Universidade Federal do Rio Grande do Sul
Instituto de Informática
Teoria da Computação N - INF05501

Definição dos Trabalhos

Prof. Alleff Dymytry
Prof. Rodrigo Machado
Semestre: 2024/1

1 Programação em Máquina de Registradores Norma

Entrega: até DOMINGO, ~~14/ABRIL~~ 21/ABRIL, 23:59h

Instruções: utilize o **Simulador de Máquina Norma** disponível em

`http://www.inf.ufrgs.br/~rma/simuladores/norma.html`

para desenvolver as rotinas pedidas abaixo. Cada programa deve ser nomeado

`<nro questao><nro item>.mn`

Exemplo: `1a.mn`, `1b.mn`, `2a.mn`, ...(preste atenção em maiúsculas e minúsculas)

Envie (via Moodle da turma) um arquivo .ZIP contendo todos os programas desenvolvidos, junto com um arquivo de texto indicando os componentes do grupo. Os grupos devem ter 3 ou 4 integrantes. Somente um componente do grupo deverá fazer a submissão (pelo grupo inteiro). O simulador novo de Máquina Norma (`https://www.inf.ufrgs.br/pet/pinguim/norma/`) pode ser utilizado para desenvolvimento, mas programas só serão considerados corretos se estiverem rodando corretamente no simulador original comentado primeiramente. Existem pequenas diferenças na programação dos dois simuladores.

1. Escreva programas que implementem as seguintes funções numéricas do tipo $\mathbb{N} \to \mathbb{N}$
   (a) $f(x) = x^2 + 3x$
   (b) $f(x) = \sum_{i=0}^{x} i^2$
   (c) $f(x) = \begin{cases} 1 & \text{se } x \text{ é múltiplo de 2} \\ 2 & \text{se } x \text{ não é múltiplo de 2 mas é múltiplo de 3} \\ 0 & \text{caso contrário} \end{cases}$
   (d) $f(x) = x!!$ onde $x!!$ é a operação de duplo fatorial descrita no seguinte link:
   `https://pt.wikipedia.org/wiki/Duplo_fatorial`
   Observação: note que $x!!$ **não é** a composição $(x!)!$ de duas funções fatoriais, e sim uma nova notação para função $(\ )!! : \mathbb{N} \to \mathbb{N}$

2. Utilizando a codificação de pares por um único número natural vista em aula e descrita a seguir:

$$cod(a, b) = 2^a \times 3^b$$

   crie programas para a máquina Norma que implementam as seguintes funções:
   (a) $\mathsf{pol}(x, y) = x^2 + 3y, \quad \mathsf{pol} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$
   (b) $\mathsf{div}(x, y) = \begin{cases} (q, r) & \text{se } x > 0 \text{ e } x = q * y + r \text{ para } (r < y) \\ (0, 0) & \text{se } x = 0 \text{ ou } y = 0 \end{cases}, \quad \mathsf{div} : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{N}$
   (c) $\mathsf{foo}(x) = (x - 1, \lfloor \frac{x}{2} \rfloor), \quad \mathsf{foo} : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$
   onde $\lfloor r \rfloor$ representa o maior número inteiro que é menor ou igual ao número real r.

3. Considere a sequência de números T_n, com $n \in \mathbb{N}$, definida pela seguinte recorrência: $T_0 = 2$, $T_1 = 5$, $T_n = 3T_{n-1} - 2T_{n-2}$ para $n > 1$. Por exemplo, $T_8 = 767$.
   (a) Construa um programa monolítico para máquina Norma que tenha $f(n) = T_n$ como função computada.

2 Programação em Cálculo Lambda

Entrega: até DOMINGO, 12/MAIO, 23:59h

Utilize o **Simulador de Cálculo Lambda** disponível em

`http://www.inf.ufrgs.br/~rma/simuladores/lambda.html`

para desenvolver as rotinas pedidas abaixo. O trabalho consistirá em um único arquivo nomeado

`trabalho.lam`

no qual os subitens de cada questão devem constar como definições no arquivo principal, com os **nomes definidos abaixo** (os casos de teste usarão os nomes mencionados – **não altere** os mesmos e **preste atenção** em maiúsculas e minúsculas, assim como na **ordem dos argumentos**). A expressão principal do programa não será considerada na correção.

