

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**O DEPARTAMENTO DE INFORMÁTICA**  
**CIÊNCIA DA COMPUTAÇÃO**

**CAUÊ FELCHAR**

**TRABALHO 1: GAMES E AUTÔMATOS**

**CAUÊ FELCHAR**

**TRABALHO 1: GAMES & AUTÔMATOS**

Trabalho da disciplina Linguagens formais  
e Autômatos.

Orientador: Prof. Dr. Gleifer Vaz Alvez

## SUMÁRIO

O sumário deve ser colocado como último elemento pré-textual. O sumário deve ser apresentado conforme o que segue:

Os elementos pré-textuais não aparecem no sumário;

A ordem dos elementos do sumário deve ser conforme os indicativos das seções que compõem o trabalho;

O texto de um documento deve ser dividido em seções, sendo a seção **primária** a principal (1), seguida de outras divisões, como a seção **secundária** (1.1); seção **terciária** (1.1.1); seção quaternária (1.1.1.1) até a seção quinária (1.1.1.1.1), inclusive; Porém, recomenda-se limitar a numeração progressiva até a seção terciária apenas;

O espaçamento entre títulos no sumário deve ser simples.

Para atualizar as listas e o sumário, clique com o botão direito do mouse sobre o índice em Atualizar campo e selecione **Atualizar apenas o número de página** ou **Atualizar o índice inteiro**.

<b>1</b>	<b>INTRODUÇÃO</b>	<b>4</b>
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>4</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>11</b>

## 1 INTRODUÇÃO

Neste trabalho será estudado a modelagem e implementação de um Autômato finito a partir da interação entre dois personagens de um jogo, com todos os seus estados e funcionamentos, tal implementação poderá, então, ser utilizada para rodar quaisquer Autômatos finitos, tal como será produzido um diagrama de estados do jogo.

## 2 DESENVOLVIMENTO

Analisar o funcionamento de um sistema, real ou virtual requer de abstração, para fazê-lo observou qual era a interação do usuário com o vendedor:



Notou-se um padrão de até 4 mensagens, portanto modelou-se o alfabeto contendo 4 símbolos:  $e_0, e_1, e_2, e_3$ . Também notou-se, ao coletar dados, que não é possível completar o jogo somente falando com o personagem, também é necessário conseguir dinheiro, então, para não perder o foco da abstração, abstraiu-se o ato de conseguir dinheiro como o novo símbolo do alfabeto:  $m$ .

Para modelar os estados da interação utilizei da opção save do jogo e do VLC para gravar as interações, que por sua vez foram transcritas ao modelo do FSM, após estudos definiu-se os estados como:

- ss: Estado inicial do protagonista, onde o mesmo tem 2 escolhas.
  - m > s0s.
  - qualquer outra ação, inclusive a palavra vazia > s0.
- s0: primeira interação com o vendedor, ele diz uma frase terminada com: swen.
- s1: fala outra vez com o vendedor, ele termina dizendo: "nevermind".
- s2: fala com o vendedor, ele diz: "wanna talk business?", mas o protagonista não tem dinheiro.
- s3: fala com o vendedor, ele diz: "wanna talk business?", e o protagonista tem dinheiro.
- s4: O protagonista já comprou o mapa do vendedor, e sua interação com o mesmo está em seu estado final.
- nvm: O vendedor diz: "never mind" e oferece o mapa
- ofc: O vendedor diz: "of course" á pergunta: "is this some kind of code?" e similares
- cen: o vendedor diz "close enough" caso o protagonista responda bem ao código apresentado em ss.
- wb: o vendedor diz "wanna bet" ao protagonista, que jura que pode conseguir o mapa em outro lugar
- ltb: o vendedor diz: "lets talk business" e oferece o mapa ao protagonista, que, neste estado, não têm dinheiro para compra-lo
- boff: o vendedor diz: "Back off kid, it's bad for business".
- nemn: o vendedor diz "never mind" e acaba a conversa com o protagonista.
- wise: o vendedor diz "A wise choice" e vende o mapa ao protagonista.
- s0m: o mesmo que o s0, porém com dinheiro.
- s1m: o mesmo que o s1, mas com dinheiro.
- s3: o mesmo que s2, mas com dinheiro, o único em que temos uma opção diferente pelo dinheiro, onde o personagem principal pode comprar o mapa.
- nvmm: o mesmo que nvm, mas com dinheiro.
- ofcm: o mesmo que ofc, mas com dinheiro.
- cenm: o mesmo que cen, mas com dinheiro.

