Aulas 1 e 2

Aula 1 - Teoria da Computação (Apostila Capítulo 0)

Quais temas serão abordados no curso?
O que é o curso de Teoria da Computação?

Questão central:

Quais são as capacidades e as limitações fundamentais dos computadores?

veja: Lance Fortnow, CACM, sept 2009

Três pontos de vista para responder:
autômatos, computabilidade e complexidade
correspondem a três áreas diferentes

O periódico Theoretical Computer Science TCS historicamente tem duas seções e classifica
Teoria A: Algorithms, Automata, Complexity and Games
Teoria B: Logic, Semantics and Theory of Programming

No PESC nos organizamos em Linhas de Pesquisa, a Teoria A na linha de AC, Algoritmos, Complexidade e Combinatória; e a Teoria B na linha de IA, Inteligência Artificial, Métodos Formais e Programação em Lógica

mais recentemente a TCS tem a seção
Teoria C: Natural Computing,
computing inspired by nature, Neural
Networks, Quantum Computing.

Voltemos aos três pontos de vista.

Teoria da Complexidade: classifica um
problema computacional em fácil ou difícil,
através da definição de algoritmo,
como solução eficiente ou ineficiente.

Veja: Resolver ou Verificar? Ciência Hoje,
Nov 2011, página web Celina

O que faz alguns problemas computacio-
nalmente difíceis e outros fáceis?
curso no terceiro trimestre.

Teoria da Computabilidade: classifica em
solúvel e não solúvel.

Apresentaremos um problema algoritmicamente
não solúvel

O problema da parada da Máquina de Turing.

Palestra do Luis Menasché, prof do IM,
autor da apostila, encerra o nosso curso.

Teoria dos Autômatos: Definições e Propriedades de modelos matemáticos de Computação.

As Teorias da Computabilidade e da Complexidade requerem uma definição precisa de um computador. Começamos dos autômatos como modelo mais simples, a máquina de estados finita, autômato finito, finite state machine.

O autômato finito modela o computador com memória pequena, mesmo assim somos capazes de reconhecer linguagens com um número infinito de palavras.

Definições: alfabeto é um conjunto finito de símbolos $\Sigma = \{a_1, a_2, \dots, a_n\}$

palavra no alfabeto Σ é uma sequência finita $w = r_1 r_2 \cdots r_\ell$, $\ell \in \mathbb{N}$, $r_i \in \Sigma$

palavra vazia ε é uma cadeia vazia, que não contém nenhum símbolo

$|w|$ é o comprimento, o número de símbolos não necessariamente distintos que ocorrem em w.

$|\varepsilon| = 0$ única palavra de comprimento zero.

Σ^* é o conjunto de todas as palavras no alfabeto Σ. Observe que $\varepsilon \in \Sigma^*$

Uma <u>linguagem formal</u> L no alfabeto Σ é um subconjunto de Σ^*, ie $L \subseteq \Sigma^*$.

Para estudar linguagens formais, dois tipos de ferramentas:

geradores de linguagens: gramáticas, expressões regulares

reconhecedores de linguagens: autômatos

Operações sobre palavras:

concatenação $w_1 \cdot w_2$ $\quad |w_1 \cdot w_2| = |w_1| + |w_2|$

operação associativa, mas não comutativa

palavra vazia satisfaz $w \cdot \varepsilon = \varepsilon \cdot w = w$

Como linguagens são conjuntos, várias operações vem da Teoria dos Conjuntos: união, interseção, complemento, diferença. A notação também:

$L_1 \setminus L_2 = L_1 \cap \overline{L_2}$ $\quad |L_1 \cdot L_2| \leq |L_1| \times |L_2|$

$\overline{L} = \Sigma^* \setminus L$; $\{\varepsilon\} \cdot L = L \cdot \{\varepsilon\} = L$; $\phi \cdot L = L \cdot \phi = \phi$

concatenação $L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1 \text{ e } w_2 \in L_2$

operação estrela: $L^* = \bigcup_{i \in \mathbb{N}} L^i$ $\quad \mathbb{N} = \{0, 1, 2, \dots\}$

potências: $L^0 = \{\varepsilon\}$; $L^i = L^{i-1} \cdot L = L \cdot L^{i-1}$

Aula 2 - Autômatos (Apostila Capítulo 1)

Um autômato finito determinístico (AFD) é uma 5-upla $M = (Q, \Sigma, \delta, q_0, F)$, onde Q é o conjunto finito de estados, Σ é o alfabeto finito, $\delta : Q \times \Sigma \rightarrow Q$ é a função de transição ou função próximo estado, $q_0 \in Q$ é o estado inicial, $F \subseteq Q$ é o conjunto de estados de aceitação, ou estados finais.

