

Procedura

O procedură constă într-o mulțime finită de instrucțiuni care pot fi executate mecanic, într-un timp determinat, cu o cantitate fixă de efort. Proc. poate avea \neq nr. intrări / ieșiri.

Algoritm

O procedură se oprește pt o anumită intrare, dc $\exists t$ (finit) at. după executarea a t instrucțiuni elementare, se nu mai există o altă instr. de executat, fie instrucțiunea 'halt' a fost executată. Procedura care se oprește pt toate intrările \rightarrow algoritm.

Funcție recursive

O procedură definește o relație între intrări \rightarrow ieșiri numită funcție recursivă parțială.

Dc. procedură \rightarrow algoritm \rightarrow funcție recursivă totală.

Procedură \rightarrow defini un limbaj: pt un sir x , procedură decide dc $x \in L$ (limbaj)
Omulime def. pînă n-o poc \rightarrow recursiv enumerabilă
alg \rightarrow recursivă.

\exists un număr mare de notății formale ut. pt. a descrie o procedură:

- 1) M. Turing. (Turing, 1936-1937)
- 2) gramatică Chomsky ϕ (Chomsky, 1959-1963)
- 3) Alg. Markov (Markov, 1951)
- 4) Calcul Lambda (Church, 1941)
- 5) ~~Sisteme~~ Post
- 6) Tag
- 7) Cele mai multe limbaje de programare

Ierarhia Chomsky

gramaticile pot fi clasificate în raport cu forma regulilor lor:

Fie $G = (V, \Sigma, R, S)$

1) Regulate (la dreapta)

$$A \rightarrow \alpha B \mid \alpha, \alpha \in \Sigma^*, A, B \in V - \Sigma$$

2) Independente de context

$$A \rightarrow \alpha, A \in (V - \Sigma), \alpha \in V^*$$

3) Dependente de context

$$\alpha A \beta \rightarrow \alpha \delta \beta$$

$$|\alpha A \beta| \leq |\alpha \delta \beta|$$

$$\alpha, \beta \in V^*, A \in V - \Sigma$$

4) O gramatică care nu are restricții la 1. ÷ 3 \Rightarrow G. nerestricționată

$$V^* (V - \Sigma) V^* \times V^*$$

Pt. fix. clasă de gram. din ierarhia Chomsky \rightarrow clasă de acceptoare care acc. aceiași clasă de limbaje:

- (1) $L \rightarrow \text{regulat} \Rightarrow \text{A.F. (D/N)}$
- (2) $L \rightarrow \text{LIC} \Rightarrow \text{A.P.D. (nondeterminist)}$
- (3) $L \rightarrow \text{LDC} \Rightarrow \text{2-way nondeterministic linear bounded automaton}$
- (4) $L \rightarrow \text{recursiv enumerabil} \Rightarrow \text{M. Turing}$

ex: GDC; ? G on $L(G) = \{a^n b^m c^n \mid n \geq 0\}$.

$$S \rightarrow a S B C / a t C$$

$$c B \rightarrow B C$$

$$t B \rightarrow t t$$

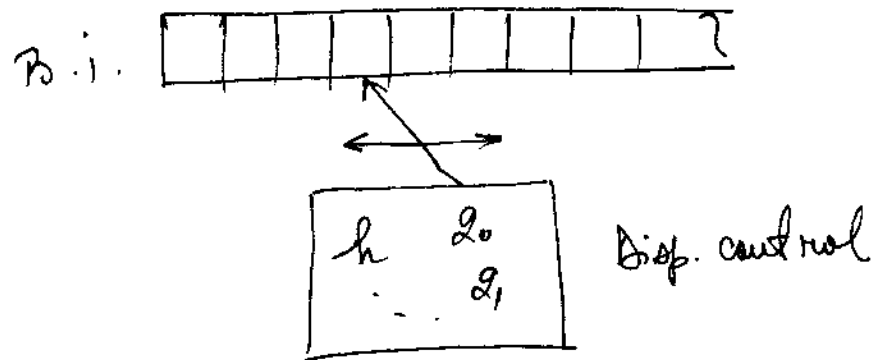
$$c C \rightarrow c c$$

$$t C \rightarrow t c$$

$$S \Rightarrow a S B C \Rightarrow a a t c B C \Rightarrow a a t t B C C \Rightarrow a a t t c c$$

Mașina Turing

M.T. constă dintr-un disp. control și o bandă. Comunicarea dintre cele 2 componente este făcută printr-un cap care poate citi / înlocui simbol pe bandă.