- wbm: o mesmo que wb, mas com dinheiro.
- ltbm: o mesmo que ltb, mas com dinheiro.
- boffm: o mesmo que boff, mas com dinheiro.
- nemnm: o mesmo que nem, porém com dinheiro.

Tendo os estados e o alfabeto, só restava a definição das transições, que, no formato do fsm, são definidas como:

- ss:m>s0m
- ss:e0>s0
- ss:e1>ss
- ss:e2>ss
- ss:e3>ss
- s0:e0>nvm
- s0:e1>nvm
- s0:e2>ofc
- s0:e3>cen
- cen:\$>ltb
- cen:e0>cen
- cen:e1>cen
- cen:e2>cen
- cen:e3>cen
- s1:m>s1m
- s1:e0>ltb
- s1:e1>ltb
- s1:e2>ltb
- s1:e3>wb
- wb:\$>s1
- wb:e0>wb
- wb:e1>wb
- wb:e2>wb
- wb:e3>wb
- nvm:\$>s1
- nvm:e0>nvm

- nvm:e1>nvm
- nvm:e2>nvm
- nvm:e3>nvm
- ofc:\$>nvm
- ofc:e0>ofc
- ofc:e1>ofc
- ofc:e2>ofc
- ofc:e3>ofc
- ltb:\$>s2
- ltb:e0>ltb
- ltb:e1>ltb
- ltb:e2>ltb
- ltb:e3>ltb
- s2:e0>boff
- s2:e1>nemn
- s2:e2>s2
- s2:e3>s2
- boff:\$>s1
- boff:e0>boff
- boff:e1>boff
- boff:e2>boff
- boff:e3>boff
- nemn:\$>s1
- nemn:e0>nemn
- nemn:e1>nemn
- nemn:e2>nemn
- nemn:e3>nemn
- wise:\$>s4
- wise:e0>wise
- wise:e1>wise
- wise:e2>wise
- wise:e3>wise

- s4:\$>s4
- s4:m>s4
- s4:e0>s4
- s4:e1>s4
- s4:e2>s4
- s4:e3>s4
- s0m:e0>nvmm
- s0m:e1>nvmm
- s0m:e2>ofcm
- s0m:e3>cenm
- cenm:\$>ltbm
- cenm:e0>cenm
- cenm:e1>cenm
- cenm:e2>cenm
- cenm:e3>cenm
- s1m:m>s1m
- s1m:e0>ltbm
- s1m:e1>ltbm
- s1m:e2>ltbm
- s1m:e3>wbm
- wbm:\$>s1m
- wbm:e0>wbm
- wbm:e1>wbm
- wbm:e2>wbm
- wbm:e3>wbm
- nvmm:\$>s1m
- nvmm:e0>nvmm
- nvmm:e1>nvmm
- nvmm:e2>nvmm
- nvmm:e3>nvmm
- ofcm:\$>nvmm
- ofcm:e0>ofcm



- ofcm:e1>ofcm
- ofcm:e2>ofcm
- ofcm:e3>ofcm
- ltbm:\$>s3
- ltbm:e0>ltbm
- ltbm:e1>ltbm
- ltbm:e2>ltbm
- ltbm:e3>ltbm
- s3:e0>boffm
- s3:e1>nemnm
- s3:e2>wise
- s3:e3>s3
- boffm:\$>s1m
- boffm:e0>boffm
- boffm:e1>boffm
- boffm:e2>boffm
- boffm:e3>boffm
- nemnm:\$>s1m
- nemnm:e0>nemnm
- nemnm:e1>nemnm
- nemnm:e2>nemnm
- nemnm:e3>nemnm

Dados o alfabeto, as transições, o estado inicial e final, podemos descrever a definição formal da quintupla ordenada:

