

Máquina de Turing Não Determinística (MTND)

Descrição

Implemente um algoritmo que simule uma Máquina de Turing Não Determinista. A entrada consiste da especificação de uma MTND e de um conjunto de palavras. A saída consiste de uma lista indicando 'S' caso a MTND reconheça a palavra em questão e 'N' caso contrário.

Observações:

1. Leitura e escrita na entrada/saída padrão.
2. Qualquer divergência na saída com relação ao formato especificado implicará em nota zero.
3. A implementação não pode fazer uso de recursão.
4. No código fonte, você deve documentar como você gerenciou o não determinismo.
5. Critério de reconhecimento: parada em estado final.

Entrada

Na primeira linha, há uma lista de estados. Na segunda linha, há o alfabeto de entrada. Na terceira linha, há o alfabeto da fita. Na quarta linha, há o símbolo especial que limita a fita à esquerda. Na quinta linha, há o símbolo branco da fita. Na sexta linha, há o número total n de transições. Para cada uma das n linhas seguintes, há uma quintupla $\langle a, b, c, d, e \rangle$ onde 'a' é o estado de origem, 'b' é o caractere a ser lido, 'c' é o estado de destino, 'd' é o símbolo a ser escrito e, por fim, 'e' é a direção, imóvel (I), esquerda (E) e direita (D). Em seguida, há um caractere informando o estado inicial. Em seguida, há uma lista de estados finais. Por fim, há uma lista de palavras de teste a ser reconhecida. Os itens da listas serão separados por espaço em branco. A palavra vazia é representada por *.

Saída

Seu programa deve imprimir para cada palavra de teste 'S' se a MTND reconhece a palavra ou 'N' caso contrário.

Exemplos

| Entrada | Saída |
|----------------------|-------|
| 0 1 2 3 4 | N |
| a b | S |
| A B * | N |
| < | N |
| * | N |
| 10 | S |
| 0 a 1 A D | |
| 1 a 1 a D | |
| 1 B 1 B D | |
| 1 b 2 B E | |
| 2 B 2 B E | |
| 2 a 2 a E | |
| 2 A 0 A D | |
| 0 B 3 B D | |
| 3 B 3 B D | |
| 3 * 4 * E | |
| 0 | |
| 4 | |
| * ab ba abb aab aabb | |

Relatório Técnico

Você deve apresentar um relatório técnico de pelo menos duas páginas que contenha pelo menos os seguintes tópicos e respondendo as perguntas abaixo:

1- Introdução: O quê? Qual a importância?

2- Projeto e Implementação do Algoritmo: como projetou o algoritmo, quais as estruturas de dados, como gerenciou o não determinismo.

3- Metodologia: qual metodologia de software utilizada, como realizou testes, como controlou versões.

4- Resultados e Conclusões: estudo empírico de tempo de execução para reconhecimento de palavras de vários tamanhos, máquinas com variados número de transições. Apresente um gráfico que ilustre o tempo de execução à medida que o tamanho da palavra ou do número de transições da máquina cresce, apresente uma regressão linear (veja que você terá de pensar em features, por exemplo, tamanho da palavra, número de transições) para prever o tempo de execução com base no tamanho da palavra (relembre os conceitos vistos em métodos estatísticos).

É imprescindível a análise com base em regressão linear. Sugiro a leitura da seção 1.4 deste livro: <https://www.dropbox.com/s/1rjvzl8xi13da8u/Robert%20Sedgewick%2C%20Kevin%20Wayne-Algorithms-Addison-Wesley%20Professional%20%282011%29.pdf?dl=0>

Enviar o relatório para meu e-mail: rlopes@ufrb.edu.br

O relatório vale 60% da nota

Não forneço casos de teste. Vocês precisam desenvolver habilidades de teste de software.