

1. Turingov stroj

1.1. UVOD

Obrada podataka: svrsishodna djelatnost koja ima za cilj da se iz raspoloživih podataka dobije **tražena informacija**

Komponente:

- podatak
- algoritam
- **izvršitelj**

- Podatak – objekt u obradi
- Algoritam – precizna uputa (“recept”) koja opisuje transformaciju ulaznih podataka u traženi rezultat
- Izvršitelj ?

Algoritam (Abu Ja'far Mohammed ibn al- Khowarizmi; *algorism*) – postupci transformacije grupirani u **korake** algoritma

Svojstva algoritma:

- Određenost
- Konačnost
- Širina primjene (područje uporabe)

Primjer: Euklidov algoritam (Euclid, 400 B.C.) – rješava zadatke tipa: Za dana dva prirodna broja a , b treba naći njihovu najveću zajedničku mjeru.

1. korak: Promatrajte dva broja: a i b .

Prijeđite na sljedeći korak;

2. korak: Usporedite brojeve a i b .

Prijeđite na sljedeći korak;

3. korak: Ako su promatrani brojevi jednaki, svaki daje **traženi rezultat** – obustavite postupak računanja. Ako brojevi nisu jednaki prijeđite na sljedeći korak;

4. korak: Ako je prvi promatrani broj manji od drugog, zamijenite im mjesta. Prijeđite na sljedeći korak;

5. korak: Oduzmite drugi broj od prvog i promatrajte taj drugi broj i ostatak. Prijeđite na korak **2**.

Zadaci:

1. Naći najveću zajedničku mjeru brojeva $a = 28$
 $b = 124$.
2. Naći najveću zajedničku mjeru brojeva 250 i 111.
3. Za neki $n \in \mathbb{Z}^+$ naći najveću zajedničku mjeru dvaju pozitivnih brojeva $8n+3$ i $5n+2$.

Euklidov algoritam namijenjen izvršitelju - **čovjeku**

Kako bi izgledao algoritam namijenjen stroju (računalu)?

Zadatak 4. Napisati program u višem programskom jeziku (npr. u C, C++ ili Pascalu) za Euklidov algoritam.

Program EuclideanAlgorithm (input, output);

Var

a, b, c, d, r: integer;

Begin

Writeln ('Zelimo naci najveću zajedničku mjeru, odnosno najveći');

Writeln ('djelitelj dvaju pozitivnih brojeva a i b');

Write ('a = ');

Read (a);

Write ('b = ');

Read (b);

r := a Mod b;

d := b;

While r < 0 do

Begin

c := d;

d := r;

r := c Mod d;

End

Writeln ('Najveća zajednička mjera brojeva 'a : 0,' i 'b, 0,' je 'd:0');

End.

Opaska: u programu je namjerno ugrađena logička pogreška – nađite je

Drugi pristup:

Računanje je proces određivanja izlazne supstitucije, za zadanu određenu ulaznu supstituciju, koja se pokorava svim specifičnim svojstvima problema.

(Garey & Johnson, 1979)

izlazna supstitucija?
ulazna supstitucija?
specifična svojstva problema?

Primjer: Problem naprtnjače

Zadano: i) Naprtnjača nosivosti (kapaciteta) C [kg];
ii) N objekata (predmeta) koje bi trebalo ponijeti;
Svaki je predmet određen svojom težinom w_k i
vrijednosti v_k ; $k = 1, 2, \dots, N$;

Potrebno je naći **udio F_k svakog predmeta tako da:**

- a) ne prenatrpate naprtnjaču;
- b) maksimizirate vrijednost koju nosite

(Opaska: Predmeti se mogu rastaviti na dijelove čija je vrijednost proporcionalna njihovoj težini)

Parametri i varijable problema:

$C, N, w_1, v_1, F_1, \dots, w_N, v_N, F_N$

Primjer ulazne supstitucije:

