MAC 414
Autômatos, Computabilidade e Complexidade
aula 10 — 19/10/2020

Como definir algoritmo?

Como definir algoritmo?

O termo algoritmo já era usado no século XIX, mas não muito popular (entre matemáticos).

Como definir algoritmo?

O termo algoritmo já era usado no século XIX, mas não muito popular (entre matemáticos).

Só ganhou popularidade com o surgimento da Ciência da Computação.

Como definir algoritmo?

O termo algoritmo já era usado no século XIX, mas não muito popular (entre matemáticos).

Só ganhou popularidade com o surgimento da Ciência da Computação.

Em 1900 Hilbert propôs uma série de problemas importantes para o novo século. Alguns foram logo resolvidos, outros ainda estão em aberto. Outros foram resolvidos durante o século.

Como definir algoritmo?

O termo algoritmo já era usado no século XIX, mas não muito popular (entre matemáticos).

Só ganhou popularidade com o surgimento da Ciência da Computação.

Em 1900 Hilbert propôs uma série de problemas importantes para o novo século. Alguns foram logo resolvidos, outros ainda estão em aberto. Outros foram resolvidos durante o século.

Entscheidungsproblem (1928): algoritmo para decidir se um sentença pode ser provada a partir de axiomas dados.

Hilbert

Hilbert

Enunciado do problema 10:

> *Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.*

Hilbert

Enunciado do problema 10:

> *Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.*

A palavra *algoritmo* não aparece.

Hilbert

Enunciado do problema 10:

> *Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.*

A palavra *algoritmo* não aparece.

Outros que não usaram: Godel, Post, Turing, Church.

Mas tinham o conceito.

Hilbert

Enunciado do problema 10:

> *Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.*

A palavra *algoritmo* não aparece.

Outros que não usaram: Godel, Post, Turing, Church.

Mas tinham o conceito.

Atualmente, "algoritmo" caiu na boca do povo e significa algo misterioso que acontece no Google ou no Facebook.

Hilbert

Enunciado do problema 10:

> *Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined in a finite number of operations whether the equation is solvable in rational integers.*

A palavra *algoritmo* não aparece.

Outros que não usaram: Godel, Post, Turing, Church.

> Mas tinham o conceito.

Atualmente, "algoritmo" caiu na boca do povo e

> significa algo misterioso que acontece no Google ou no Facebook.

Mas a gente sabe o que é. ☺

Turing

Turing

Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem", Proceedings of the London Mathematical Society, Series 2, 42 (1936-7), pp 230-265.

Turing

Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem", Proceedings of the London Mathematical Society, Series 2, 42 (1936-7), pp 230-265.

Quase simultaneamente, Godel e Church provaram resultados equivalentes com formalismos completamente diferentes.

Turing

Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem", Proceedings of the London Mathematical Society, Series 2, 42 (1936-7), pp 230-265.

Quase simultaneamente, Godel e Church provaram resultados equivalentes com formalismos completamente diferentes.

As *Máquinas de Turing* podem ser definidas das mais variadas formas. O essencial:

- Controle finito, conjunto finito de instruções.

Turing

Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem", Proceedings of the London Mathematical Society, Series 2, 42 (1936-7), pp 230-265.

Quase simultaneamente, Godel e Church provaram resultados equivalentes com formalismos completamente diferentes.

As *Máquinas de Turing* podem ser definidas das mais variadas formas. O essencial:

- Controle finito, conjunto finito de instruções.
- Simples o suficiente para se poder provar coisas a respeito.

Máquina de Turing

Máquina de Turing

Baseada no **L&P**. **Sipser** e **Rich** definem diferente.

Máquina de Turing

Baseada no **L&P**. **Sipser** e **Rich** definem diferente.

Uma Máquina de Turing é uma quíntupla $(K, \Sigma, \delta, s, H)$, onde

- K é o conjunto finito de estados.
- Σ é o alfabeto. Símbolo especial: $\sqcup$ (espaço) ; símbolos $\rightarrow$ e $\leftarrow$ e $\triangleright$ (início) não estão em Σ.
- $s \in K$ é o estado inicial.
- $H \subseteq K$ é o conjunto de estados de parada.
- A função de transição
  $\delta : K \setminus H \times \Sigma \cup \{\triangleright\} \rightarrow K \times \Sigma \cup \{\leftarrow, \rightarrow\}$
  satisfazendo $\delta(p, \triangleright) = (q, \rightarrow)$.

Significado

Uma MT tem uma fita de entrada e uma cabeça de leitura que aponta uma posição por vez. A transição depende do estado atual e do que é lido na fita; a primeira componente indica o novo estado:

- Se $\delta(p, a) = (q, b)$, escreve b nessa posição.

