



Teoria da Computação e compiladores

Documentação Vending Machine

Máquina de Doces

Nome e RA dos Alunos: Grupo: 13

- Eduardo dos Santos Berlatto, RA: 821220735
 - Guilherme Silva da Costa, RA: 82426746
- Mariana Rosendo de Freitas, RA: 821236352
- Pedro Henrique Freire de Oliveira, RA: 821224361
 - Rafael da Silva Rodrigues, RA 822165658
 - Vitor Nunes Oliveira, RA: 821224765

Índice

Resumo	3
1. Introdução	3
2. Descrição do Problema	3
• Valores Aceitos	3
• Produtos Disponíveis	3
• Funcionalidades Exigidas	4
3. Fundamentação Teórica e Diagrama do Autômato	4
• Fundamentação Teórica	4
• Consulta de Estados	4
• Alfabeto	4
• Função de Transição	5
• Diagrama do Autômato	5
4. Metodologia e Implementação	5
• Ambiente de Desenvolvimento	5
• Estrutura de Arquivos	6
• Fluxo de Operação	6
5. Resultados e Discussão	6
6. Conclusão	7
7. Referências Bibliográficas	7

Resumo

Este documento apresenta a modelagem, implementação e avaliação de um sistema simulador de uma máquina de doce, desenvolvido com base em um Autômato Finito Determinístico (AFD). O sistema foi projetado para aceitar três tipos de notas/moedas (R\$1,00, R\$2,00 e R\$5,00) e fornecer três tipos de doces (A, B e C), com preços de R\$6,00, R\$7,00 e R\$8,00, respectivamente. A máquina ativa as opções de doces de acordo com o saldo acumulado e realiza a devolução do troco quando necessário.

A implementação foi realizada em Python utilizando a biblioteca Pygame para a interface gráfica, combinando conceitos teóricos e práticos de autômatos e animação.

Palavras-chave: Máquina de doce, Autômato Finito Determinístico, Pygame, Simulação, Troco.

1. Introdução

Com o avanço da tecnologia e a busca por soluções interativas para problemas do cotidiano, a modelagem de sistemas automatizados tem ganhado destaque tanto na pesquisa quanto na indústria. A presente documentação descreve o desenvolvimento de um simulador para uma máquina de doce, cujo funcionamento é orientado por um Autômato Finito Determinístico (AFD). A escolha deste formalismo permite uma abordagem clara e sistemática para o gerenciamento de estados e transições necessárias para a operação da máquina, englobando a aceitação de valores monetários, seleção de produtos e o processo de devolução do troco.

Este trabalho integra conceitos teóricos extraídos de referenciais clássicos sobre linguagens formais e autômatos com uma implementação prática voltada à criação de uma interface interativa e visualmente atrativa para o usuário.

2. Descrição do Problema

O projeto parte de um case onde se especifica uma máquina de doce que apresenta as seguintes características:

- **Valores Aceitos:** A máquina aceita apenas três tipos de notas/moedas: R\$1,00, R\$2,00 e R\$5,00.
- **Produtos Disponíveis:** São oferecidos três tipos de doces:
 - Doce A – R\$6,00
 - Doce B – R\$7,00
 - Doce C – R\$8,00
- **Funcionalidades Exigidas:**
 - Atualização dinâmica do saldo à medida que as notas são inseridas.

- Ativação das opções de seleção do doce conforme o valor acumulado.
- Liberação do doce escolhido e, se aplicável, devolução do troco.
- Animação que simula o doce sendo “disparado” pela máquina.

O sistema deve contemplar os finais de operação, que envolvem a entrega do doce com ou sem troco, conforme o valor inserido e o preço do produto selecionado.

3. Fundamentação Teórica e Diagrama do Autômato

3.1. Fundamentação Teórica

A utilização de um Autômato Finito Determinístico (AFD) para modelagem de sistemas comerciais, como máquinas de venda automática, é uma abordagem consolidada na literatura de linguagens formais e automóveis computacionais. Onde:

Estados representam os valores acumulados inseridos na máquina.

Transições ocorrem quando o usuário insere dinheiro.

Estados finais indicam a liberação do doce e do troco, quando aplicável.