$C = 14 \text{ kg}, N = 3, w_1 = 4 \text{ kg}, v_1 = 30 \$$

$w_2 = 6 \text{ kg}, v_2 = 48 \$$

$w_3 = 7 \text{ kg}, v_3 = 50 \$$

Rješenje (izlazna supstitucija):

U naprtnjaču stavljamo:

- Cijeli predmet 1 i cijeli predmet 2
- 4/7 predmeta 3

Ukupna vrijednost u naprtnjači: 106,57 \$

Algoritam:

- **korak:** Razvrstaj objekte na temelju omjera vrijednosti i težine
(predmet 2 = $48/6 = 8$ \$/kg; predmet 1 = $30/4 = 7,5$ \$/kg;
predmet 3 = $50/7 = 7,14$ \$/kg)

- **korak:** Ponavljaj sve dok se naprtnjača ne prenatrpa:

Iz skupa predmeta uzmi predmet s najvećim omjerom vrijednost/težina i smjesti ga cijelog u naprtnjaču.

- **korak:** Izvadi iz naprtnjače posljednji predmet i razdijeli ga tako da upravo njegovi dijelovi popune naprtnjaču.

Algoritam je namijenjen čovjeku.

Kako bi izgledao algoritam namijenjen stroju?

- Računalo;
- Inteligentni stroj s vidnom percepcijom i drugim senzorima (npr. težine, pritiska) opremljen mehaničkim hvataljkama, bazom znanja, strojem za zaključivanje i sučeljem za razumijevanje prirodnog jezika;

1.2. Specijalan slučaj: Izvršitelj **Turingov stroj**

(A. M . Turing, “On Computable Numbers, with an Application to the Entscheidungsproblem”, *Proc. of the London Math. Society*, 2nd Series, 42: pp.230-265,1936.)

Raščlanjivanje računskog postupka na **vrlo jednostavne, elementarne operacije** kod Turingovog stroja (TS) svedeno je do graničnih mogućnosti.

Definicija TS iz teorije automata:

$$TS = (Q, S, T, b, q_0, q_f, \delta)$$

Komponente Turingovog stroja:

Vanjska memorija TS je s obje strane neograničena vrpca (traka) podijeljena na polja – svako polje može sadržavati **samo jedan** znak.

TS barata konačnim brojem **znakova (simbola)**:

s_1, s_2, \dots, s_k koji oblikuju **vanjsku abecedu stroja**:

$$S = \{s_1, s_2, \dots, s_k\}$$

U skupu S nalazi se i **prazni (pusti) simbol b:**

-upisivanjem praznog znaka u bilo koje polje vrpce briše se znak koji se tamo nalazio;

(Specifičnost simbola b!)

Izraz zapisan na vrpci predstavljen je konačnim nizom znakova vanjske abecede (različitih od praznog znaka);

Skup $T = \{t_1, t_2, \dots, t_{|T|}\}$ – skup simbola koji se pojavljuju na vrpci bez praznih simbola:

$$T = S/b$$

Rad Turingovog stroja:

- Na početku rada stroja na vrpici se nalazi zapisan **početni izraz (početna, ulazna informacija)**;
- Rad stroja se odvija u **taktovima**
(Odvijanjem taktova stroj početnu informaciju preoblikuje u (među)informaciju, itd.; na kraju svakog takta znaci zapisani na vrpici stroja oblikuju odgovarajuću (među)informaciju);

\mathcal{A} – početna informacija zapisana na vrpici na početku rada stroja

Dva slučaja:

- Nakon konačnog broja taktova stroj se zaustavlja dajući Stop-signal i pri tomu je na vrpici zapisana informacija \mathcal{B} .

Stroj je primjenjiv za početnu informaciju \mathcal{A} i prerađuje je u informaciju ili rezultat \mathcal{B} .

2. Stroj nikad ne staje i nikad ne daje Stop-signal.

Stroj nije primjenjiv za početnu informaciju *\mathcal{A}* ..

