Controle de Tráfego Urbano Inteligente

Trabalho Final do Curso de Teoria da Computação

João Roberto - 222217111
Caio Mello - XXXX
Vinicius - YYYYYYYYY
Ciclano - ZZZZZZZZ

Período Letivo
2025.2

Índice

1. Introdução . . . 3
   1.1. Enunciado do problema . . . 3
   1.2. Contextualização e importância na área de cidades inteligentes. . . . 3
   1.3. Abstrações, notações específicas e corolários significativos . . . 4
2. Máquina de uma única fita . . . 6
   2.1. Módulos e submódulos . . . 6
      2.1.1. Dentro do Módulo A . . . 6
      2.1.2. Função ψ . . . 7
   2.2. Máquina na íntegra . . . 7
   2.3. Análise de complexidade . . . 7
   2.4. Exemplos . . . 7
3. Máquina de múltiplas fitas . . . 7
   3.1. Módulos . . . 7
   3.2. Máquina na íntegra . . . 7
   3.3. Análise de complexidade . . . 7
   3.4. Exemplos . . . 7

1. Introdução

1.1. Enunciado do problema

A Companhia Turing Traffic Systems está desenvolvendo um sistema de controle de tráfego urbano baseado em Máquinas de Turing para gerenciar os semáforos de uma cidade inteligente. O objetivo é determinar qual deve ser o próximo estado do semáforo principal de um cruzamento, considerando informações sobre o fluxo de veículos.

O cruzamento possui três avenidas que se encontram no mesmo ponto, e cada avenida possui sensores que contam o número de veículos aguardando em cada direção.

O sistema deve decidir a próxima cor do semáforo principal de acordo com a seguinte política:

***I** - Se uma avenida tiver mais da metade do total de veículos somados das três avenidas, ela terá prioridade, e o semáforo para essa avenida ficará verde, enquanto os outros ficarão vermelho.*

***II** Caso nenhuma avenida tenha mais da metade dos veículos, o sistema escolhe a avenida com maior número de veículos.*

***III** Em caso de empate, a avenida A tem prioridade sobre B, e B sobre C (ordem A > B > C).*

1.2. Contextualização e importância na área de cidades inteligentes.

lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero lero

1.3. Abstrações, notações específicas e corolários significativos

Como parte desse trabalho, temos a missão de modular Máquinas de Turing que solucionem o problema preposto. Para tal, recebermos como entrada uma string representando a distribuição de carros dentre as avenidas A, B e C, seguindo o formato:

$$\varphi_0 ... \varphi_{n-1} \varphi_n \mid \forall x \in \mathbb{N} \land x \leq n \leftrightarrow \varphi_x \in \{a, b, c\}$$

Onde cada carro na avenida A será uma letra 'a', da B, 'b', e C, 'c'.

Será útil também a definição de uma função para cálculo da cardinalidade de uma letra, construída segundo o descrito abaixo:

Seja w uma palavra e x uma letra

$$\gamma_x(w) : \Sigma^* \to \mathbb{N}$$

$$\gamma_x(w) = |\{i \in \{1, ..., |w|\} \land i = x\}|$$

Onde γ é a função de cardinalidade.

De forma direta, $\gamma_x(w)$ nos retorna o número de ocorrências da letra x na palavra w.

Durante este documento, iremos com frequência omitir o (w) para tornar a leitura mais leve.

A partir do enunciado da questão, abstraímos duas decisões cujo nossos automômatos terão de tomar:

1. Determinar a avenida com quantidade de carros superior a metade do total de automóveis em todas avenidas.
   1.1. Ou seja, **determinar a letra x cuja cardinalidade é superior ao tamanho total da string w de entrada**:

   $$x \in \{a, b, c\} \mid \gamma_x(w) > \frac{\gamma_a(w) + \gamma_b(w) + \gamma_c(w)}{2}$$

   1.2. Caso mais de uma avenida se qualifique, devemos respeitar a A sobre B, B sobre C e, por transitividade, A sobre C.
2. Determinar a avenida com a maior quantidade de carros.
   2.1. Ou seja, **determinar a letra x de maior cardinalidade na palavra w**:

   $$x, y, z \in \{a, b, c\} \land y \neq z \land z \neq x \land x \neq z$$

   $$\gamma_x(w) > \gamma_y(w) \land \gamma_x(w) > \gamma_z(w)$$

   2.2. Em caso de empate, devemos respeitar a ordem de prioridade definida em 1.2.

Também é importante considerar que a computação descrita em 2. só deve ocorrer caso haja empate em 1.2.

Para composição das propriedades descritas, consideramos algumas equivalências para viabilizar o processo de modelagem do automômato que são essenciais para compreensão e validação dele.

A primeira é centrada na propriedade 1.1:

$$\gamma_x(w) > \frac{\gamma_a(w) + \gamma_b(w) + \gamma_c(w)}{2}$$

$$2 * \gamma_x(w) > \gamma_a(w) + \gamma_b(w) + \gamma_c(w)$$

Dado que $x \in \{a, b, c\}$,

$$\gamma_x(w) \in \{\gamma_a(w), \quad \gamma_b(w), \quad \gamma_c(w)\}$$

Generalizando,

$$\text{Sejam} \quad x, y, z \in \{a, b, c\} \mid x \neq y \wedge y \neq z$$

$$2 * \gamma_x(w) > \gamma_x(w) + \gamma_y(w) + \gamma_z(w)$$

$$\gamma_x(w) > \gamma_y(w) + \gamma_z(w) \tag{I}$$

O que nos dá

$$x, y, z \in \{a, b, c\} \mid x \neq y \wedge y \neq z$$

$$\gamma_x(w) > \frac{\gamma_x(w) + \gamma_y(w) + \gamma_z(w)}{2} \leftrightarrow \gamma_x(w) > \gamma_y(w) + \gamma_z(w)$$

Portanto, basta que nosso automômato compute propriedade I para que também valha 1.1.

