



Fundação Universidade Federal do ABC

Av. dos Estados, 5001, Santa Terezinha, Santo André/SP, CEP 09210-580

Projeto de Linguagens Formais e Autômatos referente ao primeiro quadrimestre de 2024.

**Alunos:**

André Huang Hong

RA: 11202021840

Kleverson Nascimento Pinto

RA: 11075716

Laís Kaori Y. Rodrigues

RA: 11201921254

**Professor:**

Prof. Alexandre Donizeti Alves

**Modelagem de autômato finito baseado em máquina de pegar pelúcia.**

Santo André

07 de Abril de 2024

**Sumário**

**Introdução..... 3**

**Descrição do problema..... 3**

**Aplicação..... 4**

**Resultados..... 6**

**Considerações finais..... 6**

**Referências..... 7**

## Introdução

Autômatos Finitos são modelos matemáticos poderosos utilizados para descrever sistemas computacionais simples, mas eficazes. De acordo com o livro "Introduction to the Theory of Computation" (Introdução à Teoria da Computação), de Michael Sipser, publicado em 2005, um autômato finito é definido como uma 5-tupla,  $M = (Q, \Sigma, \delta, q_0, F)$

Onde:

- $Q$  é um conjunto finito de estados,
- $\Sigma$  é um conjunto finito de símbolos de entrada,
- $\delta$  é a função de transição de estado, que mapeia  $Q \times \Sigma$  em  $Q$ ,
- $q_0$  é o estado inicial,  $q_0 \in Q$ ,
- $F$  é um conjunto de estados de aceitação,  $F \subseteq Q$ .

Neste projeto, exploramos a aplicação desses conceitos para criar uma versão virtual e interativa da popular máquina de pegar ursinhos de pelúcia como aquelas encontradas em parques de diversão e fliperamas. Essa máquina é um dispositivo mecânico controlado por um jogador que tenta manipular uma garra para pegar um urso de pelúcia de um recipiente cheio de objetos.

**Figura 1: Máquina de pelúcia**



**Fonte: Multivisi, 2024.**

O autômato finito será projetado para representar os estados possíveis da máquina, as transições entre esses estados e as condições de aceitação ou falha.

## Descrição do problema

O problema que escolhemos é a simulação de uma máquina de pegar ursinhos de pelúcia utilizando um Autômato Finito Determinístico (AFD). O desafio principal é projetar e implementar um autômato finito determinístico que modele o comportamento da garra, determinando se ela consegue ou não pegar o ursinho de pelúcia com base em certas condições e entradas.

As questões que precisamos considerar são:

- Escolha do modelo: Qual modelo de máquina será baseado o projeto?
- Modelagem do ambiente: Como representar o espaço onde os ursinhos de pelúcia estão localizados e a garra que tentará pegá-los?
- Definição do estado inicial e final: Quais são os estados iniciais e finais do autômato, e como eles estão relacionados ao sucesso ou fracasso em pegar um ursinho de pelúcia?
- Transições entre estados: Como as ações da garra (mover-se para frente, para trás, para direita, para esquerda) são traduzidas em transições de estado no autômato?
- Critérios de sucesso: Quais são os critérios que determinam se a garra conseguiu ou não pegar um ursinho de pelúcia?

## Aplicação

Neste projeto iremos simular uma máquina de pelúcia com a seguinte arquitetura:

**Figura 2: Tipo de máquina a ser simulada**



**Fonte: Rain Cahigas, 2024**

Para solucionar os problemas encontrados anteriormente, mapeamos a máquina vista de cima como um plano dividido em 25 posições, definindo 13 como a posição padrão inicial da garra e 23 como a abertura para passagem das pelúcias.

**Figura 3: Mapeamento da máquina vista de cima**

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Definimos o estado inicial sendo o estado em que a máquina não possui crédito para iniciar uma jogada e o estado final como o estado em que a garra consegue pegar o ursinho e o usuário da máquina retira o prêmio com sucesso. No total o autômato possui 27 estados sendo eles:

- Estado 0: estado inicial;
- Estados de 1 a 25: estados de posições dentro da máquina;
- Estado 26: estado final.

As ações processadas pela máquina formam o conjunto do alfabeto aceito pelo autômato, que incluem: inserir crédito, mover para esquerda, mover para direita, mover para frente, mover para trás e descer a garra.

As funções de transição de cada estado são:

	Mover para esquerda	Mover para direita	Mover para trás	Mover para frente	Descer garra	Inserir crédito
→0	0	0	0	0	0	13
1	1	2	6	1	0	1
2	1	3	7	2	0	2
3	2	4	8	3	0	3
4	3	5	9	4	0	4
5	4	5	10	5	26	5
6	6	7	11	1	0	6
7	6	8	12	2	26	7
8	7	9	13	3	0	8
9	8	10	14	4	0	9
10	9	10	15	5	0	10
11	11	12	16	6	0	11
12	11	13	17	7	0	12
13	12	14	18	8	0	13
14	13	15	19	9	0	14
15	14	15	20	10	0	15
16	16	17	21	11	0	16
17	16	18	22	12	0	17
18	17	19	23	13	0	18
19	18	20	24	14	0	19
20	19	20	25	15	0	20
21	21	22	21	16	0	21
22	21	23	22	17	0	22
23	22	24	23	18	0	23
24	23	25	24	19	26	24
25	24	25	25	20	0	25
26*	26	26	26	26	26	26

Utilizando Python, criamos uma classe que representa a máquina de pegar ursinhos de pelúcia, primeiramente implementamos uma função que randomiza a posição das pelúcias, excluindo a posição 23 que é a saída dos ursinhos na máquina. Para fins de exemplo, fixamos três posições de ursinhos dentro da máquina previamente como sendo as posições 5, 7 e 24.