Mecanismul de control acționează în pași discreti; la fec. pas execută 2 acțiuni: într-un mod care depinde de starea curentă și de simbolul de pe bandă indicat de capul citire / scriere:

1) pune disp. de control într-o nouă stare

2) Re

a) scrie un simbol pe bandă în poz. curentă, înlocuindu-l pe cel existent

b) deplas. cap. o poz. la stg / dr

B.i. \rightarrow are capăt stg, se încheie la dr. indefinit

Dc. M.T. încearcă să mute capul peste limite stg. a B.i. \rightarrow încearcă să funcționeze.

Există o stare specială - de halt - $h \Rightarrow$ încheierea aghiilor M.T.

Convenție \rightarrow B.i. conține în fiecare pătrat \rightarrow '#'

L, R \rightarrow deplas. capului o poziție la stg / dr.

Def.

O M. Turing este un tuple (K, Σ, δ, s) , unde:

$K \rightarrow$ mulțimea finită a stărilor, $\neq h$

$\Sigma \rightarrow$ alfabetul de intrare, $\ni \#$, $\ni L, R$

$s \in K \rightarrow$ st. inițială

$\delta: K \times \Sigma \rightarrow (K \cup \{h\}) \times (\Sigma \cup \{L, R\})$

Dc. $q \in K$, $a \in \Sigma$, $\delta(q, a) = (p, h)$

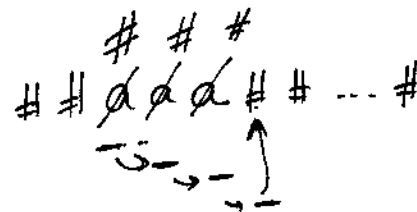
ex: ? M.T. $M = (K, \Sigma, \delta, \Delta)$ care înlocuiește toate simbolul 'a' cu '#' mergând spre dr.

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

$$\Sigma = \{a, \#\}$$

$$\Delta = q_0$$

q	∇	$\delta(q, \nabla)$
q_0	a	$(q_1, \#)$
q_0	#	$(q_0, \#)$
q_1	a	(q_0, a)
q_1	#	(q_0, R)



ex: ? M.T. $M = (K, \Sigma, \delta, \Delta)$ care se deplas. la stg. până la # și se oprește

$$K = \{q_0\}$$

$$\Sigma = \{a, \#\}$$

$$\Delta = q_0$$

q	\triangleright	$\delta(q, \triangleright)$
q_0	a	(q_0, L)
q_0	$\#$	$(h, \#)$

Def.

Configurația unei M.T. $M = (K, \bar{Z}, \delta, \Delta)$ este un membru din:

$$(K \cup \{h\}) \times \bar{Z}^* \times \bar{Z} * (\bar{Z}^* (\bar{Z} - \{ \# \}) \cup \{e\})$$

Def.

Fie $M = (K, \bar{Z}, \delta, \Delta)$ și fie $(q_1, w_1, a_1, u_1), (q_2, w_2, a_2, u_2)$ config. ale lui M .

