rightarrow \text{v}\}), \{\text{sm} = (\text{proc} \dots), \text{v} = 3, \text{m}\}) \Rightarrow$

Criação da função Max3

$((\langle \text{SMax} = \text{fun } \{\$ X\} \langle s \rangle \text{ end} \rangle, \{\text{Max3} \rightarrow \text{m}, \text{A} \rightarrow \text{v},$
 $\text{X} \rightarrow \text{x}\}), \{\text{m} = (\text{Max3} = \text{fun } \{\$ X\} \text{ if } \text{X} > 3 \text{ then } \text{X} \text{ else } 3$
 $\text{end}, \{\text{Value} \rightarrow \text{v}\}), \text{v} = 3, \text{x}\}) \Rightarrow (\dots)$

Acima, vemos que a função Max3, por meio do procedimento SpecialMax, foi criada. Além disso, vale dizer que as partes grifadas correspondem as closures, o primeiro sendo o closure de SpecialMax, com ambiente contextual vazio, e o segundo sendo o closure de Max3, com ambiente contextual Value->v, visto que Value é um identificador livre em Max3. No caso de Max5, o resultado é análogo: o construtor SpecialMax irá criar uma função cuja variável Value vale 5, isto é, uma função que determinará qual dos números, X ou 5, é maior.

Ap1 2023

Questão 1)

```
local MulByN N in
  N = 3
  MulByN = proc {$ X ?Y}
    Y=N*X
  end

  {MulByN A B}
```

Considerando {A->10, B->x1, N->4}

Máquina abstrata

Início

$((\langle s \rangle, \emptyset), \emptyset) \Rightarrow$

$((\langle \text{MulByN} = \text{proc } \{ \$ X ?Y \} \text{ Y} = \text{N} * \text{X end} \rangle, \{ \text{MulByN} \rightarrow m, \text{N} \rightarrow n, \text{X} \rightarrow x, \text{Y} \rightarrow x1 \}), \{ m = (\text{proc } \{ \$ X ?Y \} \text{ Y} = \text{N} * \text{X end} \rangle, \{ \text{N} \rightarrow n \}), n=3, x, x1 \}) \Rightarrow$

Executando a chamada

```
(((<{MulByN A B}>, {MulByN->m, A->x, B->x1, N->n'})),
{m=(proc {$ X ?Y} Y = N*X end>, {N->n})), n=3, n'=4,
x=10, x1})) =>
```

Aqui, a operação que está dentro do procedimento é executada.

```
((<Y=N*X>, {A->x, B->x1, N->n'})), {m=(proc {$ X ?Y} Y
= N*X end>, {N->n})), n=3, n'=4, x=10, x1=30)
```

O parâmetro X vale 10, visto que A, que foi atribuído a 10, foi passado como seu argumento, e N vale 3, uma vez que, apesar de haver o mapeamento N->4 no ambiente da chamada, há o mapeamento N->3 salvo no ambiente contextual na função devido ao closure. Daí, devido ao escopo estático, o cálculo realizado é 3*10, resultando em Y=B=30.

Caso o escopo fosse dinâmico, o valor utilizado para N seria o valor mais recente que N recebeu, isto é, N=4. A mudança a ser realizada na máquina abstrata para suportar o escopo dinâmico é simplesmente armazenar apenas o código da função no closure, sem seu ambiente contextual. Assim, o programa utilizaria o valor mais recente atribuído à variável N.

Questao 2)

Respondido na lista.