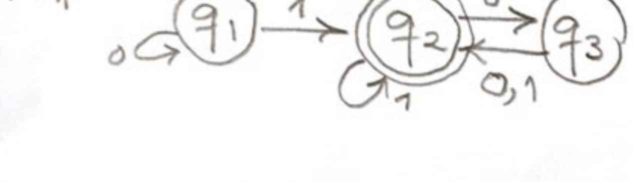
Exemplo: $M_1 = (Q, \Sigma, \delta, q_0, F)$, onde
$Q = \{q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$ alfabeto binário,
δ função descrita pela tabela
q_1 é o estado inicial
$F = \{q_2\}$

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

$\delta(q_1, 0) = q_1$

$\delta(q_2, 0) = q_3$

Representação através do Diagrama de Estados



convenção:
$\rightarrow$ ○ estado inicial
◎ estados finais

Dizemos que: M aceita a cadeia $w = w_1 w_2 \dots w_\ell$ se existe sequência de estados $r_0, r_1, \dots, r_\ell \in Q$ tal que: $r_0 = q_0$; $\delta(r_i, w_{i+1}) = r_{i+1}$, $0 \leq i < \ell$; $r_\ell \in F$

M reconhece linguagem A se $A = \{w \mid M \text{ aceita } w\}$

A é a linguagem do autômato M, $L(M) = A$

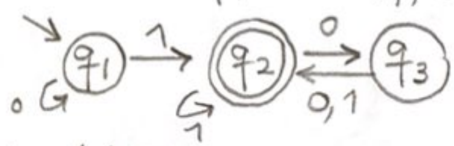
Pela definição de AFD, qualquer cadeia $w \in \Sigma^*$, ao ser recebida como entrada a partir de q_0, percorrerá $|w|$ transições usando δ e parará em $q \in Q$

$q \in F \iff M$ aceita w.

A é uma <u>linguagem regular</u>, se algum autômato finito M a reconhece, isto é, se $A = L(M)$.

Definimos: configuração (q, w), $q \in Q$, $w \in \Sigma^*$;
configuração seguinte $(q,w) \vdash (q', w')$, se $w = \sigma w'$, onde $\sigma \in \Sigma$ e $\delta(q, \sigma) = q'$;
computação em M, $C_0 \vdash \cdots \vdash C_n$, $C_0 \vdash^* C_n$, onde $|w| = n$;
$w \in \Sigma^*$ é aceita por M se $(q_0, w) \vdash^* (f, \varepsilon)$, onde $f \in F$.

Volte ao exemplo M_1

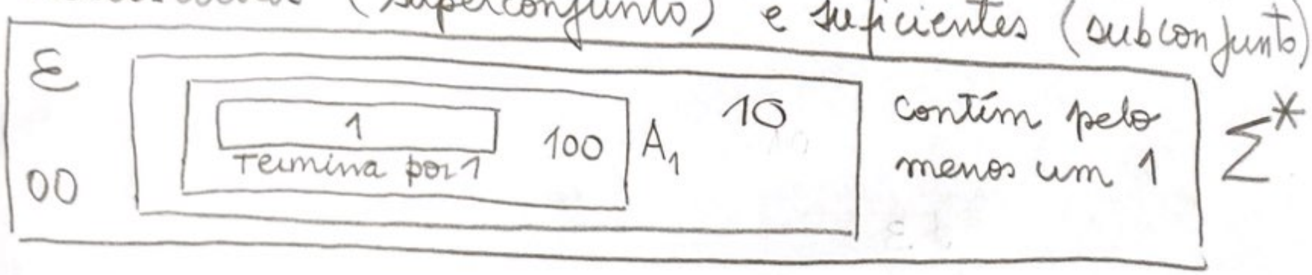


M_1 recebe como entrada $w = 1101$:

$q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{1} q_2$ $\quad$ M_1 aceita w

$(q_1, 1101) \vdash (q_2, 101) \vdash (q_2, 01) \vdash (q_3, 1) \vdash (q_2, \varepsilon)$ computação em M_1

Determinamos $A_1 = L(M_1) \subseteq \Sigma^*$ através de condições necessárias (superconjunto) e suficientes (subconjunto):



Distinguimos condição

necessária: $w \in A_1 \Rightarrow w$ contém pelo menos um 1.
para sair do estado inicial q_1, precisa consumir um 1

suficiente: $w \in A_1 \Leftarrow w$ termina por 1
todo 1 leva ao estado final q_2.

