Lógica e Computabilidade 2023.2

Hugo Nobrega

Lista de Exercícios Extra

As entregas podem ser feitas em duplas, mas lembre-se que
não poderá haver repetição de duplas em listas diferentes!
Entregue todas as questões marcadas com * até o fim do período:
23 de dezembro às 23:59[1]

Questão 1. Diga se cada uma das linguagens abaixo é decidível (i.e., reconhecida por alguma máquina de Turing) ou não, e prove sua resposta.

*** a.** PASSOS $:= \{w \in \{0,1\}^* \mid w$ codifica alguma máquina M, alguma entrada x para M e algum $n \in \mathbb{N}$, tais que M para em no máximo n passos quando executada com entrada $x\}$

*** b.** FINITO $:= \{w \in \{0,1\}^* \mid w$ codifica máquina M tal que M aceita apenas uma quantidade finita de possíveis entradas$\}$

c. EQUIVALENTES $:= \{w \in \{0,1\}^* \mid w$ codifica máquinas M e N que aceitam exatamente as mesmas entrada$\}$

Questão 2. Seja $f : (\{0,1\}^*)^n \to \{0,1\}^*$ uma função n-ária parcial (i.e., uma função cujas saídas são sempre elementos de $\{0,1\}^*$ e cujos argumentos de entrada são n elementos de $\{0,1\}^*$, para algum $n > 0$, mas não necessariamente *todas* as n-uplas desse tipo são argumentos possíveis). Dizemos que uma máquina M *computa* f se:

- M tem como alfabeto de entrada $\{0, 1, \texttt{\#}\}$;
- M tem pelo menos 2 fitas: uma fita *de entrada* e uma fita *de saída*, mais alguma quantidade finita (talvez zero) de fitas *de rascunho*;
- no início de qualquer execução de M, todas as fitas exceto a de entrada estão vazias;
- quando $(w_0, \ldots, w_{n-1}) \in (\{0,1\}^*)^n$ é uma entrada possível para f, então M chega a um estado terminal quando executada com $w_0\texttt{\#}w_1\texttt{\#}\cdots\texttt{\#}w_{n-1}$ inicialmente na fita de entrada;
- quando $(w_0, \ldots, w_{n-1}) \in (\{0,1\}^*)^n$ é uma entrada possível para f, e M é executada com $w_0\texttt{\#}w_1\texttt{\#}\cdots\texttt{\#}w_{n-1}$ inicialmente na fita de entrada, então quando M chega ao seu estado terminal, o conteúdo da fita de saída é exatamente $f(w_0, \ldots, w_{n-1})$.

[1] Essa lista foi alterada em 19 de dezembro, modificando a Questão 4.

Note: quando $(w_0, \dots, w_{n-1}) \in (\{0,1\}^*)^n$ **não** é uma entrada possível para f e M é executada com $(w_0, \dots, w_{n-1})$ inicialmente na fita de entrada, **nada afirmamos** sobre o comportamento de M (não importa o que acontece nesse caso)! Além disso, a máquina pode usar no seu alfabeto de fita quaisquer outros símbolos além de $0, 1, \#, B$ ("blank", símbolo representando o vazio).

Prove que as funções a seguir são computáveis:

a. "Soma em base unária": $f(1^x, 1^y) = 1^{x+y}$. Aqui, f só tem como entradas pares de palavras sem ocorrências de 0.

b. "Dobro em base unária": $f(1^x) = 1^{2x}$. Aqui, f só tem como entradas palavras sem ocorrências de 0.

* **c.** "Sucessor em base binária": $f(w)$ é o binário que representa o sucessor do número natural cuja representação binária é w. Aqui, f tem como entrada qualquer w não-vazia.

* **d.** "Dobro em base binária": $f(w)$ é o binário que representa o dobro do número natural cuja representação binária é w. Aqui, f tem como entrada qualquer w não-vazia.

* **e.** "Fatorial em base unária": $f(1^n) = 1^{n!}$, sendo $n!$ o *fatorial* de n. Aqui, f só tem como entradas palavras sem ocorrências de 0.

***Questão 3.** Dê exemplo de função parcial $f : (\{0,1\}^*)^n \to \{0,1\}^*$ que não seja computável, para algum $n \in \mathbb{N}$ (e prove que ela não é computável).

Questão 4 (Busy Beaver, o "castor ocupado")**.** É fato (e você não precisa provar!) que para cada $n \in \mathbb{N}$, existe uma quantidade finita de máquinas de Turing M satisfazendo:

- M tem exatamente n estados *além* do seu estado terminal;
- M tem apenas 1 fita;
- M tem alfabeto de entrada $\varnothing$ (vazio) e de fita $\{1, B\}$;
- quando executada com sua fita vazia (i.e., com B em todas as células),

M **sempre chega ao estado terminal** após um número finito de passos.

Assim, para cada $n \in \mathbb{N}$, há uma máquina com essas propriedades que deixa uma quantidade *máxima* de 1s escrita na fita ao parar (não necessariamente todos juntos); chamamos de *número de Busy Beaver de n*, denotado $\mathsf{BB}(n)$, essa quantidade (algumas fontes usam a notação $\Sigma(n)$ para o que estamos denotando por $\mathsf{BB}(n)$). Até hoje só se conhecem os valores exatos de $\mathsf{BB}(n)$ para $n \leq 4$. Por exemplo, $\mathsf{BB}(3) = 6$, $\mathsf{BB}(4) = 13$, $\mathsf{BB}(5) \geq 4098$, e $\mathsf{BB}(6)$ certamente não caberia nessa página: ele é maior do que o número

$$10 \uparrow\uparrow 15 := 10 \text{ elevado a } (10 \text{ elevado a } (10 \text{ elevado a } (\dots \text{ elevado a } 10)))$$

com 15 números 10 aparecendo na expressão!

* **a.** Encontre (e prove que estão corretos) os valores de $\mathsf{BB}(0)$ e $\mathsf{BB}(1)$.

* **b.** Prove que $\mathsf{BB}(2) \geq 4$.