3.2. Consulta de Estados

No presente projeto, o sistema é modelado considerando os seguintes estados:

- **Waiting:** Estado inicial, onde a máquina aguarda a inserção de valores.
q0: Estado inicial (saldo = R\$0,00).
- **Selecting:** Estado em que o sistema permite que o usuário selecione o doce de interesse, contanto que o saldo seja suficiente.
q1, q2, ..., q8: Estados intermediários representando os valores acumulados.
- **Dispensing:** Estado que indica a liberação do doce.
qA, qB, qC: Estados finais para a liberação dos doces sem troco.
- **Returning Change:** Estado em que, após a compra, se realiza a devolução do troco para o usuário.
qA', qB', qC': Estados finais para a liberação dos doces com troco.

A modelagem garante que cada transição seja determinada pela entrada do usuário, conforme as regras de negócio definidas.

3.3. Alfabeto

O alfabeto será composto pelos valores das notas:

$$\Sigma = \{1, 2, 5, A, B, C\}$$

1	Simboliza inserção de R\$1,00 no autômato
2	Simboliza inserção de R\$2,00 no autômato
5	Simboliza inserção de R\$5,00 no autômato
A	Simboliza a escolha do doce A
B	Simboliza a escolha do doce B
C	Simboliza a escolha do doce C

3.4. Função de Transição

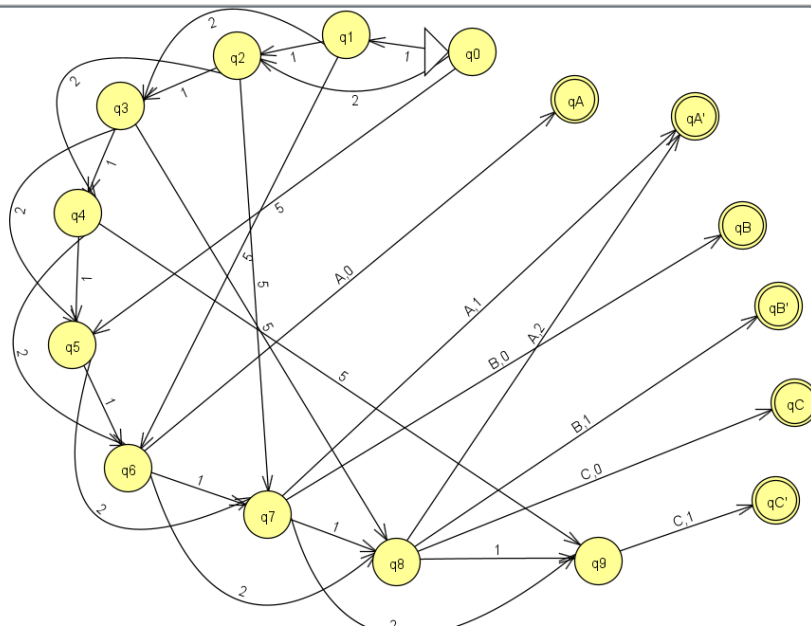
As transições ocorrerão conforme o valor acumulado:

- **Exemplo:** $q_0 \xrightarrow{(1)} q_1$, $q_1 \xrightarrow{(2)} q_3$, $q_5 \xrightarrow{(1)} q_6$ (doce “A” liberado)

3.4. Diagrama do Autômato

O diagrama elaborado pelo grupo representa graficamente os estados e as transições do sistema. Para tanto, foram considerados os valores acumulados, a ativação da seleção do doce e a eventual devolução do troco.

Cada transição é disparada por ações do usuário, como a inserção de notas ou a escolha do doce. O diagrama foi validado em um simulador de autômatos disponível online, garantindo a conformidade com a solução proposta no case.



4. Metodologia e Implementação

4.1. Ambiente de Desenvolvimento

O sistema foi desenvolvido utilizando a linguagem Python, com a biblioteca **Pygame** aplicada para a criação de uma interface gráfica interativa.

Essa escolha possibilitou a implementação de uma animação que simula o funcionamento real da máquina, incluindo a inserção de dinheiro, atualização do saldo, seleção do doce e a visualização do processo de entrega (representado por uma animação de queda do produto).

