

Teoria dos Autômatos

Conceitos Gerais e Práticos

Euclides Palma Paim

Programa Interdisciplinar de Pós-Graduação em Computação Aplicada

Universidade do Vale do Rio dos Sinos – Unisinos

São Leopoldo, Brazil

euclidespaim@gmail.com

Abstract—This document is a short explanation of automata theory, presents concepts about the theory of computation. We abroad formal definitions of deterministic and non-deterministic automaton and illustrate a practical example of modeling that uses a finite automata. This paper describes a hypothetical implementation of a system cash machine and its operations.

Index Terms—automata, finite state machines, theory of computation.

I. INTRODUÇÃO

Teoria dos autômatos é um ramo da ciência da computação que estabeleceu suas raízes durante o século passado quando matemáticos desenvolveram tanto teórica como literalmente máquinas que imitavam certas características humanas realizando cálculos mais rápidos e precisos [1]. A palavra autômato esta intrinsicamente ligada com a palavra automação e indica execução automática de processos específicos relacionados à produção. Em outras palavras, a teoria dos autômatos lida com a lógica computacional de máquinas, conhecidas como autômatos. Através dos autômatos cientistas da computação são capazes de compreender como máquinas computam informações e resolvem problemas ou como uma função pode ser definida como computável ou dedutível.

A teoria dos autômatos é o estudo de máquinas abstratas e autômatas e como os problemas computacionais podem ser mais bem resolvidos combinando o uso de diferentes tipos de modelos computacionais, representa de forma finita uma linguagem formal que pode ser um conjunto infinito

II. TEORIA DOS AUTÔMATOS

A. Objetivos

O objetivo maior da teoria dos autômatos é desenvolver métodos pelos quais cientistas de computadores podem descrever e analisar de forma dinâmica o comportamento de sistemas discretos, nos quais sinais são periodicamente exibidos [2]. O comportamento desses sistemas discretos é determinado pela maneira que o sistema é construído para armazenamento e combinação de elementos. Algumas características dessas máquinas incluem: entradas, saídas e estados. Entradas podem ser definidas como sequências de símbolos selecionadas de um conjunto finito I .

A saber, conjunto I é o conjunto de $x_1, x_2, x_3 \dots x_k$ onde k é o número de entradas. Saídas são sequências de símbolos selecionadas de um conjunto infinito Z . A saber, conjunto Z é o conjunto de $y_1, y_2, y_3 \dots y_m$ onde m é o numero de saídas. E estados um conjunto finito Q no qual a definição depende do tipo de automação.

B. Principais Autômatos

Há quatro grandes famílias de autômatos:

- Máquina de estado finito.
- Autômatos de pilha.
- Autômatos com memória linear.
- Máquina de Turing.

As famílias de autômatos acima podem ser interpretadas de uma forma hierárquica, onde a máquina de estado finito é o autômato mais simples e a máquina de Turing o mais complexo. Uma máquina de Turing é uma máquina de estado finito ainda que o inverso não seja verdadeiro.

III. DEFINIÇÃO

Um autômato finito tem algumas partes descritas a seguir. Ele tem um conjunto de estados e regras para ir de um estado a outro, dependendo dos símbolos de entrada. Tem um alfabeto de entrada que indica os símbolos permitidos. Tem um estado inicial e um conjunto de estados aceitos. A definição formal diz que um autômato finito é uma lista de cinco objetos: conjunto de estados, alfabeto de entrada, regras de movimento, estado inicial e estados aceitos. Em linguagem matemática essa lista é frequentemente chamada de 5-tuple. Assim definimos autômatos finitos como uma tupla contendo essas cinco partes.

A. Autômato Finito

Um autômato é representado formalmente por uma tupla quintupla $(Q, \Sigma, \delta, q_0, F)$, onde:

- Q é um conjunto finito de estados.
- Σ é um conjunto finito de símbolos, chamado de alfabeto do autômato.
- δ é a função de transição, isto é, $\delta: Q \times \Sigma \rightarrow Q$.
- q_0 é o estado inicial, do autômato antes de qualquer entrada ser processada, onde $q_0 \in Q$.
- F é um conjunto de estados de Q (isto é, $F \subseteq Q$) chamado de estados de aceitação [3].

B. Autômato Finito Determinístico

Um autômato finito determinístico AFD ou máquina de estados finita determinística é aquela que se encontra em um único estado depois de ler uma sequência aleatória de entradas, ou seja, para cada entrada existe somente um estado para o qual o autômato pode se deslocar partindo do estado atual. Isso quer dizer que a função de transição realiza o mapeamento de um par composto por estado e símbolo para um único estado. Deste modo cada palavra da entrada será sempre associada a um único estado final como mostra a Figura 1.

