

Trabalho 1: Autômatos Finitos com Saída — Máquinas de Mealy e de Moore

Linguagens Formais e Autômatos — Bacharelado em Sistemas de Informação

Prof. Jefferson O. Andrade

Campus Serra — Instituto Federal do Espírito Santo (Ifes)

2019/2

Sumário

1	Descrição do Trabalho	1
2	Sintaxe dos Arquivos de Entrada e Saída	2
2.1	Sintaxe para Definição de Máquina de Mealy	2
2.2	Sintaxe para Definição de Máquina de Moore	3
3	Sobre a Entrega do Trabalho	5
4	Sobre a Execução do Trabalho	6

1 Descrição do Trabalho

Este trabalho de programação sobre Linguagens Formais e Autômatos propõe a implementação dos algoritmos de conversão de autômatos finitos com saída vistos em sala de aula. Mais especificamente o trabalho propõe a implementação de:

1. Conversão de Máquina de Mealy para Máquina de Moore equivalente.
2. Conversão de Máquina de Moore para Máquina de Mealy equivalente.

As definições das máquinas de Mealy e de Moore, tanto de entrada quanto de saída do programa devem ser lidas (na entrada) e gravadas (na saída) de acordo com o formato definido na Seção 2.

O programa resultante da implementação deste trabalho deve ser um programa “de linha de comando”, ou seja, não deve ter interface gráfica de qualquer natureza. Será construído para ser ativado do prompt de comando do sistema operacional (Linux) de acordo com a seguinte sintaxe:

```
$ programa -i <arquivo-de-entrada> -o <arquivo-de-saída>
```

Caso o arquivo de entrada contenha a definição de uma máquina de Mealy, o programa automaticamente gerará a máquina de Moore equivalente e gravará a sua definição no arquivo de saída indicado; e vice-versa.

Caso não seja possível ler uma definição correta de uma máquina de Mealy ou de uma máquina de Moore do arquivo de entrada (erro de sintaxe no arquivo de entrada), deve ser gerada uma mensagem de erro, e o programa deve encerrar.

2 Sintaxe dos Arquivos de Entrada e Saída

As definições das máquinas de Mealy e de Moore serão lidas de arquivos em formato texto simples. Os formatos são pré-definidos e caso haja erro de formatação no arquivo de entrada o programa deve emitir uma mensagem de erro e abortar o processamento.

A primeira linha do arquivo de entrada terá apenas uma palavra, “mealy” ou “moore”, indicando o tipo de máquina que o arquivo descreve.

2.1 Sintaxe para Definição de Máquina de Mealy

O formato do arquivo texto para definição das Máquinas de Mealy será o seguinte:

- Linha 1: conterá a palavra **mealy**;
- Linha 2: conterá uma sequência de strings separadas por espaços representando os nomes dos estados. Por exemplo: **s0 s1 s2 s3 s4 sf**.
- Linha 3: conterá uma sequência de símbolos separados por espaços representando o alfabeto de entrada.¹ Por exemplo: **a b c**.
- Linha 4: conterá o nome do estado inicial.
- Linha 5: conterá uma lista com o nome dos estados finais, separados por espaços. Caso não haja estados finais, esta linha estará a string **--**.
- Linha 6: conterá uma sequência de símbolos separados por espaços representando o alfabeto de saída. Por exemplo: **X Y Z**.
- Linha 7 em diante: representam a função de transição, e cada linha terá o seguinte formato:
 1. nome do estado inicial
 2. símbolo da transição (da cadeia de entrada)
 3. nome do estado final
 4. lista de símbolos de saída (zero ou mais símbolos de saída)

