

Universidade do Estado do Amazonas
Escola Superior de Tecnologia
Data: 9 de Maio de 2019
Professora: Elloá B. Guedes
Disciplina: Fundamentos Teóricos da Computação

PROJETO PRÁTICO III SIMULANDO MÁQUINAS DE TURING

1 Apresentação

Entende-se por *Máquina de Turing Universal* uma Máquina de Turing capaz de simular uma Máquina de Turing arbitrária sobre uma entrada ω qualquer. Para tanto, a Máquina Universal recebe como entrada em sua fita a descrição da Máquina a ser simulada e a respectiva entrada. Este é o princípio teórico que inspirou a Arquitetura de Von Neumann e que usamos até hoje!

Neste projeto prático, vamos usar o nosso computador como uma Máquina de Turing Universal, executando Máquinas de Turing sobre entradas! No nosso caso, vamos nos restringir à simulação de Máquinas de Turing decisoras, para evitar o caso de ter que lidar com Máquinas de Turing que entram em *loop infinito* sobre certas entradas.

Em nosso projeto, as Máquinas de Turing a serem simuladas são determinísticas cujo cabeçote pode mover-se para direita (D), esquerda (E) ou permanecer parado (P). Estas máquinas serão fornecidas sob a forma de dicionários contendo a função de transição (**delta**), o estado de aceitação (**aceita**) e o estado inicial (**inicial**). Todos os estados são numerados em ordem crescente a partir do zero. Seu objetivo então será simular esta máquina e determinar a saída produzida considerando certas entradas. Para fins de simplificação, será considerado o alfabeto de entrada $\Sigma = \{0, 1\}$ e o alfabeto da fita $\Gamma = \Sigma \cup \{x, \#, b\}$, em que b representa o branco na fita.

As entradas para este projeto são as seguintes:

1. **Dicionário.** Contém as informações essenciais da Máquina de Turing a ser simulada:

- **inicial:** mapeado para um inteiro correspondendo ao índice do estado inicial;
- **aceita:** mapeado para um inteiro correspondendo ao índice do estado de aceitação;
- **delta:** mapeada para uma lista de tuplas, em que cada tupla corresponde a uma instrução que a Máquina de Turing é capaz de realizar. Cada tupla é da seguinte forma: (x, y, u, v, w) , em que:
 - i. x é um inteiro, correspondendo ao estado atual da Máquina;
 - ii. y é um inteiro, correspondendo ao novo estado que a Máquina irá assumir;
 - iii. u é um símbolo de Γ , correspondendo ao conteúdo da fita na qual o cabeçote está posicionado;
 - iv. v é um símbolo de Γ , correspondendo ao novo conteúdo que deve ser escrito na fita na posição que o cabeçote está posicionado;

v. $w \in \{D, E, P\}$ corresponde à movimentação do cabeçote.

2. **Inteiro.** Representa a quantidade de palavras que serão fornecidas como entrada, uma de cada vez;
3. **Palavras.** Palavras de comprimento maior igual a 1, em que apenas uma palavra é disposta por linha. Cada palavra deverá ser computada, ter estado final da fita impresso na tela e o estado final “ACEITA” ou “REJEITA” deverá ser impressa. Não havendo transição a partir de um determinado estado e do conteúdo da fita onde o cabeçote está posicionado, deve-se rejeitar a entrada.

2 Exemplos de Entradas e Saídas

Entrada	Saída
<pre>{ 'inicial': 0, 'aceita': 1, ' delta: [(0,0,'0','1','D'),(0,0,'1','0','D'), ,(0,2,'b','b','E'), (2,1,'0','0','P'), (2,1,'1','1','P')] } 3 100 10101 000000000</pre>	<pre>011 ACEITA 01010 ACEITA 111111111 ACEITA</pre>

3 Observações Importantes

- Lembre-se, a entrada de dados é feita via `input` e a saída via `print`;
- Atenha-se exatamente ao padrão de entrada e saída fornecidos nos exemplos. Qualquer mensagem adicional na entrada ou na saída de dados pode culminar em incorretude;
- Para facilitar a leitura da entrada, é permitido utilizar o pacote `ast` e a função `literal_eval`;
- Em caso de plágio, todos os envolvidos receberão nota zero!

4 Prazos Importantes

- **Início.** 09/05/2019 as 13h (horário do servidor)
- **Encerramento.** 20/05/2019 às 23h55min (horário do servidor)