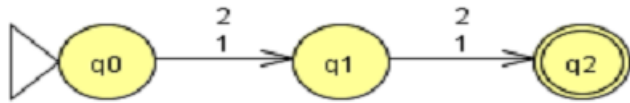


Fig. 1. Exemplo de Autômato Finito Determinístico.

As imagens foram obtidas através da ferramenta JFLAP¹ usada para o estudo de linguagens formais e autômatos finitos não-determinísticos [4].

C. Autômato Finito Não Determinístico

O não determinismo é um conceito útil que tem tido grande impacto na teoria da computação [3]. Nem sempre a função de transição de um autômato finito realiza o mapeamento de um par composto por estado e símbolo para um único estado. Um autômato finito não determinístico AFND é definido como uma máquina de estados finita onde para cada entrada existem diversos estados para os quais o autômato poderá se deslocar partindo do estado atual. Possibilitando desse modo que cada palavra de entrada possa ser associada a diversos estados finais como mostra a Figura 2.

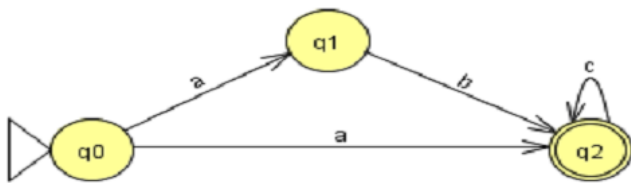


Fig. 2. Exemplo de Autômato Finito Não Determinístico.

D. Equivalência Entre Autômatos Finitos

Autômatos finitos determinísticos ou não determinísticos reconhecem as mesmas classes de linguagens [5]. Um autômato finito não determinístico pode ser transformado em um autômato finito determinístico que aceita a mesma linguagem mas o número de estados do AFD pode ser exponencial com relação ao AFND. Todo o autômato finito não determinístico tem um equivalente autômato finito determinístico.

IV. EXEMPLO DE APLICAÇÃO

Como forma de exemplificar os estudos apresentados até aqui, nas seções seguintes serão descritas as etapas da modelagem de um sistema hipotético de caixa eletrônico. Por simplificação a máquina modelada suprimiu alguns estados existentes em sistemas de dispensação convencionais.

A. Alfabeto de Entrada

Este conjunto de símbolos está associado a cada uma das possíveis transições entre os estados existentes onde:

$$\Sigma = \{cm, ck, si, ci, cs, sq, sa, ex, df, np, sf, rc, fi\}$$

A Tabela I descreve as transições correspondentes a cada um dos símbolos deste conjunto.

TABELA I. ALFABETO

Σ	Transições
cm	Cartão mal inserido
ck	Cartão ok
si	Senha incorreta
ci	Conta incorreta
cs	Conta ok
sq	Saque
sa	Saldo
ex	Extrato
df	Data incorreta
np	Nova operação
sf	Saldo insuficiente
rc	Retira cartão
fi	Finaliza operação

$\Sigma = \{cm, ck, si, ci, cs, sq, sa, ex, df, es, np, sf, rc, fi\}$

B. Conjunto de Dados Possíveis

Este conjunto descreve os estados possíveis que a máquina pode assumir à medida que o usuário realiza transações e é definido como:

$$Q = \{q0, q1, q2, q3, q4, q5, q6\}$$

A tabela II apresenta e descreve os estados possíveis que cada um dos símbolos desse conjunto pode assumir. Cada estado é uma ação hipotética que a máquina poderá assumir. Para fins de elucidação do conteúdo foram adotados estados reduzidos.

¹<http://www.jflap.org/>

TABELA II. ESTADOS

Q	Descrição
q0	Insere cartão
q1	Aguarda senha
q2	Aguarda operação
q3	Exibe saldo
q4	Aguarda data
q5	Aguarda valor
q6	Retira cartão

 $Q = \{q0, q1, q2, q3, q4, q5, q6\}$

C. Estágios Passo a Passo

Os estágios do autômato finito determinístico proposto para solucionar o problema são descritos a seguir:

- 1) O estado **q0** é o estado inicial e a máquina encontra-se aguardando a ação do usuário, nesse estado é verificado se o cartão está *ok* (*ck*) ou não (*cm*).
- 2) No estado **q1** o usuário já inseriu o cartão e a máquina aguarda a entrada correspondente a senha e conta corrente. Se *senha incorreta* (*si*) ou *conta incorreta* (*ci*) o autômato permanece nesse estado.
- 3) O próximo estado é o estado **q2** neste ponto o usuário deverá escolher a operação a ser realizada sendo elas: *saldo* (*sa*), *extrato* (*ex*), *saque* (*sq*), ou encerrar a operação (*fi*). Para cada ação o autômato avançará ao estado correspondente, **q3**, **q4**, **q5** ou **q6** respectivamente.
- 4) O estado **q3** exibe o saldo do cliente, se uma *nova operação* (*np*) quer ser realizada ou autômato retorna ao estado **q2**. Se um *saque* (*sq*) vai ser realizado o autômato avança para o estado **q5** ou ainda *finaliza a operação* (*fi*) indo ao estado **q6**.
- 5) O estado **q4** aguarda a data para informar o extrato ao cliente, a partir desse estado ele pode avançar ao estado **q5** *saque* (*sq*), retornar ao estado **q2**, para uma *nova operação* (*np*), permanecer nesse estado caso a data informada esteja incorreta (*df*) ou avançar até o estado **q6** finalizando (*fi*).
- 6) O estado **q5** aguarda o valor para saque, caso valor esteja *fora dos limites* (*sf*) ele continua nesse estado; retorna ao **q2** para *nova operação* (*np*) ou avança para o estado final **q6**.
- 7) O estado **q6** é o estado final onde o cliente deve *retirar o cartão* (*rc*) para finalizar a operação.

A Figura 3 mostra o diagrama de transições realizadas pelo sistema. Todos os estados e transições apresentados na imagem foram discutidos previamente. Para a modelagem e estudo do autômato foi usada a ferramenta JFLAP. Para fins de compreensão dos conceitos discutidos, forma adotadas simplificações ao processo de funcionamento real de um sistema de caixa eletrônico.

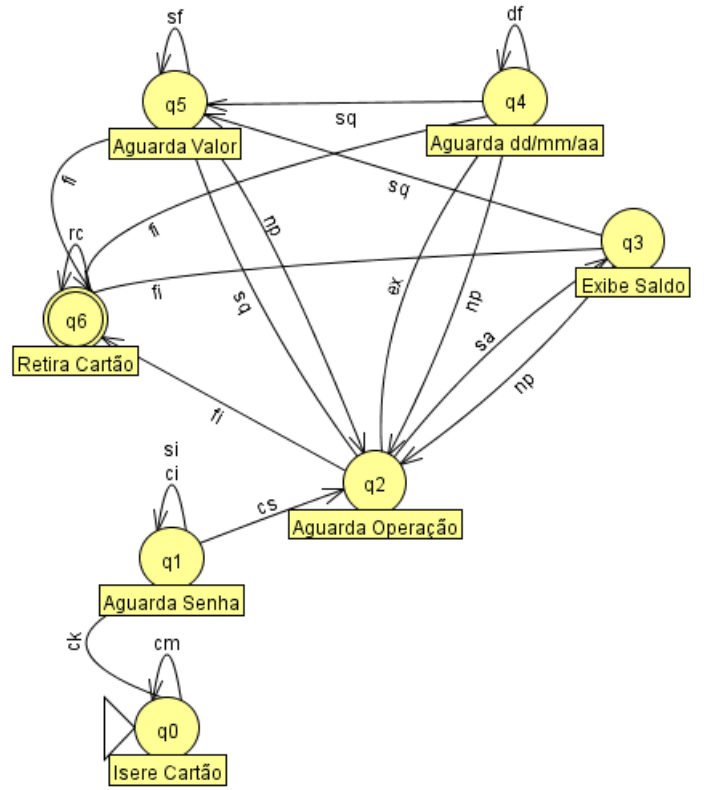


Fig. 3. Diagrama de Transições do Sistema.

V. CONCLUSÃO

Nesse artigo foram descritos os conceitos formais de autômatos finitos determinísticos e não determinísticos e logo a seguir foram abordadas as etapas de implementação de um autômato finito determinístico para solucionar um problema hipotético relacionado ao funcionamento de um sistema de caixa eletrônico.

REFERENCES

- [1] J. Von Neumann, "Design of computers, theory of automata and numerical analysis. collected works, vol. v," 1961.
- [2] K. Culik II, "Variations of the firing squad problem and applications," *Information Processing Letters*, vol. 30, no. 3, pp. 152–157, 1989.
- [3] M. Sipser, "Introduction to the theory of computation," *ACM Sigact News*, vol. 27, no. 1, pp. 27–29, 1996.
- [4] S. H. Rodger and T. W. Finley, *JFLAP: an interactive formal languages and automata package*. Jones & Bartlett Learning, 2006.
- [5] J. E. Hopcroft, R. Motwani, and J. D. Ullman, "Introduction to automata theory, languages, and computation," *Acm Sigact News*, vol. 32, no. 1, pp. 60–65, 2001.