Teoria da Computação

Exame de Época Especial

Universidade da Beira Interior

Quinta-Feira 1 de Setembro de 2005 das 9h30 às 12h00 - sala 6.06

A consulta dos apontamentos da disciplina (**e só esses**) é tolerada. Proibido o uso de calculadora e de telemóvel. Qualquer fraude implica reprovação na disciplina. Só serão corrigidas as provas **legíveis**.

Relembramos que, na tradição da axiomática de Peano, a notação $\mathbb{N}$ refere-se ao conjunto dos naturais incluindo o 0. Referiremo-nos ao conjunto dos naturais sem o 0 por $\mathbb{N}^*$.

1 Conjuntos, Relações, Ordens e Conjuntos Ordenados, Reticulados e CPOs

Sejam A o conjunto $\{2^n \mid n \in \mathbb{N}\}$ e $\mid$ $(\subseteq \mathbb{N}^2)$ a relação de divisibilidade ($a \mid b \triangleq$ *a divide b*). Admita que:

1. $(A, |)$ é um conjunto totalmente ordenado;
2. $(A, |)$ é um conjunto bem fundado;
3. $(A, |)$ é um reticulado.

Demonstre ou refute a seguinte afirmação: $(A, |)$ é um CPO.

2 Funções Recursivas, Indução Bem Fundada e Pontos Fixos

Seja f a seguinte função OCaml.

```
let rec f a l a =
match l with
  []  > a
| el::li  >
     if el > e then f e li a
       else f e li el::a
```

Demonstre a terminação da função f.

3 Indução Estrutural

Neste exercício vamos considerar uma definição indutiva das fórmulas da lógica proposicional.

Seja $\mathcal{V} \triangleq \{P, Q, R, S, \ldots\}$ um conjunto numerável de variáveis chamadas variáveis proposicionais. Seja $\mathcal{C}$ o conjunto de conectivas $\{\wedge, \vee, \rightarrow, \neg, \perp, \top\}$. O conjunto $\mathcal{P}rop$ das fórmulas proposicionais é definido como o menor subconjunto X do monoíde livre $(\mathcal{V} \cup \mathcal{C} \cup \{``('', ``)''\})^*$ verificando os (B) e (I) seguintes:

(B):
1. Para todo o $x \in \mathcal{V}$, x pertence a X
2. $\top$ pertence a X
3. $\perp$ pertence a X

(I):
1. Seja F uma fórmula de X ($F \in X$), então $\neg F \in X$
2. Sejam F e G duas fórmulas de X (i.e. $F, G \in X$), então $(F \wedge G) \in X$
3. Sejam F e G duas fórmulas de X, então $(F \vee G) \in X$
4. Sejam F e G duas fórmulas de X, então $(F \rightarrow G) \in X$

1. Dê o princípio de indução associada a esta definição indutiva;
2. Seja $npe : \mathcal{P}rop \rightarrow \mathbb{N}$, a função que devolve o número de parêntesis esquerdos da fórmula em parâmetro. De forma semelhante, seja $npd : \mathcal{P}rop \rightarrow \mathbb{N}$, a função que devolve o número de parêntesis direitos da fórmula em parâmetro. Demonstre que $\forall x \in \mathcal{P}rop, npe(x) = npd(x)$.

4 Dedução Natural

Demonstre, em dedução natural a seguinte tautologia:

$$(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (B \rightarrow C))$$

5 Cálculo λ

O cálculo λ original não é fortemente normalizável. Isto é, existem termos cuja redução por $\rightarrow_\beta^*$ não termina. No entanto, o cálculo λ subjacente ao sistema COQ (o Cálculo das Construções Indutivas) é por sua vez fortemente normalizável. Em termos de modelos da computação, qual é a importância deste teorema? As respostas esperadas poderão por exemplo estabelecer:

- a relação entre este teorema e o conceito da decidibilidade;
- ou a relação entre este teorema e o problema da paragem;
- ou ainda a relação entre o poder computacional dos diferentes cálculos e este teorema.

6 OCaml

Considere o tipo das árvores binárias e a função de percurso do programa seguinte:

```
type 'a bin_tree =
    Empty
  | Node of 'a bin_tree * 'a * 'a bin_tree;;


let rec percurso ab =
  match ab with
      Empty > []
    | Node (e,a,d) > percurso e @ [a] @ percurso d
;;
```

Proponha uma versão recursiva terminal (*tail recursive* em inglês) da função *percurso*. Uma possibilidade é escrever a função `percurso_tail_rec (abl:'a bin_tree list) (acc: 'a list)`, tal que:

$$percurso\ ab = percurso\_tail\_rec\ [ab]\ [\,]$$

Neste contexto *abl* arquiva a lista das subárvores que ainda falta tratar e *acc* acumula o estado actual do percurso.