Para os casos em que jogador desce a garra em posições que não há ursinhos, o retorno é sempre “false” pois não é possível pegar nenhum objeto. Em casos em que a garra desce na posição que existe uma pelúcia, há a possibilidade da garra pegar o ursinho (retornar “true”) e há possibilidade da garra não pegar o ursinho (retornar “false”) com uma chance de 50% para cada saída. Este cenário considera a possibilidade de existir diversos fatores que fazem a garra não conseguir pegar o objeto, como o posicionamento da pelúcia, precisão de encaixe da garra, entre outras variáveis.

Para os casos em que uma pessoa tenta inserir créditos no meio da jogada, sem estar no estado inicial, a máquina rejeita e continua no estado atual. Em casos que o jogador tenta iniciar uma jogada sem inserir créditos a garra permanece imóvel, assim como acontece no cenário real.

A imagem do autômato com as posições dos ursinhos em 5, 7 e 24 gerado pela biblioteca “automathon” pode ser visualizada no seguinte link:

<https://github.com/KleversonNascimento/projeto-lfa/blob/main/automato.png>

## Resultados

A partir do autômato modelado podemos simular como seria uma jogada em uma máquina de ursinhos. No momento inicial a máquina está parada e sem créditos (estado 0), a partir do momento que uma pessoa insere créditos, a máquina passa para o estado em funcionamento e a garra pode ser controlada partindo da posição definida como padrão (estado 13). Em seguida, a garra pode acessar qualquer posição dentro da máquina (estados de 1 a 25) utilizando os movimentos “mover para a esquerda”, “mover para a direita”, “mover para frente” e “mover para trás” em qualquer ordem e quantidade de movimentos.

Assim que o jogador escolhe uma posição e desce a garra, há 50% de chance dele conseguir pegar um ursinho e chegar no estado final (estado 26) e 50% de chance dele não conseguir pegar um ursinho e a máquina voltar para o estado inicial que só irá avançar para o próximo estado quando alguém inserir crédito para jogar.

Quando analisamos algumas palavras de entrada para o autômato com as posições das pelúcias fixadas em 5, 7 e 24 podemos determinar o seu comportamento:

- Entrada 1: {'inserir crédito', 'mover para direita', 'mover para trás', 'mover para trás', 'descer garra'} é aceita pelo autômato e tem chance de pegar ursinho.
- Entrada 2: {'inserir crédito', 'mover para direita', 'mover para direita', 'mover para trás', 'mover para esquerda', 'descer garra'} é aceita pelo autômato e não pega ursinho.
- Entrada 3: {'mover para direita', 'inserir crédito', 'mover para direita', 'mover para trás', 'mover para trás', 'descer garra'} é aceita pelo autômato e não pega ursinho.
- Entrada 4: {'mover para direita', 'mover para direita', 'mover para trás', 'mover para esquerda', 'descer garra'} não é aceita pelo autômato pois não inseriu crédito.
- Entrada 5: {'inserir crédito', 'mover para direita', 'mover para direita', 'mover para trás', 'mover para esquerda', 'descer garra', 'mover para trás'} não é aceita pelo autômato pois não é possível acessar nenhum estado a partir do estado final.

Desse modo, a linguagem aceita pelo autômato são palavras de qualquer tamanho que possuem pelo menos um “inserir crédito” e terminam com “descer garra”.

## Considerações finais

O projeto envolveu a exploração prática de Autômatos Finitos Determinísticos (AFDs) em uma simulação de máquina de pegar ursinhos de pelúcia. Ao criar um modelo de ambiente e definir estados, transições e condições de sucesso, pudemos desenvolver um autômato capaz de imitar o comportamento de garras enquanto procuram ursinhos de pelúcia. Testar o autômato com simulações mostrou sua capacidade de aceitar movimentos válidos e rejeitar movimentos inválidos, melhorando a compreensão prática dos conceitos teóricos. O uso de Python e bibliotecas relevantes simplificou o modelo de construção e ofereceu uma visão clara de como a máquina funciona. Esta

experiência prática foi essencial para aplicar o que aprendemos e mostrar o potencial de AFDs na modelagem de sistemas de computação simples – abrindo caminho para trabalhos futuros em teorias de computação e automação.

## **Referências**

[SIPSER 2005] SIPSER, M. - “Introdução à Teoria da Computação”, Editora Cengage, Tradução da 2ª edição norte-americana, 2005.