Significado

Uma MT tem uma fita de entrada e uma cabeça de leitura que aponta uma posição por vez. A transição depende do estado atual e do que é lido na fita; a primeira componente indica o novo estado:

- Se $\delta(p, a) = (q, b)$, escreve b nessa posição.
- Se $\delta(p, a) = (q, \leftarrow)$, anda para a esquerda uma posição.

Significado

Uma MT tem uma fita de entrada e uma cabeça de leitura que aponta uma posição por vez. A transição depende do estado atual e do que é lido na fita; a primeira componente indica o novo estado:

- Se $\delta(p, a) = (q, b)$, escreve b nessa posição.
- Se $\delta(p, a) = (q, \leftarrow)$, anda para a esquerda uma posição.
- Se $\delta(p, a) = (q, \rightarrow)$, anda para a direita uma posição.

Significado

Uma MT tem uma fita de entrada e uma cabeça de leitura que aponta uma posição por vez. A transição depende do estado atual e do que é lido na fita; a primeira componente indica o novo estado:

- Se $\delta(p, a) = (q, b)$, escreve b nessa posição.
- Se $\delta(p, a) = (q, \leftarrow)$, anda para a esquerda uma posição.
- Se $\delta(p, a) = (q, \rightarrow)$, anda para a direita uma posição.

  O início $\triangleright$ empurra a leitora para a direita.

Exemplo

$$K = \{s, q, h\}, \Sigma = \{a, \sqcup\}, H = \{h\}$$

δ	a	$\sqcup$	$\triangleright$
s	$(q, \sqcup)$	$(h, \sqcup)$	$(s, \rightarrow)$
q	(s, a)	$(s, \rightarrow)$	$(q, \rightarrow)$

Exemplo

$$K = \{s, q, h\}, \Sigma = \{a, \sqcup\}, H = \{h\}$$

δ	a	$\sqcup$	$\triangleright$
s	$(q, \sqcup)$	$(h, \sqcup)$	$(s, \rightarrow)$
q	(s, a)	$(s, \rightarrow)$	$(q, \rightarrow)$

e se $\delta(q, \sqcup) = (q, \rightarrow)$?

Formalização

Formalização

Uma configuração de M é um elemento de

$$K \times \triangleright\Sigma^* \times (\Sigma^* \setminus \Sigma^* \sqcup).$$

Formalização

Uma configuração de M é um elemento de

$$K \times \triangleright\Sigma^* \times (\Sigma^* \setminus \Sigma^* \sqcup).$$

(p, u, v) significa estado p, fita uv, cabeça de leitura na primeira letra de v.

Formalização

Uma configuração de M é um elemento de

$$K \times \triangleright\Sigma^* \times (\Sigma^* \setminus \Sigma^* \sqcup).$$

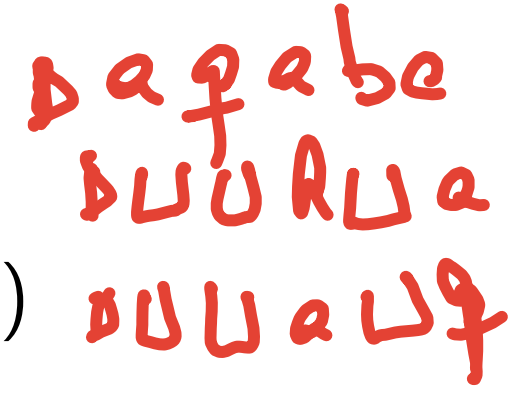
(p, u, v) significa estado p, fita uv, cabeça de leitura na primeira letra de v.

Notação:

$(q, \triangleright a, aba) \longrightarrow (q, \triangleright a\underline{a}ba)$

$(h, \triangleright \sqcup\sqcup, \sqcup a) \longrightarrow (h, \triangleright \sqcup\sqcup\underline{\sqcup}a)$

$(q, \triangleright \sqcup\sqcup a\sqcup, \lambda) \longrightarrow (q, \triangleright \sqcup\sqcup a\sqcup\underline{\sqcup})$



Computação

Computação

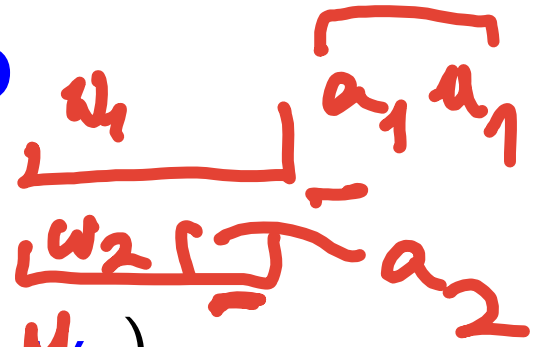
Computação em um passo

$$(p, w_1 \underline{a}_1 w_1) \vdash_M (q, w_2 \underline{a}_2 w_2)$$

se existe $b \in \Sigma \cup \{\rightarrow, \leftarrow\}$ tal que $\delta(p, a_1) = (q, b)$ e

