

Depurando Autômatos Finitos

O objetivo deste trabalho é aplicar os conhecimentos obtidos sobre autômatos finitos para construir uma ferramenta que simule a execução de um autômato finito sobre uma palavra de entrada. A idéia para a realização deste trabalho consiste em imprimir, na saída padrão, as configurações instantâneas que o processamento de uma palavra produz em um autômato finito.

1 Idéia geral e funcionamento do trabalho

A ferramenta a ser implementada como parte deste trabalho deve ser capaz de ler uma especificação de um AFD / AFN de um arquivo de entrada e imprimir, na saída padrão, as configurações produzidas como parte da execução do AF especificado. Sua ferramenta deve funcionar em linha de comando e aceitar um nome de arquivo como parâmetro de entrada.

O formato do arquivo descrevendo a entrada para a ferramenta é apresentado a seguir.

1.1 Especificação do arquivo de entrada

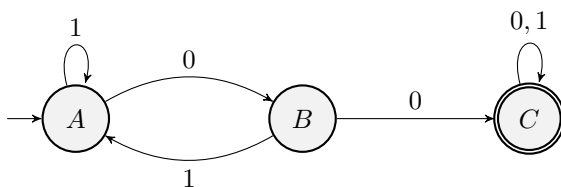
O arquivo de entrada deve ser um arquivo de texto simples. A primeira linha conterá os nomes de todos os estados iniciais, separados por um ou mais caracteres em branco seguido de um ponto e vírgula e os nomes de todos estados finais também separados por espaços em branco.

As demais linhas do arquivo contém a descrição de transições no formato:

`<estado de origem> <símbolo do alfabeto> <estado de destino>`

A linha indica que existe uma transição sobre o símbolo do alfabeto saindo do estado de origem para o estado de destino. É importante que símbolos sejam formados por apenas um caractere. As últimas linhas do arquivo especificarão as palavras que deverão ser processados pelo AF. Linhas para palavras de teste iniciam com o prefixo “test:”.

Como exemplo, considere o seguinte AF:



Este autômato é representado no formato de entrada para a ferramenta da seguinte forma:

```
A ; C
A 1 A
A 0 B
B 1 A
B 0 C
C 0 C
C 1 C
test: 1001
test: 10
test: lambda
```

Note que a especificação apresentada para o autômato anterior possui três palavras para simulação. Quando houver mais de uma palavra especificada, a ferramenta deverá executar o AF sobre cada uma destas. Observe que a palavra vazia é especificada pela string λ .

1.2 Resultado esperado

A ferramenta deverá produzir como resultado todas as configurações instantâneas da execução de um AF sobre uma palavra de entrada. Um exemplo da saída esperada para o AF especificado anteriormente e a palavra 1001 seria:

```
Iniciando o processamento de 1001
Configuração inicial: [A, 1001]
Executando a transição A -1-> A resulta em [A, 001]
Executando a transição A -0-> B resulta em [B, 01]
Executando a transição B -0-> C resulta em [C, 1]
Executando a transição C -1-> C resulta em [C,  $\lambda$ ]
C é um estado final.
A palavra 1001 é aceita pelo AF.
```

2 O que deve ser entregue?

Você deverá implementar essa ferramenta usando uma das seguintes linguagens: C (ou C++), Java, Haskell, Racket ou Python. Além de todo o código implementado, você deverá entregar um relatório que deve:

- Descrever a estrutura do código de sua implementação.
- Detalhes de como compilar o código produzido de sua implementação.
- Descrever testes realizados por você na implementação de sua ferramenta. Todos os arquivos produzidos por você para testar sua ferramenta deverão ser entregues junto com o relatório e seu código.

3 Considerações Finais

1. Esse trabalho deverá ser resolvido individualmente por cada um dos alunos.
2. Plágios não serão tolerados. Em caso de plágio, o professor irá tomar as medidas cabíveis de acordo com a legislação atual da universidade.
3. Todo o código produzido, em conjunto com um relatório em formato PDF, deverá ser entregue em um único arquivo .ZIP utilizando a tarefa “Entrega de trabalho prático” na plataforma Moodle.