$Q = \{q_0, q_1, q_2, \dots, q_{|Q|-1}\}$ – skup **unutarnjih stanja**

q_0 - početno stanje stroja (stanje stroja na početku 1.takta);

q_f - konačno stanje stroja (!);

Sustav elementarnih operacija i naredbi:

- u svakom taktu naredba propisuje samo zamjenu

pojedinačnog znaka s_i , upisanog u promatranom

polju, nekim drugim znakom s_j :

- ako je $j = i$ sadržaj se promatranog polja ne mijenja;

- ako je $s_j = b$ sadržaj se promatranog polja briše.

- ako je $j \neq i$ tada se s_i zamjenjuje sa s_j ;

Turingov stroj ima **glavu za čitanje i pisanje (R/W head)**.²⁰

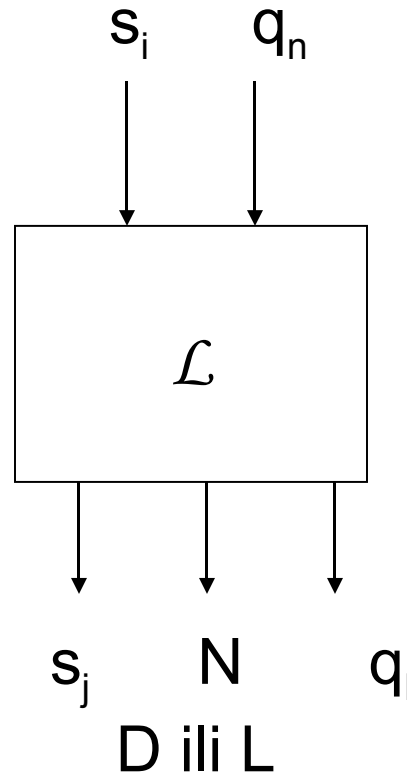
Glava za čitanje i pisanje, pri prijelazu s jednog takta na drugi, može se pomaknuti naviše za jedno polje (adresa se promatranog polja, može promijeniti za +1 ili -1 ili 0);

Tri “standardne” adresne naredbe:

- D – promatraj desno susjedno polje;
- L – promatraj lijevo susjedno polje;
- \emptyset ili N – ostani na istom polju.

Skup naredbi za pomak glave $P = \{N \text{ ili } \emptyset, D, L\}$

Obrada informacije u Turingovom stroju odvija se u **logičkom bloku \mathcal{L}** koji se može nalaziti u nekom od konačnog broja unutarnjih stanja $Q = \{q_0, q_1, q_2, \dots, q_{|Q|-1}\}$.



Ulazna dvojka: (s_i, q_n)

Izlazna trojka: (s_j, p, q_l)