$L(M_1) = A_1 = \{ w \mid w$ contém pelo menos um 1, e um número par de zeros segue o último 1 $\}$ 2.3

Teorema: A_1 é a linguagem do autômato M_1.

prova: Provaremos por indução em $|w|$ que o estado atingido é

$\boxed{q_1 \mid q_2 \mid q_3}\ \Sigma^*$

$q_1 \iff w = \varepsilon$ ou w só contém zeros;

$q_2 \iff w$ contém 1 e o último 1 é seguido de um número par de zeros;

$q_3 \iff w$ contém 1 e o último 1 é seguido de um número ímpar de zeros.

base: $|w| = 0$, $w = \varepsilon$, não saímos do estado inicial q_1. Sempre vale $w = \varepsilon \iff w$ atinge estado inicial q_1.

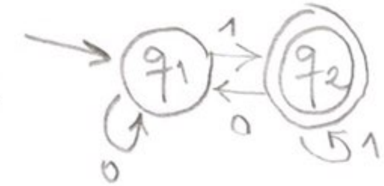
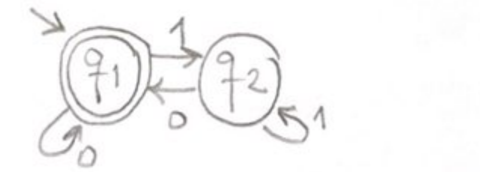
passo de indução. suponha $|w| = k > 0$, portanto $w = x0$ ou $x1$, o argumento depende do último símbolo, e a hipótese de indução vale para $x \in \Sigma^*$.

suponha $w = x0$. Precisamos provar que w satisfaz as três equivalências, farei a primeira. Se w atinge q_1, viemos de q_1 consumindo símbolo 0, e por indução x é a cadeia vazia ou só contém zeros. Logo, w só contém zeros. Por outro lado, como w não é a cadeia vazia, a hipótese é w só contém zeros. Logo, x só contém zeros ou é a cadeia vazia, e por indução x atinge q_1, logo consumindo 0, w atinge q_1.

suponha $w = x1$. Faça como exercício!

QED ▨

Compare M_1 com dois autômatos

M_2  M_3 

<u>Teorema</u> $L(M_2) = \{w \mid w \text{ termina com } 1\}$

<u>prova</u>: $\supseteq$ Toda transição 1 leva ao estado final. Portanto, se w termina com 1, então M_2 aceita w.
$\subseteq$ Apenas a transição 1 leva ao estado final. Portanto, se w é aceita por M_2, então w termina com 1.

QED ■

prova alternativa: por indução em $|w|$, o estado atingido é $q_1 \Leftrightarrow w$ não termina com 1; $q_2 \Leftrightarrow w$ termina com 1.
base: $|w| = 0$, $w = \varepsilon$. Não saímos de q_1.
passo de indução: $|w| = k > 0$. Dois casos $w = x0$ ou $w = x1$.
suponha $w = x0$. Então w não atinge q_2, sabemos que w atinge q_1: vindo de q_1 com x terminando em 0, ou vindo de q_2 com x terminando em 1.
suponha $w = x1$. Então w não atinge q_1, sabemos que w atinge q_2: vindo de q_1 com x terminando em 0, ou vindo de q_2 com x terminando em 1. ■

<u>Teorema</u> $L(M_3) = \{w \mid w \text{ termina com } 0\} \cup \{\varepsilon\}$

<u>prova</u>: Veja a prova para $L(M_2)$ e a mudança apenas do estado final ■

$L(M_2)$	$L(M_3)$

Σ^*

Observe que 1. $\varepsilon \in L \Leftrightarrow q_0 \in F$
2. $L(M_3) = \overline{L(M_2)}$
3. AFD M de L com estados finais F fornece AFD M' de $\overline{L}$ com estados finais $Q \setminus F$

A classe das linguagens regulares é fechada por complemento.

Operações regulares: sejam A e B linguagens, definimos as operações regulares união $A \cup B = \{x \mid x \in A$ ou $x \in B\}$; concatenação $AB = \{xy \mid x \in A$ e $y \in B\}$; estrela $A^* = \{x_1 x_2 \ldots x_k \mid k \geq 0$ e $x_i \in A\} = \bigcup_{i \in \mathbb{N}} A^i$

Observe que $\varepsilon \in A^*$ sempre, e que se $\varepsilon \notin A$, então $A \subsetneq A^*$.

<u>Teorema</u>: A classe das linguagens regulares é fechada sob operação de união.

<u>prova</u>: se M_i reconhece A_i, $i = 1, 2$, então defina M que reconhece $A_1 \cup A_2$.
assuma A_i sob o mesmo Σ, $i = 1, 2$.
$M = (Q, \Sigma, \delta, q_0, F)$, onde
$Q = Q_1 \times Q_2$, Σ é o mesmo, $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
$q_0 = (q_1, q_2)$, $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$. ▨

<u>Teorema</u>: Fecho por interseção

<u>prova</u>: acima, mude apenas $F = F_1 \times F_2$. ▨