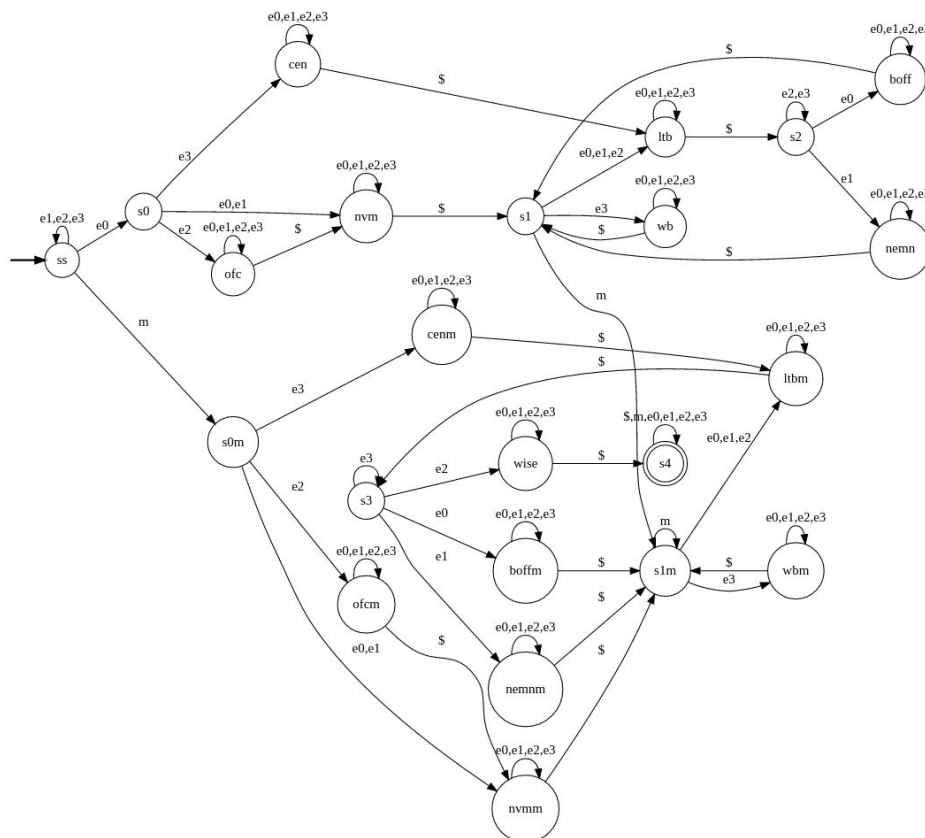
$M = ($

- $Q = \{ss, s0, s1, s2, s4, nvm, ofc, cen, wb, ltb, boff, nemn, wise, s0m, s1m, s3, nvmm, ofcm, cenm, wbm, ltbm, boffm, nemnm\},$
- $E = \{m, e0, e1, e2, e3\},$
- $\sigma = \text{descrito acima},$
- $q0 = ss,$

-  $F = \{s4\}$

)

O AFND resultante pode ser visto com o seguinte diagrama de estados:



A subsequente implementação do AFND de transições vazias foi feita em C. Ao invés de implementar autômato em si, implementou-se um interpretador de AFNDs com transições vazias, que aceita uma sintaxe do FSM levemente modificada.

As modificações a sintaxe do FSM foram as que seguem:

A primeira linha deve conter, respectivamente, o número de estados, o número de palavras no alfabeto, e o número de estados terminais.

Ao final da descrição das transições deve ser utilizada a sigla EOF para indicar o fim da descrição.

Após a descrição do autômato pode ser feita a inserção da palavra a ser processada, que deve ser separada por vírgulas.

Dada tal entrada o programa irá imprimir um traço de como foi a execução, dando os estados intermediários e símbolos lidos.

O programa imprimirá se a palavra foi aceita ou rejeitada.

### **3 CONCLUSÃO**

Ao modelar e interpretar o AFND com transições vazias de um jogo, consegui aprender sobre o que é importante se considerar em todos os passos de seu funcionamento e interpretação, fica mais claro entender o funcionamento de algo uma vez que existe uma implementação sua sobre o assunto, portanto foi importante fazer o projeto.