Os Autômatos Finitos

2.1 Introdução

O conceito de autômato finito fornecerá uma primeiro proposta da noção de procedimento efetivo (a escolha final para procedimento efetivo serão as maquinas de Turing)

Os autômatos finitos correspondem a uma ferramenta importante na resolução de vários problemas em computação (e engenharia).

Ex: Busca de string de caráter num texto e analise léxica em compilação

Como funciona um programa executado num computador?

O computador é composto de:

- um processador : conjunto de registradores inclusive um contador de instruções (program counter = PC)

- uma memória onde se encontra o programa (se for um programa embutido a memória pode ser morta = ROM)

- uma memória viva para estocar dados de entrado e de processamento

Como representar de modo abstrato os ciclos de execução?

A execução de uma instrução pode ser vista como a transformação do conteúdo da memória e dos registradores.

O conteúdo da memória e dos registradores é chamado de ESTADO do programa.

Cada ciclo de execução transforma o estado presente num novo estado.

Tal transformação pode ser vista como uma função cujo conjunto origem é o conjunto dos estados e cujo conjunto destino é também o conjunto dos estados.

Tal função é chamada de função de transição.

A partir da definição da função de transição e do estado inicial do programa, podemos saber exatamente a evolução futura do programa (comportamento deterministico)

O conceito de estado de um sistema (em computação ou engenharia) correspondera um conceito bastante genérico.

O estado de um sistema num momento especifico corresponde então a toda informação necessária para prever a evolução futura do sistema.

Por exemplo, em mecânica, sabendo da posição e velocidade de um sólido podemos prever a trajetória futura do objeto.

Em particular, não é necessário saber a seqüência de passos que levaram ao estado presente para prever o futuro (hipótese Markoviano no caso dos processos estocasticos)

Para um programa especifico rodando num computador especifico, o número de estados será então muito grande mas FINITO!

Ex: um computador de 8Mbytes de memória com 16 registradores de 32 bits tem um número de estados igual a :

$$2^{2^{26}+2^9} = 2^{67109376}$$

Um programa executado por um computador pode ser formalmente representado (modelado) então por:

- um conjunto finito de estados (discretos)

- uma função de transição definida sobre o conjunto dos estados discretos

- um estado inicial

A execução de um programa por um computador é formalmente obtida pela aplicação sucessiva da função de transição a partir do estado inicial.

Em vez de supor que os dados de entrada se encontram inicialmente na memória do computador, podemos considerar que tais dados são fornecidos ao programa caráter por caráter.

Neste caso precisa-se acrescentar no modelo do programa um dispositivo de leitura de caráter.

Se o tempo entre duas leitura sucessivo de caráter é maior que o tempo de execução de um ciclo de funcionamento do programa, podemos supor que um ciclo de funcionamento ficará travado até a próxima leitura do próximo caráter a ser lido.

Considerando um ciclo de execução do programa + o dispositivo de leitura de caráter, podemos imaginar uma primeira descrição de um modelo de procedimento efetivo (chamado autômato finito)

2.2 Representação informal do autômatos finitos

Um autômato finito determinístico é composto dos seguintes elementos:

- uma fita de entrada infinita para armazenar o dado de entrada (uma palavra) assim como um dispositivo de leitura

- um conjunto finito de estados com em particular:

* um estado inicial

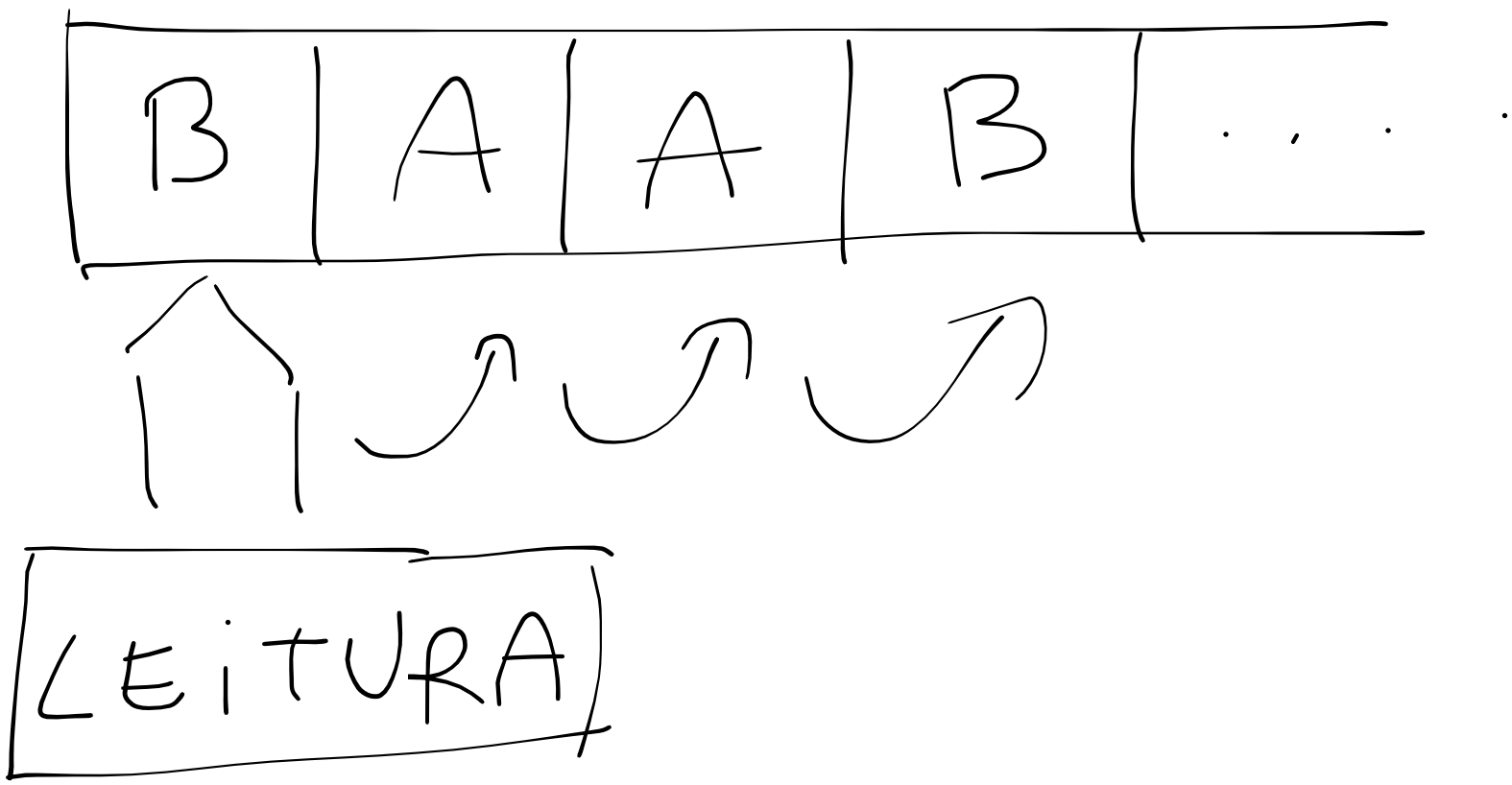
* estados de aceitação que representam os casos onde as palavras de entrada são aceitas (resposta sim do problema binário correspondente)

- uma função de transição que a partir do estado presente e do símbolo lido vai produzir um estado seguinte

Inicialmente o programa se encontra no estado inicial e a cabeça de leitura se encontra no primeiro símbolo da palavra de entrada.

A cada etapa de execução, o autômato lê um símbolo da fita, determina o próximo estado (baseando se na função de transição) e movimenta a cabeça de leitura de uma posição para a direita.

O autômato pára quando toda palavra de entrada foi lida e a palavra é aceita pelo autômato se o estado final é um dos estados de aceitação.


B
A
A
B
. . .
LEITURA

2.3 Definição Formal de um autômato finito determinístico

Um autômato finito determinístico é definido pela tupla de 5 valores $M = (Q,\Sigma,\delta,s,F)$ onde:

- Q é um conjunto finito de estados

- Σ é um alfabeto

- $\delta: Q\text{x}\Sigma \rightarrow Q$ é a função de transição (x produto cartesiano)

- $s \in Q$ é o estado inicial

- $F \subseteq Q$ é o conjunto dos estados de aceitação

Um autômato finito determinístico é uma maquina que resolva um problema ou que reconhece (aceita) uma certa linguagem.

Como qualquer maquina, existe a noção de configuração do autômato que representa toda informação necessária para determinar a evolução futura da maquina.

A configuração é representada pelo estado do autômato e pela parte da palavra de entrada que não foi lida ainda.

É um elemento dado por $(Q,w) \in Q \times \Sigma^*$

Cada etapa da execução de um autômato modifica então a configuração do autômato.

