

ACH2043 - Introdução à Teoria da Computação

Exercício de Programação: Emulação de Autômato Finito Não Determinístico

Objetivo:

Desenvolver um programa emulador de autômatos finitos não-determinísticos (AFNs).

Seu programa deve receber um único arquivo-texto contendo as especificações de m AFNs

M_1, M_2, \dots, M_m e, para cada autômato M_i , um conjunto de cadeias de teste $w_{i,1}, w_{i,2}, \dots, w_{i,n_i}$.

Deverá executar os autômatos M_1, M_2, \dots, M_m sobre suas respectivas cadeias de teste e devolver um arquivo-texto de valores 0/1, indicando quais cadeias são aceitas (1) ou rejeitadas (0) pelo seu autômato correspondente.

Em outras palavras, seu programa deverá decidir se cada cadeia de teste $w_{i,j}$ pertence ou não à linguagem reconhecida por M_i , $i = 1 \dots m, j = 1 \dots n_i$.

O programa deverá ser desenvolvido em C ou em Java, e deverá ser executável via linha de comando do DOS.

Chamada do programa e especificações dos arquivos

A chamada do programa será:

Implementação em C:

afn.exe <ArqTeste> <ArqSaida>

Implementação em Java:

java afn <ArqTeste> <ArqSaida>

Implementação em Python:

python afn.py -i <ArqTeste> -o <ArqSaida>

<ArqTeste>:

Este arquivo conterá as especificações dos AFNs e suas respectivas cadeias de testes.

O arquivo será organizado em blocos de linhas, da seguinte maneira:

m (número de autômatos de teste)

Bloco de linhas com a especificação do autômato 1

Bloco de linhas com as cadeias de teste do autômato 1

Bloco de linhas com a especificação do autômato 2

Bloco de linhas com as cadeias de teste do autômato 2

...

Bloco de linhas com a especificação do autômato m

Bloco de linhas com as cadeias de teste do autômato m

A especificação de cada autômato será formada por um bloco de linhas no seguinte formato:

- A primeira linha será um cabeçalho contendo os campos: $q \ s \ t \ q_0 \ a$
onde q é o número de estados, s é o número de símbolos do alfabeto Σ_ϵ (*incluindo a cadeia vazia*), t é o número de transições e a é o número de estados de aceitação. Note que todos esses parâmetros são todos estritamente positivos.
- A segunda linha conterá os índices dos estados de aceitação.
- As demais linhas do bloco (linhas 3 a $t + 2$) conterão as especificações das transições, na forma:
 $\langle \text{índice estado corrente} \rangle \ \langle \text{índice símbolo} \rangle \ \langle \text{índice novo estado} \rangle$.

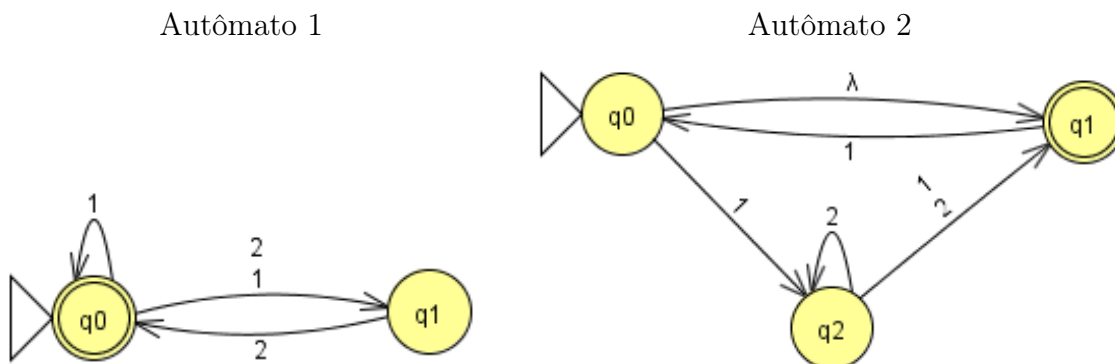
Observações:

- Os campos serão separados por espaços ou tabulação.
- Os estados serão indexados de 0 a $q - 1$, e os símbolos do alfabeto serão indexados de 0 a $s - 1$. A palavra vazia ϵ corresponderá ao índice 0.

A especificação das cadeias será formada por um bloco de linha no seguinte formato:

- A primeira linha conterá o número de cadeias de teste (inteiro positivo).
- Cada uma das demais linhas do bloco conterá uma cadeia, com seus símbolos separados por espaços. A cadeia vazia será representada por 0.

O exemplo abaixo ilustra um arquivo de testes referente aos dois automatos da figura abaixo:



```

2          numero de automatos de teste
2 3 4 0 1  cabecalho automato 1: 2 estados, 3 simbolos, 4 transicoes
0          estados de aceitacao
0 1 0      transicoes
0 1 1
0 2 1
1 2 0
6          numero de cadeias de teste do automato 1
1          1a cadeia de teste
1 1        2a cadeia
1 1 1
1 2 2 1 1 1 2 2 1
2 2 2 1
2 1 2 2
3 3 6 0 1  cabecalho automato 2: 3 estados, 3 simbolos, 6 transicoes
1          estados de aceitacao
0 0 1      exemplo de transicao com cadeia vazia
1 1 0
0 1 2
2 2 2
2 1 1
2 2 1
8          numero de cadeias de teste do automato 1
0          cadeia de teste 1 (cadeia vazia)
1          cadeia de teste 2
1 2 2 2 2 2 1
1 1 1 2 1 1 1
1 2 1 1
1 2 2 1 2 2
2
1 1 2 2 1 2

```

<ArqSaida>:

O arquivo de saída conterá seqüências de 0s e 1s separados por espaço, indicando a aceitação (1) ou rejeição (0) das cadeias pelos respectivos autômatos. Cada linha conterá os resultados dos testes das cadeias de um autômato.

Para arquivo de entrada ilustrado acima, o arquivo de saída resultante seria composto pelas seguintes linhas

```

1 1 1 1 0 0      Aceitacoes/rejeicoes das cadeias de teste do automato 1
1 1 1 1 1 0 0 0  Aceitacoes/rejeicoes das cadeias de teste do automato 2

```

Entrega do trabalho:

Condições da entrega:

- O trabalho poderá ser feito em duplas, devidamente identificados na primeira linha do código-fonte.
- A entrega deverá ser realizada através do ambiente e-disciplinas, até 13/10/2020. O projeto deverá ser entregue em um diretório compactado (formato ZIP), nomeado da seguinte forma: d<numerouspaluno1>_<numerouspaluno2>

Esse diretório deverá conter:

- O programa-fonte (em Java, Python ou C) com nome afn.c, afn.py ou afn.java
- Para entregas em C ou Java, a respectiva versão compilada, de nome afn.exe ou afn.class.
- Para entregas em Python, um arquivo-texto intitulado packages.txt com os nomes de todos os pacotes externos que deverão ser instalados via comando "pip install"(um nome de pacote por linha).
- O código-fonte deverá ser compilável no gcc ou no Eclipse-Java/Python.
- Não é necessário que os dois alunos da dupla enviem o mesmo EP, basta um envio por dupla.
- Dúvidas a respeito das especificações ou a respeito da implementação do trabalho serão sanadas até o dia 06/10/2020. Dúvidas encaminhadas após este prazo serão ignoradas.
- Além da correção do programa, será considerada a qualidade da documentação do código fonte.
- Se houver evidência de plágio entre trabalhos de grupos distintos de qualquer uma das turmas, os mesmos serão desconsiderados.