$p \in P$

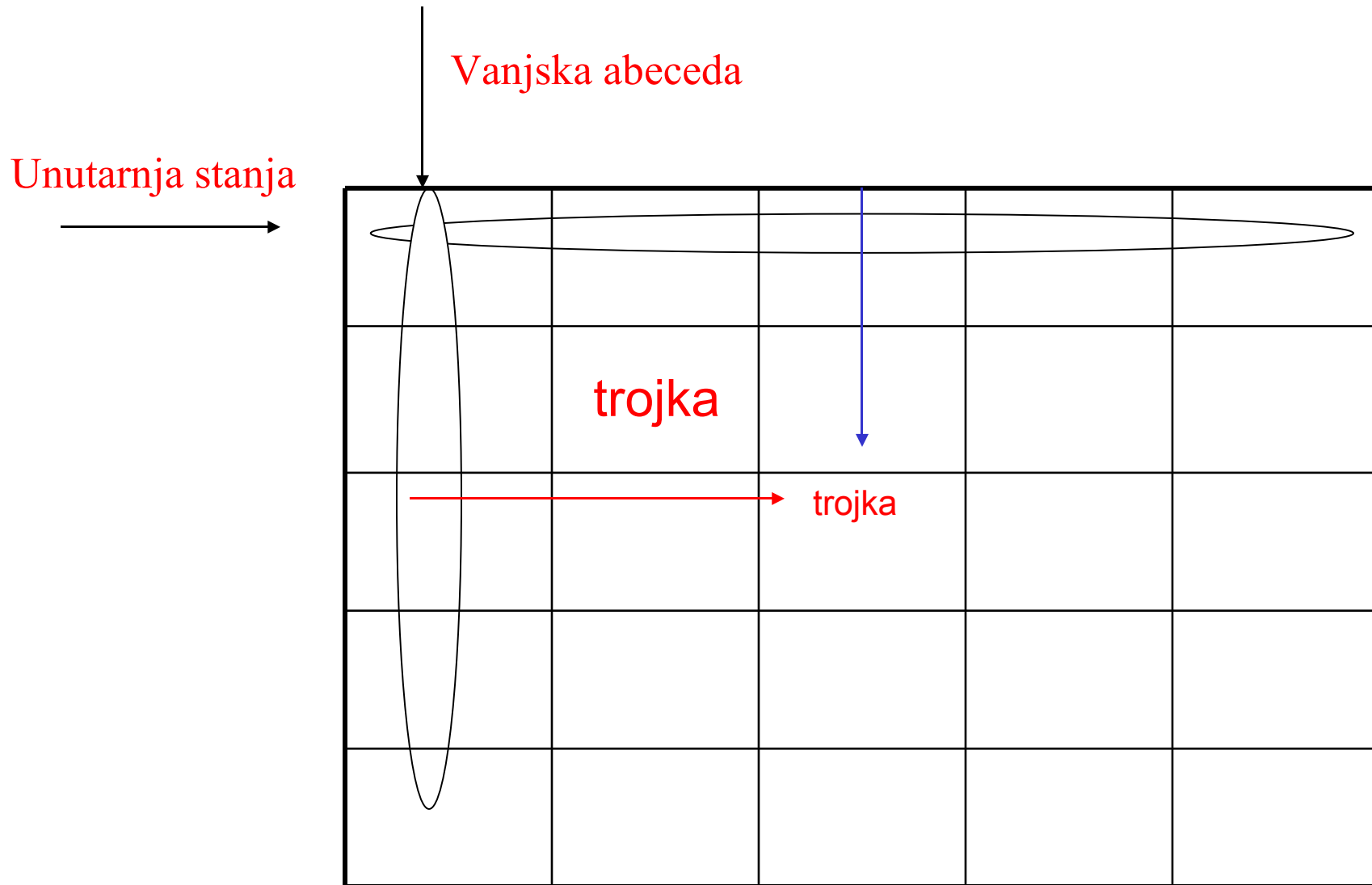
Logički blok \mathcal{L} ostvaruje funkciju koja svakoj dvojci na ulazu pridružuje izlaznu trojku.

δ - logička funkcija stroja

$$\delta: S \times Q \rightarrow S \times P \times Q$$

Logička se funkcija stroja može prikazati tablicom koja se naziva **funkcionalna shema TS**

	q_1	q_2	$q_3 \dots$	
\wedge	$q_4 D \wedge$	$q_3 L $	$q_1 D \alpha$	\dots
$ $	$q_2 N \alpha$	$q_4 D \wedge$	$q_3 N \beta$	\dots
α	$q_1 D \alpha$	$q_2 D \alpha$	$q_4 N \alpha$	\dots
β	\dots	\dots	\dots	\dots



Turingov stroj:

$$TS = (Q, S, T, b, q_0, q_f, \delta)$$

Q – skup unutarnjih stanja stroja;

S – skup simbola vanjske abecede;

$T = S \setminus b$;

b – prazan simbol;