Assuma que são utilizados numerais de Church para representação de naturais, e as codificações habituais de pares ordenados e listas para estruturas de dados. Quando houver múltiplos argumentos, assume-se que se utilizará a técnica de Currying (receber um argumento de cada vez sucessivamente) a menos que pares ordenados sejam explicitamente indicados (através de parênteses e vírgulas). O símbolo ␣ será utilizado para representar um espaço em branco.

Envie (via Moodle da turma) um arquivo .ZIP contendo o arquivo do programa desenvolvido, junto com um arquivo de texto indicando os componentes do grupo. Os grupos devem ter 3 ou 4 integrantes. Somente um componente do grupo deverá fazer a submissão (pelo grupo inteiro).

1. Escreva termos lambda que implementem as seguintes rotinas.

   (a) $\textbf{igual} \,␣\, a \,␣\, b = \begin{cases} \textbf{true} & \text{se } a = b \\ \textbf{false} & \text{se } a \neq b \end{cases}$

   (b) $\textbf{pol} \,␣\, a \,␣\, b = a^2 + 3b$

   (c) $\textbf{multTres} \,␣\, n = \begin{cases} \textbf{true} & \text{se } n \text{ múltiplo de 3} \\ \textbf{false} & \text{caso contrário} \end{cases}$

   (d) $\textbf{multFrac} \,␣\, (a, b) \,␣\, (c, d) = (ac, bd)$

   (e) $\textbf{somaFrac} \,␣\, (a, b) \,␣\, (c, d) = (ad + bc, bd)$

   (f) $\textbf{geraLista} \,␣\, n = \begin{cases} [\ ] & \text{se } n = 0 \\ [n, n-1, n-2, \ \ldots \ , 1] & \text{se } n > 0 \end{cases}$

   (g) $\textbf{multLista} \,␣\, l = \begin{cases} a_1 \times \cdots \times a_n & \text{se } l = [a_1, a_2, \ldots, a_n] \\ 0 & \text{se } l = [] \end{cases}$

   (h) $\textbf{seq} \,␣\, n = T_n = \begin{cases} 2 & \text{se } n = 0 \\ 5 & \text{se } n = 1 \\ 3T_{n-1} - 2T_{n-2} & \text{se } n > 1 \end{cases}$

3 Programação em Máquina de Turing

Entrega: até DOMINGO, 09/JUNHO, 23:59h

Instruções:

Utilize o **Simulador de Máquina de Turing** disponível em

`http://www.inf.ufrgs.br/~rma/simuladores/turing.html`

para desenvolver as rotinas pedidas abaixo. Cada programa deve ser nomeado

`<nro questao><nro item>.mt`

Exemplo: `1a.mt`, `1b.mt`, `2a.mt`, ...(preste atenção em maiúsculas e minúsculas no nome do arquivo e nos símbolos usados pelas máquinas de Turing)

Envie (via Moodle da turma) um arquivo .ZIP contendo o arquivo do programa desenvolvido, junto com um arquivo de texto indicando os componentes do grupo. Os grupos devem ter 3 ou 4 integrantes. Somente um componente do grupo deverá fazer a submissão (pelo grupo inteiro).

Considere a seguinte codificação de números naturais por strings de um único símbolo (unário):

$$\begin{array}{ccc} 0 & \mapsto & \varepsilon \\ 1 & \mapsto & a \\ 2 & \mapsto & aa \\ 3 & \mapsto & aaa \\ & \vdots & \end{array}$$

Similarmente, considere a codificação de pares de números naturais (m, n) através da justaposição das respectivas codificações em unário, utilizando-se o símbolo a para representar m e o símbolo b para o representar n:

$$\begin{array}{ccc} (2,3) & \mapsto & aabbb \\ (0,4) & \mapsto & bbbb \\ (3,3) & \mapsto & aaabbb \\ (0,0) & \mapsto & \varepsilon \end{array}$$

---

1. Para a função abaixo do tipo $\mathbb{N} \to \mathbb{N}$, desenvolva uma Máquina de Turing M sobre o alfabeto $\{a\}$ que a compute. O resultado deve ser deixado na fita em unário, utilizando exclusivamente o símbolo a:

   (a) $f(x) = \lfloor \frac{x}{2} \rfloor$
   Exemplos:
   $\langle M \rangle(aaaaaaaa) = aaaa$
   $\langle M \rangle(aaa) = a$
   $\langle M \rangle(a) = \varepsilon$

   (b) $f(x) = 2^x$
   Exemplos:
   $\langle M \rangle(\varepsilon) = a$
   $\langle M \rangle(a) = aa$
   $\langle M \rangle(aaa) = aaaaaaaa$

2. Para cada função abaixo do tipo $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$, desenvolva uma Máquina de Turing M sobre o alfabeto $\{a, b\}$ que a compute. O resultado deve ser deixado na fita em unário, utilizando exclusivamente o símbolo a:

   (a) $f(m, n) = 2m + n$
   Exemplos:
   $\langle M \rangle(aabbb) = aaaaaaa$
   $\langle M \rangle(abbbb) = aaaaaa$
   $\langle M \rangle(\varepsilon) = \varepsilon$

3. Para cada função a seguir do tipo $\mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{N}$, desenvolva uma Máquina de Turing M sobre o alfabeto $\{a, b\}$ que a compute.

   (a) $f(x, y) = (y + 2, 3x)$
   Exemplos:
   $\langle M \rangle(aabb) = bbbbaaaaaa$
   $\langle M \rangle(aaa) = bbaaaaaaaaa$
   $\langle M \rangle(ab) = bbbaaa$

4. Para cada alfabeto Σ e linguagem $\mathcal{L} \subseteq \Sigma^*$ especificados abaixo, desenvolva uma máquina de Turing M sobre Σ tal que

   $$\text{ACEITA}(M) = \mathcal{L} \ \wedge \ \text{LOOP}(M) = \varnothing$$

   (a) $\Sigma = \{0, 1\}, \ \mathcal{L} = \{0^m 1^n \mid m > 0 \ \wedge \ n > 0\}$
   (b) $\Sigma = \{a, b\}, \ \mathcal{L} = \{(ab)^n \mid n \text{ ímpar}\}$
   (c) $\Sigma = \{a, b, c\}, \ \mathcal{L} = \{a^n b^n c^n \mid n \geq 0\}$
   (d) $\Sigma = \{0, 1\}, \ \mathcal{L} = \{w \mid w \in \Sigma^* \ \wedge \ w = w^R\}$

4 Redução de problemas

Entrega: até DOMINGO, 30/JUNHO, 23:59h

O grupo deverá realizar as provas das seguintes reduções de problemas. Os textos contendo as provas **detalhadas** e **completas** deverão ser enviados via Moodle no questionário correspondente. Somente um componente do grupo deverá fazer a submissão (pelo grupo inteiro). Nos textos de cada redução deverá constar os nomes completos e números de matrículas de cada membro do grupo.

1. Considere os seguinte problemas de decisão:

    - Problema da Parada (PP).
    Entrada: um par (M, w), onde M é uma máquina de Turing sobre o alfabeto Σ e $w \in \Sigma^*$.
    Pergunta: $w \in (ACEITA(M) \cup REJEITA(M))$?
    - Problema da Aceitação da Palavra Vazia (PAPV).
    Entrada: uma máquina de Turing M sobre alfabeto Σ.
    Pergunta: $\varepsilon \in ACEITA(M)$?

    Prove que PAPV é um problema indecidível utilizando uma redução envolvendo PP.

2. Considere os seguinte problemas de decisão:

    - Problema da aceitação (PA).
    Entrada: um par (M, w), onde M é uma máquina de Turing sobre o alfabeto Σ e $w \in \Sigma^*$.
    Pergunta: $w \in ACEITA(M)$?
    - Problema da totalidade (PT).
    Entrada: uma máquina de Turing M sobre alfabeto Σ.
    Pergunta: a função computada $\langle M \rangle : \Sigma^* \rightharpoonup \Sigma^*$ é total?

    Escreva uma redução válida $r : PA \Rightarrow PT$.

3. Considere o seguinte problema de decisão:

    - Problema da Mesma Linguagem de Rejeição (PMLR).
    Entrada: um par (M_1, M_2) onde M_1 e M_2 são máquinas de Turing.
    Pergunta: $REJEITA(M_1) = REJEITA(M_2)$?

    PMLR é decidível ou indecidível? Prove sua resposta com uma redução.