Def: A configuração (q',w') é derivável numa única etapa a partir da configuração (q,w) pela maquina M se:

- $w = \sigma w'$ (o primeiro caráter de w é σ)

- $q' = \delta(q,\sigma)$ (q' é o resultado da aplicação da função de transição)

NOTAÇÃO:

$$(q, w) \vdash_M (q', w')$$

Def: A configuração (q',w') é derivável a partir da configuração (q,w) pela maquina M se existe $k \geq 0$ e as configurações (qi,wi) com $0 \leq i \leq k$ tais que:

- (q,w) = (qo,wo)

- (q',w') = (qk,wk)

- para todo $o \leq i \leq k$, (qi,wi) |- (qi+1,wi+1)

NOTAÇÃO:

$$(q, w) \vdash_M^* (q', w')$$

Def: a execução de um autômato sobre uma w é a seqüência de configurações:

(S,w) |- (q1,w1) |- (q2,w2) |- ... |- (qn,ε)

onde s é o estado inicial e ε a palavra vazia

Def: Uma palavra w é aceita pelo autômato M se

$$(s, w) \vdash_M^* (q, \varepsilon)$$

com $q \in F$

Def: A linguagem aceita pelo autômato M é o conjunto das palavras L(M) tais que:

$$L(M) =$$

$$\{w \in \Sigma^* \mid (s, w) \vdash_M^* (q, \varepsilon)$$

$$\text{com } q \in F\}$$

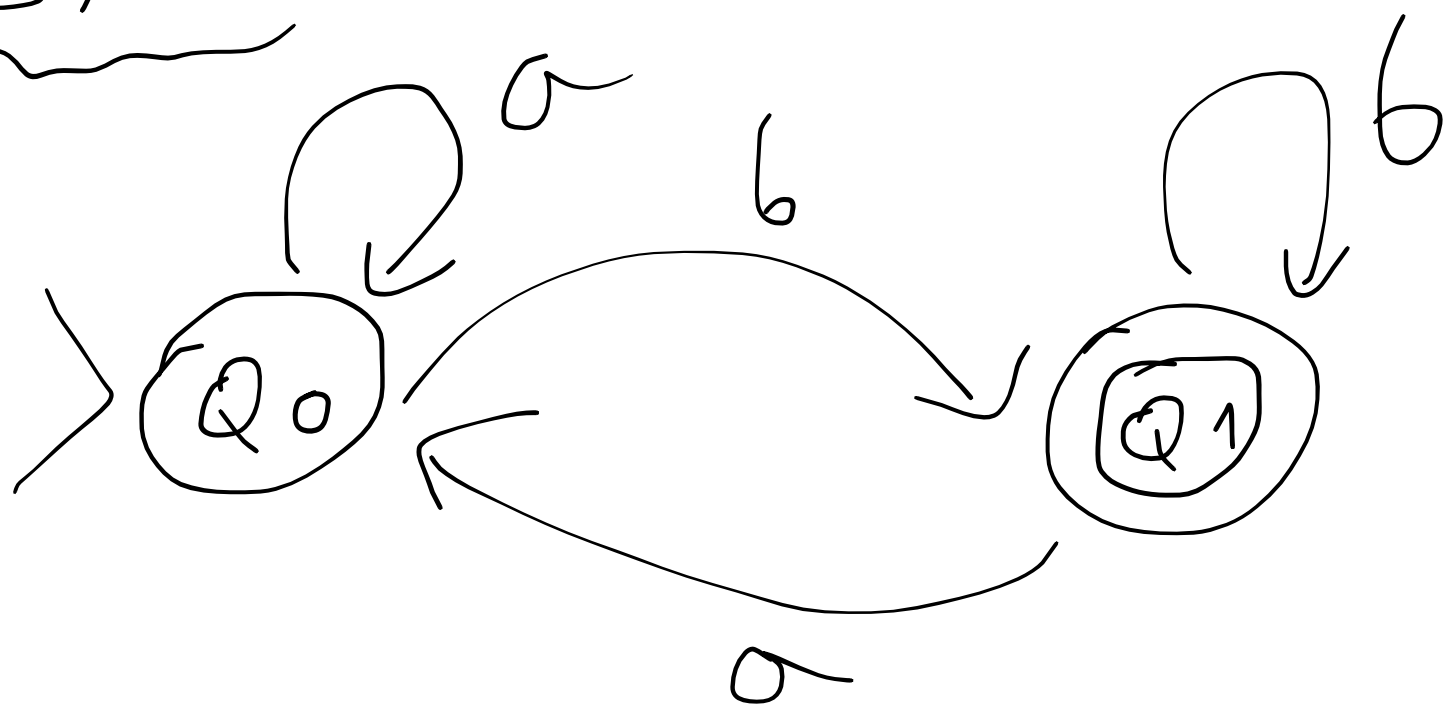
Na pratica, um autômato é representado por um grafo (grafo das transições/estados) onde os estados são representados por vértices e as transições por arcos. O estado inicial é apontado por uma seta e os estados de aceitação são representados por um circulo duplo.

Ex1: Representar o autômato que aceita as palavras que terminam pela letra b com $\Sigma=\{a,b\}$

Ex2: Representar o autômato que aceita a linguagem dada por {w | w não tem 2 a's consecutivos}

Dar exemplos de seqüências de configurações para os autômatos correspondentes.