Atunci:

$$(q_1, w_1, a_1, u_1) \xrightarrow{M} (q_2, w_2, a_2, u_2)$$

$$\Rightarrow h \in \bar{Z} \cup \{L, R\}$$

$$\delta(q_1, a_1) = (q_2, h)$$

$$1) \ell \in \bar{Z}, w_1 = w_2, u_1 = u_2, a_2 = \ell$$

$$2) \ell = L, w_1 = w_2 a_2$$

$$a) u_2 = a_1 u_1, \text{ dc } a_1 \neq \# , u_1 \neq e$$

$$b) u_2 = e, a_1 = \#, u_1 = e$$

$$3) \ell = R, w_2 = w_1 a_1$$

$$a) u_1 = a_2 u_2$$

$$b) u_1 = u_2 = e, a_2 = \#$$

Obs:

$\ell = L, w_1 = e \Rightarrow (q_1, w_1, a_1, u_1)$ nu duce la nicio config. pd. c \bar{e}

$\forall w_2 \in \bar{Z}^*, a_2 \in \bar{Z}$ at $w_1 = w_2 a_2$

\Rightarrow M.T. se agat \bar{c} .

Def.

\vdash_m^* \Rightarrow închiderea reflexivă și tranzitivă a \vdash_m

Calcul cu M. Turing

Convenții:

- simbol de intrare este scris aliniat la limita stg. a benzii cu capul poziționat pe '#' care delim. dr. simbol
- aceeași convenție pt poz. capului pt rezultatul calculat.

Def:

Fie $\Sigma_0, \Sigma_1 \neq \emptyset$. Fie $f: \Sigma_0^* \rightarrow \Sigma_1^*$. M. Turing $M = (K, \Sigma, \delta, \Delta)$ calculează f ,
dc $\Sigma_0, \Sigma_1 \subseteq \Sigma$, $\forall w \in \Sigma_0^*$, $f(w) = u$ atunci $(\Delta, \#w\#) \vdash_M^* (h, \#u\#)$

Dc $\forall M \Rightarrow f \rightarrow f$ funcție Turing calculabilă.

ex: $\Sigma_0 = \Sigma_1 = \{a, b\}$

$f: \Sigma_0^* \rightarrow \Sigma_1^*$

$f(w) = \bar{w}$, \bar{w} este rezultatul înlocuirii $a \leftrightarrow b$ în w .

$$M = (K, \Sigma, \delta, \Delta)$$

$$K = \{q_0, q_1, q_2\}$$

$$\Sigma = \{a, b, \#\}$$

$$\Delta = q_0$$

$\delta:$	q	Σ	$\delta(q, \Sigma)$
	q_0	a	(q_1, L)
	q_0	b	(q_1, L)
	q_0	$\#$	(q_1, L)
	q_1	a	(q_0, b)
	q_1	b	(q_0, a)
	q_1	$\#$	(q_2, R)
	q_2	a	(q_2, R)
	q_2	b	(q_2, R)
	q_2	$\#$	$(b, \#)$

$(q_0, \# a a b \#) \xrightarrow{\gamma} (q_1, \# a a \underline{b})$

$\xrightarrow{\gamma} (q_0, \# a a \underline{a})$

$\xrightarrow{\gamma} (q_1, \# a \underline{a} a)$

$\xrightarrow{\gamma} (q_0, \# a \underline{b} a)$

$\xrightarrow{\gamma} (q_1, \# \underline{a} b a)$

$\xrightarrow{\gamma} (q_0, \# \underline{t} t a)$

$\xrightarrow{\gamma} (q_1, \# \underline{t} t a)$

$\xrightarrow{\gamma} (q_2, \# \underline{t} t a)$

$\xrightarrow{\gamma} (q_2, \# t t a \#)$

$\xrightarrow{\gamma} (h, \# t t a \#)$

Noțiunea de funcție calculabilă pe siruri poate fi extinsă:

$$f: (\Sigma_0^*)^k \rightarrow \Sigma_1^*, k > 0$$

$$M = (K, \Sigma, \delta, \Delta), \Sigma_0, \Sigma_1 \subseteq \Sigma.$$

$$\forall w_1, \dots, w_k \in \Sigma_0^*, f(w_1, \dots, w_k) = u, \text{ atunci } (\Delta, \#w_1\#w_2\#\dots\#w_k\#) \vdash_M^* (h, \#u\#).$$