1. $b \in \Sigma, w_2 = w_1, u_2 = u_1$ e $a_2 = b$, ou

Computação



Computação em um passo

$$(p, w_1 \underline{a_1} u_1) \vdash_M (q, w_2 \underline{a_2} u_2)$$

se existe $b \in \Sigma \cup \{\rightarrow, \leftarrow\}$ tal que $\delta(p, a_1) = (q, b)$ e

1. $b \in \Sigma, w_2 = w_1, u_2 = u_1$ e $a_2 = b$, ou
2. $b = \leftarrow, w_1 = w_2 a_2$ e
   a) $u_2 = a_1 u_1$, se $a_1 \neq \sqcup$ ou $u_1 \neq \lambda$, ou
   b) $u_2 = \lambda$, se $a_1 = \sqcup$ e $u_1 = \lambda$, ou

Computação

Computação em um passo

$$(p, w_1 \underline{a_1} u_1) \vdash_M (q, w_2 \underline{a_2} u_2)$$

se existe $b \in \Sigma \cup \{\rightarrow, \leftarrow\}$ tal que $\delta(p, a_1) = (q, b)$ e

1. $b \in \Sigma, w_2 = w_1, u_2 = u_1$ e $a_2 = b$, ou
2. $b = \leftarrow, w_1 = w_2 a_2$ e
   a) $u_2 = a_1 u_1$, se $a_1 \neq \sqcup$ ou $u_1 \neq \lambda$, ou
   b) $u_2 = \lambda$, se $a_1 = \sqcup$ e $u_1 = \lambda$, ou
3. $b = \rightarrow, w_2 = w_1 a_1$ e
   a) $u_1 = a_2 u_2$, ou
   b) $u_1 = u_2 = \lambda$ e $a_2 = \sqcup$.

Computação

Computação em um passo

$$(p, w_1 \underline{a_1} w_1) \vdash_M (q, w_2 \underline{a_2} w_2)$$

se existe $b \in \Sigma \cup \{\rightarrow, \leftarrow\}$ tal que $\delta(p, a_1) = (q, b)$ e

1. $b \in \Sigma, w_2 = w_1, u_2 = u_1$ e $a_2 = b$, ou
2. $b = \leftarrow, w_1 = w_2 a_2$ e
   a) $u_2 = a_1 u_1$, se $a_1 \neq \sqcup$ ou $u_1 \neq \lambda$, ou
   b) $u_2 = \lambda$, se $a_1 = \sqcup$ e $u_1 = \lambda$, ou
3. $b = \rightarrow, w_2 = w_1 a_1$ e
   a) $u_1 = a_2 u_2$, ou
   b) $u_1 = u_2 = \lambda$ e $a_2 = \sqcup$.

Computação

Computação em um passo

$$(p, w_1 \underline{a_1} w_1) \vdash_M (q, w_2 \underline{a_2} w_2)$$

se existe $b \in \Sigma \cup \{\rightarrow, \leftarrow\}$ tal que $\delta(p, a_1) = (q, b)$ e

1. $b \in \Sigma, w_2 = w_1, u_2 = u_1$ e $a_2 = b$, ou
2. $b = \leftarrow, w_1 = w_2 a_2$ e
   a) $u_2 = a_1 u_1$, se $a_1 \neq \sqcup$ ou $u_1 \neq \lambda$, ou
   b) $u_2 = \lambda$, se $a_1 = \sqcup$ e $u_1 = \lambda$, ou
3. $b = \rightarrow, w_2 = w_1 a_1$ e
   a) $u_1 = a_2 u_2$, ou
   b) $u_1 = u_2 = \lambda$ e $a_2 = \sqcup$.

$\vdash_M^*$ é o fecho reflexivo transitivo de $\vdash_M$.

Clarificando

Suponha $w, u \in \Sigma^*$, u não terminado em $\sqcup$, $a, b \in \Sigma$.

1. $\delta(p, a) = (q, b)$
   $(p, w\underline{a}u) \vdash_M (q, w\underline{b}u)$.

Clarificando

Suponha $w, u \in \Sigma^*$, u não terminado em $\sqcup$, $a, b \in \Sigma$.