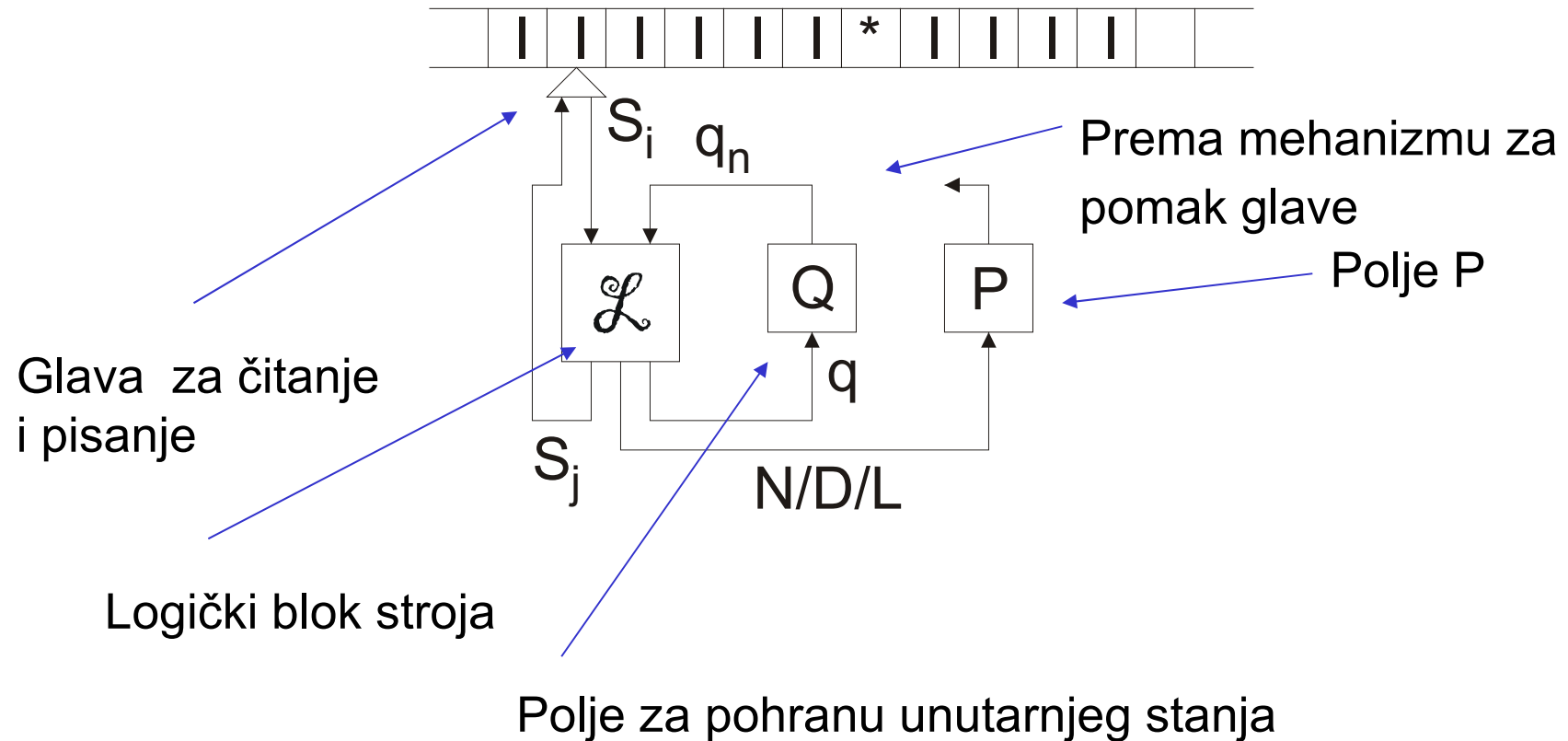
q_0 – početno stanje stroja;

q_f – konačno stanje stroja;