2. Máquina de uma única fita

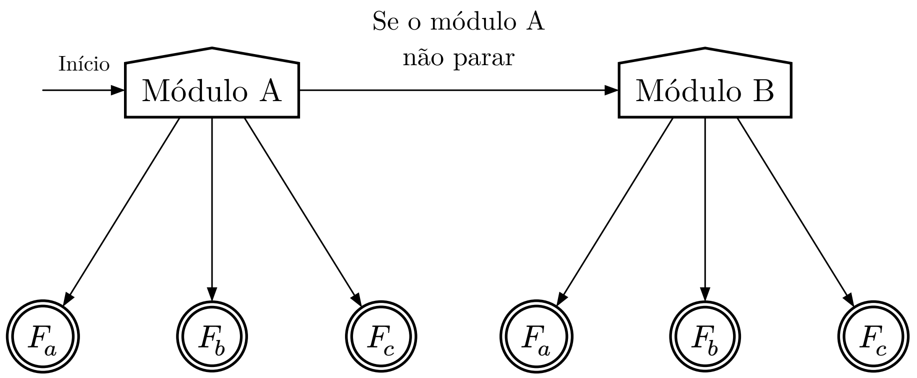
2.1. Módulos e submódulos

Para a máquina de uma fita, a qual nos referiremos por MT1, teremos dois módulos principais que chamaremos de A e B.

O módulo A será responsável por verificar a condição 1 do enunciado para alguma letra a, b ou c através da propriedade I.

Para tal, a máquina buscará parar num estado final que represente qual a decisão tomada.

Caso nenhum estado final seja alcançado dentro de A, o automômato recorrerá ao módulo B, onde avaliará a condição 2 do problema.



Fluxograma 1: Arquitetura do automômato

2.1.1. Dentro do Módulo A

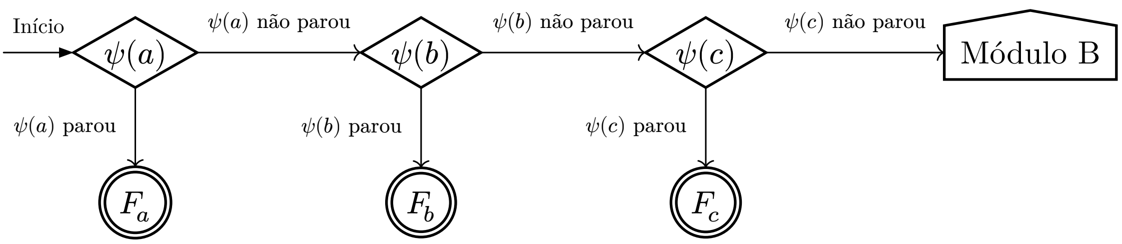
O módulo A por sua vez, irá sequencialmente analisar se a propriedade I vale para cada letra em $\{a, b, c\}$. Essa operação será feita linearmente, uma vez para cada símbolo até que a condição de parada seja cumprida ou os símbolos se esgotem.

Neste último caso entendemos que ninguém cumpriu com as condições necessárias para parada e recorremos ao módulo B.

Dessa forma, abstraímos um submódulo ψ tal que:

$\psi(x)$: Computa se x cumpre a propriedade I, caso positivo, o automômato pára.

Verificaremos na ordem $\psi(a)$, $\psi(b)$, $\psi(c)$. Garantindo a condição de desempate definida em 1.2. Uma vez que caso alguma avaliação de ψ pare, as demais não serão computadas. Assim preservamos a ordem de prioridade nos empates.



Fluxograma 2: Formato do Módulo A

2.1.2. Função ψ

Essa parte da máquina deve verificar se vale:

$$\exists x, y, z \in \{a, b, c\} \text{ onde } x \neq y \land y \neq z \land x \neq z$$

Tal que,

$$\gamma_x(w) > \gamma_y(w) + \gamma_z(w)$$

Para isto, definimos dois conjuntos X e $\overline{X}$ da seguinte forma:

Sejam,

$$n \in \mathbb{N} \mid x, \gamma_n \in \{a, b, c\} \mid w \text{ é uma palavra.}$$

$$w = \gamma_0 * \ldots * \gamma_n$$

Temos,

$$X = \{\forall \gamma_i \in w \mid \gamma_i = x\}, \text{e}$$
$$\overline{X} = \{\forall \gamma_i \in w \mid \gamma_i \neq x\}$$

Vamos computar se,

$$|X| > |\overline{X}|$$

ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis

2.2. Máquina na íntegra

ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis

2.3. Análise de complexidade

ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis

2.4. Exemplos

ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis

3. Máquina de múltiplas fitas

3.1. Módulos

ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis

3.2. Máquina na íntegra

ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis

3.3. Análise de complexidade

ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis

3.4. Exemplos

ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis ipsi literis