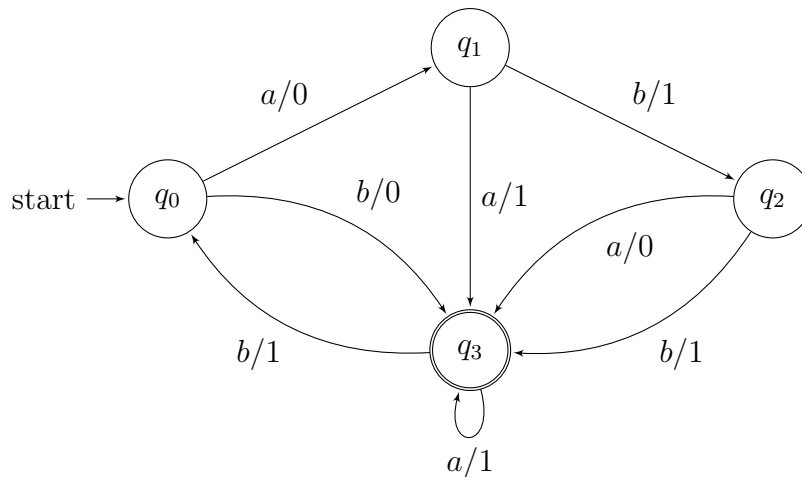


Figura 1: Exemplo de máquina de Mealy.

Seguindo a sintaxe dada acima, a definição da Máquina de Mealy mostrada na Figura 1, fica como mostrado abaixo:

```

1 mealy
2 q0 q1 q2 q3
3 a b
4 q0
5 q3
6 0 1
7 q0 a q1 0
8 q0 b q3 0
9 q1 b q2 1
10 q1 a q3 1
11 q2 a q3 0
12 q2 b q3 1
13 q3 b q0 1
14 q3 a q3 1

```

2.2 Sintaxe para Definição de Máquina de Moore

O formato do arquivo texto para definição das Máquinas de Moore será o seguinte:

- Linha 1: conterá a palavra **moore**;
- Linha 2: conterá uma sequência de strings separadas por espaços representando os nomes dos estados. Por exemplo: **s0 s1 s2 s3 s4 sf**.
- Linha 3: conterá uma sequência de símbolos separados por espaços representando o alfabeto de entrada.² Por exemplo: **a b c**.

¹Note que, embora seja comum que os símbolos do alfabeto sejam compostos por um caractere, isso não é uma regra é perfeitamente válido um alfabeto com símbolos como: **zero um dois tres muitos**.

²Note que, embora seja comum que os símbolos do alfabeto sejam compostos por um caractere, isso não é uma regra é perfeitamente válido um alfabeto com símbolos como: **zero um dois tres muitos**.

- Linha 4: conterá o nome do estado inicial.
- Linha 5: conterá uma lista com o nome dos estados finais, separados por espaços. Caso não haja estados finais, esta linha conterá a string “--” (sem as aspas).
- Linha 6: conterá uma sequência de símbolos separados por espaços representando o alfabeto de saída. Por exemplo: X Y Z.
- Linhas $7 \dots (7 + n)$: representam a função de transição, e cada linha terá o seguinte formato:
 1. nome do estado inicial
 2. símbolo da transição (da cadeia de entrada)
 3. nome do estado final
- Linha $(7 + n) + 1$: contém a string “-----” (sem as aspas). Esta linha funciona como um marcador para o final da definição da função de transição do autômato.
- Linha $(7 + n) + 2$ em diante: contém a definição da saída da Máquina de Moore. Deve haver uma linha para cada estado definido na linha 2 do arquivo. Cada linha terá o nome do estado no início, seguido por uma sequência de zero ou mais símbolos do alfabeto de saída, separados por espaço.

