TEORIA DA COMPUTAÇÃO
ENSAIO DE ORQUESTRA

RAUL H.C.LOPES

1. Indução

A notação de pontos de Curry, apresentada em sala e notas de aula e página 34 de [1], é utilizada para reduzir o uso de parênteses.

Uma seqüência de elementos de um tipo A é uma seqüência vazia (denotada por $\perp$) ou resultado de adicionar um elemento de A a uma seqüência de elementos de A. Os axiomas a seguir definem seqüências de elementos de A, denotada $seq.A$:

$$\perp \in seq.A \tag{1}$$

$$(a \triangleleft x) \in seq.A \Leftarrow x \in seq.A \wedge a \in A \tag{2}$$

$$\text{fecho universal.} \tag{3}$$

Assumindo que $x, y \in seq.A$ e que $a, b \in A$, os seguintes axiomas tratam a igualdade sobre seqüências.

$$\perp \neq (a \triangleleft x) \tag{4}$$

$$(a \triangleleft x) = (b \triangleleft y) \ .=.\ a = b \wedge x = y \tag{5}$$

O princípio da indução sobre seqüências estabelece que se P é propriedade sobre seqüências:

$$(\forall x : x \in seq.A : P.x) = \tag{6}$$

$$(P(\perp) \wedge (\forall x : x \in seq.A : \forall a : a \in A : P.x \Rightarrow P.a \triangleleft x)) \tag{7}$$

As equações a seguir definem o operador *snoc* (denotado $\triangleright$, em oposição, logicamente ao operador *cons*, $\triangleleft$).

$$\perp \triangleright a \ .=.\ a \triangleleft \perp \tag{8}$$

$$(a \triangleleft x) \triangleright b \ .=.\ a \triangleleft (x \triangleright b) \tag{9}$$

As equações a seguir definem um operador *cat* (denotado $\hat{}\hat{}$) para concatenar duas seqüências.

$$\perp \hat{}\hat{}\, y = y \tag{10}$$

$$(a \triangleleft x) \hat{}\hat{}\, y = .a \triangleleft (x \hat{}\hat{}\, y) \tag{11}$$

Pertinência em uma seqüência é definida a seguir.

$$a \notin \perp \tag{12}$$

$$a \in (b \triangleleft x) =. \; a = b \vee a \in x \tag{13}$$

(14)

Questão 1. *Defina uma relação de subseqüência tal que x é uma subseqüência de y, denotado $x \subseteq y$, quando x é obtido de y retirando-se alguns elementos (não necessariamente consecutivos) de y e preservando-se a ordem dos que restam.*

$$2 \triangleleft 4 \triangleleft 3 \triangleleft \perp \subseteq 1 \triangleleft 0 \triangleleft 2 \triangleleft 4 \triangleleft 2 \triangleleft 3 \triangleleft 2 \triangleleft \perp$$
$$(1 \triangleleft \perp) \subseteq (1 \triangleleft \perp)$$

Prove que $(x \subseteq y) \wedge (y \subseteq z) \;.\!\Rightarrow (x \subseteq z)$

Questão 2. *Considere a seguinte definição.*

$$sort.\perp = \perp \tag{15}$$

$$sort.a \triangleleft x = (sort.lsplit(a, x))^\frown(a \triangleleft sort.rsplit(a, x)) \tag{16}$$

Defina o operador $\nearrow$ tal que $(x) \nearrow$ quando x é não decrescente. Defina $lsplit$ e $rsplit$ tal que

$$(sort.x) \nearrow$$

Prove.

Questão 3. *Uma seqüência x é uma permutação de uma seqüência y quando x é seqüência que contém exatamente os mesmos elementos que y, possivelmente em uma ordem diferente.*

Em relação às definições da questão 2, prove que $sort.x$ é uma permutação de x.

2. Computabilidade

Questão 4. *Prove que o* halting problem *é insolúvel para a **LRAM**.*

Questão 5. *Define o conceito de produtividade máxima de uma **LRAM** de n nós. Prove que não é possível construir uma **LRAM** que cálcule a produtividade máxima de qualquer **LRAM** de n nós.*

Questão 6. *Mostre que o conjunto de funções computáveis via **TM** é recursivo.*

Uma máquina de Turing não-determinística (**NDTM**) é aquela em o conjunto de transições estado a estado é dado por uma relação.

Questão 7. *Defina formal **NDTM**.*

Questão 8. *Prove que qualquer função computável por uma* ***NDTM*** *é computável por uma* ***TM****.*

Uma linguagem L é **semi-decidível** se existe uma uma máquina de Turing tal que para qualquer string $x \in L$, M, dado x como input, termina em uma configuração padrão.

Questão 9. *Defina formamalmente L* ***é semi-decidível.***

Uma linguagem é decidível quando é construir uma máquina de Turing que, dado qualquer x, decide se x pertence a L.

Questão 10. *Defina formalmente L* ***é decidível.***

Questão 11. *Prove que* ***SAT*** *é decidível.*

Questão 12. *Prove que* ***FOL*** *é semi-decidível.*

Questão 13. *Defina ser (semi)decidível em tempo polinomial por uma* ***NDTM****.*

¡++¿
Dado um grafo, um circuito Hamiltoniano (**HC**) é um caminho que, partindo de um vértice, passa exatamente por uma vez por cada vértice do grafo e volta ao ponto inicial.

Questão 14. *Prove que o problema de* ***HC*** *é decidível em tempo polinomial.*

Considere um grafo que tem associada uma função que atribui pesos não negativos a cada uma de suas arestas. O peso de qualquer sub-grafo do mesmo é a soma dos pesos das arestas nele presentes.

Questão 15. *Prove que é possível construir uma* ***NDTM*** *que decide em tempo polinomial se existe um* ***HC*** *de custo menor do que algum* W.

Questão 16. *Construa, ou mostre que não é possível cosntruir, uma* ***NDTM*** *que calcula o custo do menor* ***HC*** *de um grafo dado.*

Questão 17. *Construa, ou mostre que não é possível construir, uma* ***NDTM*** *que calcula o* ***HC*** *de menor custo para um grafo dado.*

Referências

[1] Haskell B. Curry, *Foundations of mathematical logic*, Dover Publications, Inc., 1977.