4.2. Estrutura de Arquivos

O projeto está organizado em, pelo menos, dois arquivos principais:

- **main.py:** Responsável pela interface gráfica, tratamento dos eventos do usuário e integração com as funcionalidades da máquina. Este arquivo define a tela, carrega a imagem de fundo (a máquina propriamente dita) e implementa a lógica dos botões para inserção de dinheiro e seleção dos tipos de doce. Além disso, gere a animação do doce caindo (simulada pela movimentação de um objeto gráfico).
- **machine.py:** Contém a definição da classe `CandyMachine`, que gerencia os estados da máquina, o saldo acumulado, a devolução do troco e o estoque dos doces. Métodos importantes incluem:
 - `insert_money(amount)`: Adiciona o valor inserido ao saldo e verifica se a máquina deve passar para o estado de seleção.

- `select_candy(candy)`: Verifica se o saldo é suficiente para o doce escolhido, atualiza o estoque e aciona os estados de dispensa e eventual devolução de troco.
- `return_change()`: Gerencia a devolução do troco, se houver saldo remanescente após a compra.
- `reset_machine()`: Função para resetar o estado da máquina para a configuração inicial.

4.3. Fluxo de Operação

1. **Inserção de Dinheiro:** O usuário interage com botões que representam as notas de R\$1,00, R\$2,00 ou R\$5,00, acumulando o saldo na máquina.
2. **Seleção do Doce:** Uma vez que o saldo seja igual ou superior ao valor do doce desejado, os botões correspondentes aos tipos de doces (A, B, e C) são ativados, permitindo que o usuário faça a sua escolha.
3. **Processamento da Compra:** Ao selecionar um doce, o método `select_candy` é disparado, que valida a operação, atualiza o estoque e define o estado adequado (se a compra ocorre sem ou com troco).
4. **Entrega e Troco:** A animação é acionada para simular a liberação do doce na tela. Se houver saldo remanescente, este é devolvido ao usuário através da função `return_change`.
5. **Feedback Visual e Reinicialização:** O saldo, o tipo de doce comprado e o valor de troco (se houver) são exibidos na interface. Após o ciclo, a máquina retorna ao estado "Waiting".

5. Resultados e Discussão

A implementação do projeto evidenciou a aplicabilidade prática dos conceitos teóricos de autômatos no desenvolvimento de aplicações interativas.

A integração entre a modelagem do AFD e a implementação utilizando Python/Pygame permitiu:

- **Visualização Interativa:** A interface gráfica demonstra de forma intuitiva cada etapa do processo, conectando a teoria com a prática, o que enriquece a experiência do usuário.
- **Gerenciamento de Estado:** O uso da classe `CandyMachine` facilitou a manutenção e a compreensão do fluxo de estados (inserção, seleção, liberação e devolução), garantindo que as transições acontecem de forma coerente e previsível.
- **Feedback em Tempo Real:** A exibição do saldo, a ativação dos botões de seleção de doce e a animação associada à liberação do produto contribuem para um sistema que se mostra didático e funcional.

Durante o desenvolvimento, foram realizadas várias validações para assegurar a robustez do sistema, como a verificação do estoque disponível e a validação do saldo antes da efetivação da compra. Essa abordagem metódica contribuiu para a confiabilidade da solução proposta.

6. Conclusão

O presente projeto demonstra como a modelagem de um Autômato Finito Determinístico pode ser aplicada para resolver problemas reais, como o de uma máquina de doce interativa. Através da implementação em Python com o uso da biblioteca Pygame, foi possível criar uma simulação que une a teoria dos autômatos com uma interface amigável e intuitiva.

Recomenda-se que trabalhos futuros explorem a inclusão de funcionalidades adicionais, como a possibilidade de recarga do estoque e a implementação de efeitos sonoros que possam enriquecer ainda mais a experiência do usuário. A metodologia aplicada neste projeto pode servir como base para o desenvolvimento de outras aplicações interativas que demandem modelagens precisas e controle de estados.

7. Referências Bibliográficas

1. **DELAMARO, Márcio.** *Linguagens Formais e Autômatos*, 1998.
2. **HOPCROFT, John E.; ULLMAN, Jeffrey D.** *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley, 1979.
3. **MENEZES, Paulo Blauth.** *Linguagens Formais e Autômatos*, Sagra Luzzatto, 1998.