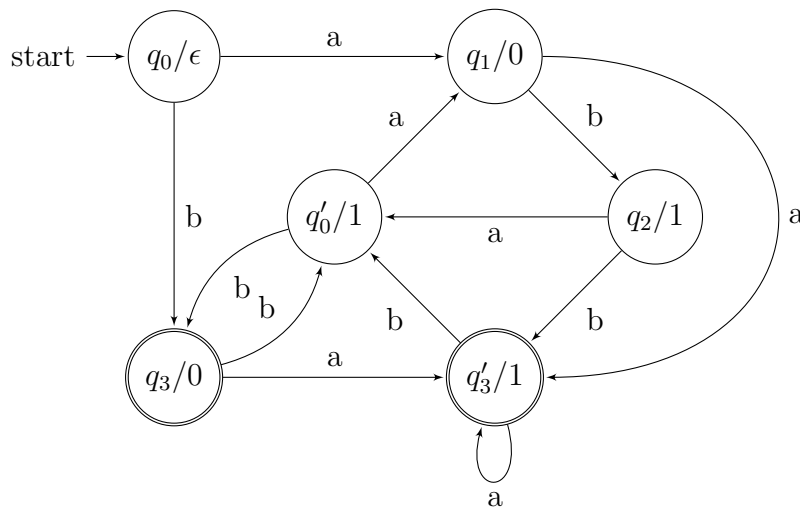


Figura 2: Exemplo de máquina de Moore.

Seguindo a sintaxe dada acima, a máquina de Moore ilustrada na Figura 2 pode ser definida como mostrado abaixo:

```

1 moore
2 q0 q0' q1 q2 q3 q3'
3 a b
4 q0
5 q3 q3'
6 0 1

```

```

7  q0 a q1
8  q0 b q3
9  q0' a q1
10 q0' b q3
11 q1 a q3'
12 q1 b q2
13 q2 a q0'
14 q2 b q3'
15 q3 a q3'
16 q3 b q0'
17 q3' a q3'
18 q3' b q0'
19 -----
20 q0
21 q0' 1
22 q1 0
23 q2 1
24 q3 0
25 q3' 1

```

3 Sobre a Entrega do Trabalho

O trabalho deve ser entregue, através do sistema Moodle institucional do Ifes. A data e horário limites para a entrega estarão definidas na atividade do Moodle.

Devem ser entregues:

- Todo o código fonte desenvolvido para o trabalho;
- Ao menos 3 exemplos de máquina de Mealy para teste;
- Ao menos 3 exemplos de máquina de Moore para teste;
- Um arquivo `Readme.md`, em formato Markdown, contendo:
 - Nome dos autores;
 - Uma descrição breve da estrutura do código fonte;
 - Procedimento para compilação do trabalho (se for o caso);
 - Nome e modo de uso do programa desenvolvido;
 - Outras informações que os autores julgarem apropriadas para o entendimento do trabalho realizado.
- Um arquivo de configuração de *build* para algum sistema de automação de *build* (ou automação de compilação), como por exemplo: *make*, *ant*, *maven*, *sbt*; ou *NAnt* e *xbuild* para Mono.

Note que alguns ambientes de desenvolvimento, tais como Eclipse e NetBeans, já geram automaticamente os arquivos de automação de compilação.

Note também que para linguagens interpretadas, como Python, por exemplo, é comum que não haja a necessidade do arquivo de *build*.

O trabalho deve ser entregue **obrigatoriamente** em um arquivo no formato ZIP contendo todos os itens descritos acima.

4 Sobre a Execução do Trabalho

- O trabalho deve ser desenvolvido para o sistema operacional Linux. O trabalho pode ser desenvolvido em outra plataforma, mas o professor irá testar o trabalho em Linux, portanto certifique-se de que o trabalho poderá ser executado em Linux sem problemas.
- O trabalho pode se desenvolvido **individualmente ou em duplas**.
- O trabalho pode ser desenvolvido em qualquer linguagem que esteja disponível para o sistema operacional Linux
- Em todos os casos, o aluno deve se certificar de que o código gerado seja compatível/portável para o sistema operacional Linux e que as instruções de compilação seja claras, precisas e completas o suficiente para serem seguidas pelo professor.
- No caso específico de linguagens como C# e F# (desenvolvidas originalmente para a plataforma **.Net**), o aluno deve se certificar de que o código gerado seja compatível com o sistema Mono.