1. $\delta(p, a) = (q, b)$
   $(p, w\underline{a}u) \vdash_M (q, w\underline{b}u)$.
2. $\delta(p, a) = (q, \leftarrow)$
   a) $(p, wb\underline{a}u) \vdash_M (q, w\underline{b}au)$,
   b) $(p, wb\underline{\sqcup}) \vdash_M (q, w\underline{b})$.

Clarificando

Suponha $w, u \in \Sigma^*$, u não terminado em $\sqcup$, $a, b \in \Sigma$.

1. $\delta(p, a) = (q, b)$
   $(p, w\underline{a}u) \vdash_M (q, w\underline{b}u)$.
2. $\delta(p, a) = (q, \leftarrow)$
   a) $(p, wb\underline{a}u) \vdash_M (q, w\underline{b}au)$,
   b) $(p, wb\underline{\sqcup}) \vdash_M (q, w\underline{b})$.
3. $\delta(p, a) = (q, \rightarrow)$
   a) $(p, w\underline{a}bu) \vdash_M (q, wa\underline{b}u)$,
   b) $(p, w\underline{a}) \vdash_M (q, wa\underline{\sqcup})$.

Exemplo

$K = \{s, q, h\}, \Sigma = \{a, \sqcup\}, H = \{h\}$

δ	a	$\sqcup$	$\triangleright$
s	$(q, \sqcup)$	$(h, \sqcup)$	$(s, \rightarrow)$
q	(s, a)	$(s, \rightarrow)$	$(q, \rightarrow)$

$(q, \triangleright\underline{\sqcup}aaa) \vdash_M$ $(s, \triangleright\sqcup\underline{a}aa) \vdash (q, \triangleright\sqcup\sqcup\underline{a}a)$

$\vdash (s, \triangleright\sqcup\sqcup\underline{a}a) \vdash (q, \triangleright\sqcup\sqcup\sqcup\underline{a})$

$\vdash (s, \triangleright\sqcup\sqcup\sqcup\underline{a}) \vdash (q, \triangleright\sqcup\sqcup\sqcup\underline{\sqcup})$

$\vdash (s, \triangleright\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}) \vdash (h, \triangleright\sqcup\sqcup\sqcup\sqcup\underline{\sqcup})$

Dados para uma MT

Dados para uma MT

Alfabeto de entrada $\Sigma_0 \subseteq \Sigma \setminus \{\sqcup\}$

Dados para uma MT

Alfabeto de entrada $\Sigma_0 \subseteq \Sigma \setminus \{\sqcup\}$

Entrada para M é $w \in \Sigma_0^*$: (M, w) se lê
"M com entrada w".

Dados para uma MT

Alfabeto de entrada $\Sigma_0 \subseteq \Sigma \setminus \{\sqcup\}$

Entrada para M é $w \in \Sigma_0^*$: (M, w) se lê
"M com entrada w".

Convenção: (M, w) significa configuração inicial

$$(s, \triangleright \underline{\sqcup} w).$$

Decidir linguagens

Decidir linguagens

Decisor: $H = \{a, r\}$

Decidir linguagens

Decisor: $H = \{a, r\}$

M aceita w se (M, w) para em a.

Decidir linguagens

Decisor: $H = \{a, r\}$

M aceita w se (M, w) para em a.

M rejeita w se (M, w) para em r.

Decidir linguagens

Decisor: $H = \{a, r\}$

M aceita w se (M, w) para em a.

M rejeita w se (M, w) para em r.

M decide $L \subseteq \Sigma_0^*$ sse
Se $w \in L$, M aceita w, se $w \notin L$, M rejeita w.

Decidir linguagens

Decisor: $H = \{a, r\}$

M aceita w se (M, w) para em a.

M rejeita w se (M, w) para em r.

M decide $L \subseteq \Sigma_0^*$ sse
Se $w \in L$, M aceita w, se $w \notin L$, M rejeita w.

L é recursiva se alguma máquina de Turing decide L.

Decidir linguagens

Decisor: $H = \{a, r\}$

M aceita w se (M, w) para em a.

M rejeita w se (M, w) para em r.

M decide $L \subseteq \Sigma_0^*$ sse
Se $w \in L$, M aceita w, se $w \notin L$, M rejeita w.

L é recursiva se alguma máquina de Turing decide L.

Prop: Se L é recursiva, seu complemento também é.

$L_3 : \{a^n b^n c^n \mid n \geq 0\}$

Du aaa bbbccc

→ até o 1º a < achou b ou c rejeita
                achou ⊔ aceita

a → A

→ até o 1º b — achou a rejeita

b → B

→ até o 1º c → chou a rejeita

c → C

← até D