δ - logička funkcija stroja

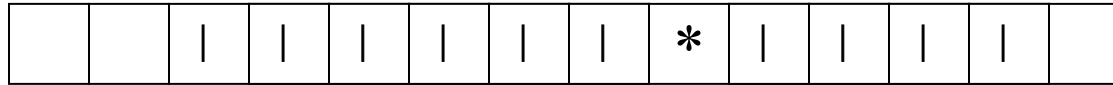
Shematski prikaz TS-a:

Vanjska memorija



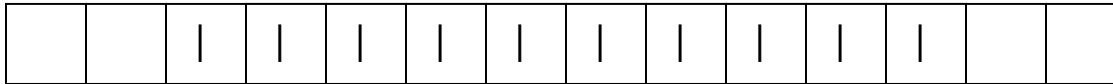
1.3. Primjeri programa za TS

Primjer 1: Zbrajanje Turingovim strojem



q_0

Početno stanje

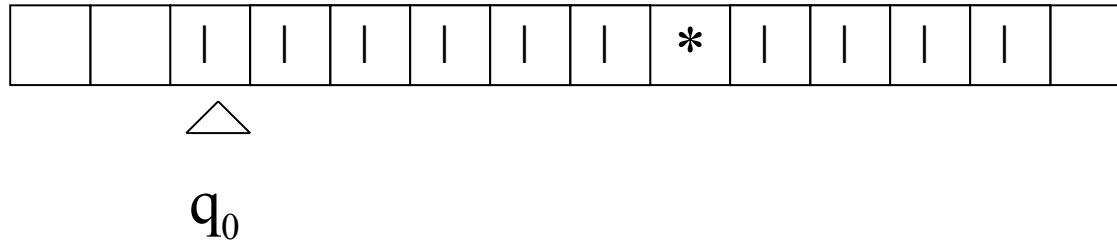


Konačno

(željeno) stanje

Zadatak: Napisati “program” za TS koji zbraja štapiće!

Rješenje:



Vanjska abeceda: $S = \{ |, *, b \}$

Skup unutarnjih stanja: $Q = \{ q_0, q_1, q_2, q_f = ! \}$

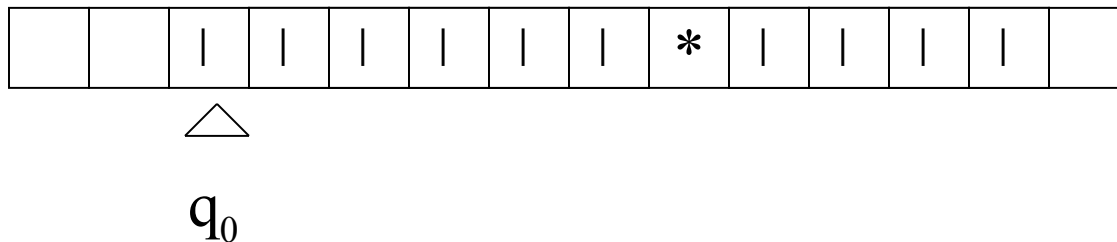
Konačno rješenje:

	q_0	q_1	q_2
	$q_2 Db$	$q_1 L $	$q_2 D $
b	$q_0 Db$	$q_0 Db$	$q_1 N $
*	$!Nb$	$q_1 L^*$	$q_2 D^*$

$$b = \wedge$$

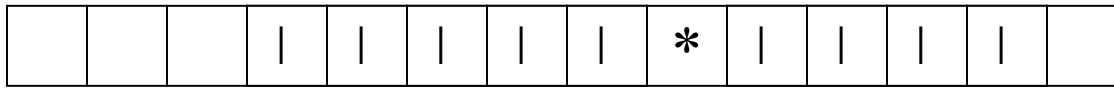
Analiza rješenja:

Početno stanje:



Prvi takt: Promatrani štapić se briše i glava se pomiče udesno te stroj prelazi u stanje q_2