Ex 1:

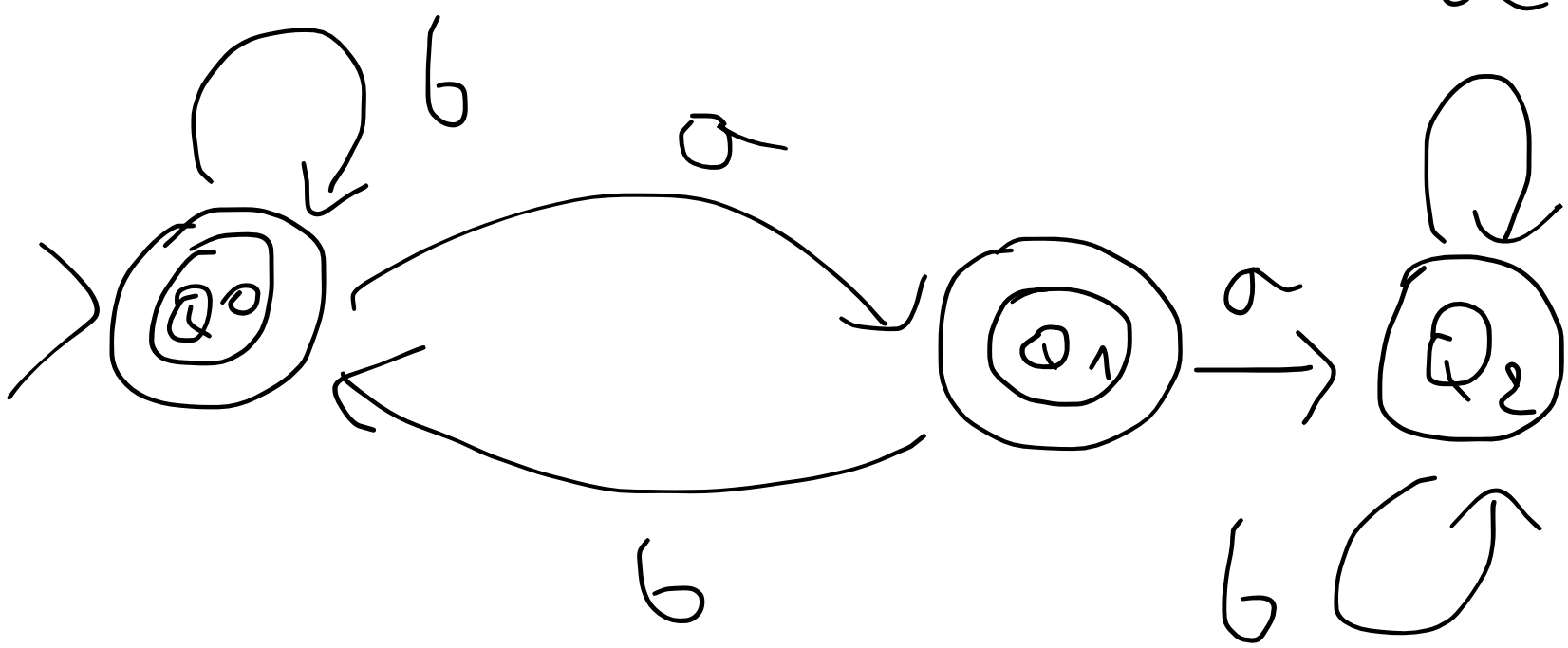


$$Q = \{q_0, q_1\}$$

$$\Sigma = \{a, b\} \qquad S = q_0$$

$$F = \{q_1\}$$

Ex 2



2.4 Limitações dos autômatos finitos

Teorema: uma linguagem é regular se é somente se ela é aceita por um autômato finito.

Isso implica uma noção de equivalência entre as linguagens regulares (problemas) e os autômatos finitos (procedimento efetivo)

Conclusão: os autômatos finitos não constituam um modelo satisfatório para modelar a noção de procedimento efetivo (programa/algoritmo) já que existem linguagens que não são regulares e pelos quais um procedimento efetivo existe !

Ex: a linguagem An Bn não é regular mas pode ser reconhecida por um procedimento efetivo.

Existe um autômato finito que reconhece a linguagem:

$$L_k = \{ a^m b^m \mid m \leq k \}$$

Ex: encontrar tal autômato para n=2

Mas não existe um autômato finito que reconhece tal linguagem para n qualquer.

A gente precisaria da noção de autômato infinito !

Mas será que na pratica existe um algoritmo que reconhece a linguagem An Bn ?

Obviamente sim !

Produzir tal algoritmo em pseudo linguagem.

Conclusão: para entender a noção de procedimento efetivo e saber que tipo de problema pode ser resolvido num tempo de cálculo razoável , é necessário estudar outros tipos de autômatos de memória infinita.

Tais autômatos serão chamados de Máquinas de Turing !