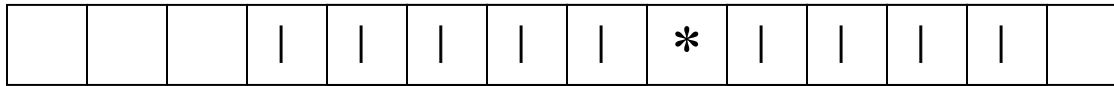
$$(|, q_0) \rightarrow (q_2, D, b)$$



q_2

Drugi takt:

$(|, q_2) \rightarrow (q_2, D, |)$



q_2

Treći takt:

$$(|, q_2) \rightarrow (q_2, D, |)$$

Četvrti takt:

$$(|, q_2) \rightarrow (q_2, D, |)$$

Peti takt:

$$(|, q_2) \rightarrow (q_2, D, |)$$

...

7. takt:

$$(*, q_2) \rightarrow (q_2, D, *)$$

...

11. takt:

$$(|, q_2) \rightarrow (q_2, D, |)$$

12. takt:

$$(b, q_2) \rightarrow (q_1, N, |)$$

13. takt:

$$(|, q_1) \rightarrow (q_1, L, |)$$

14.takt:

$$(|, q_1) \rightarrow (q_1, L, |)$$

15.takt:

$$(|, q_1) \rightarrow (q_1, L, |)$$

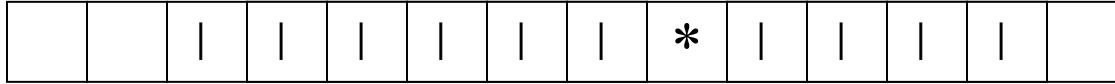
...

...

Definicija k-te konfiguracije Turingovog stroja:

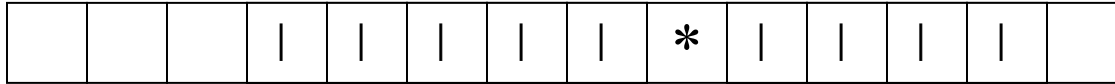
Slika vrpce s informacijom koja se na njoj nalazi u početku k-tog takta, pri čemu je ispod promatranog polja zapisan znak stanja koji ulazi u logički blok \mathcal{L} u početku k-tog takta.

Ilustracija k-tih konfiguracija:



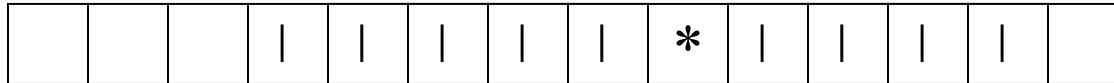
q_0

1. konfiguracija



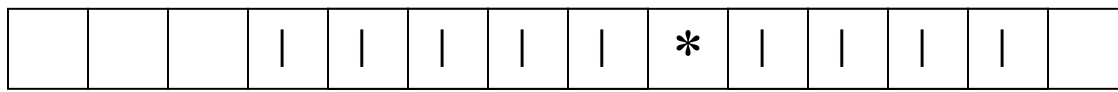
q_2

2. konfiguracija



q_2

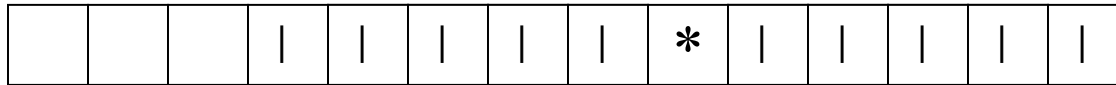
3. konfiguracija



12. konfiguracija



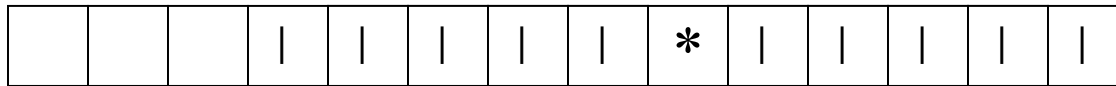
q_2



13. konfiguracija



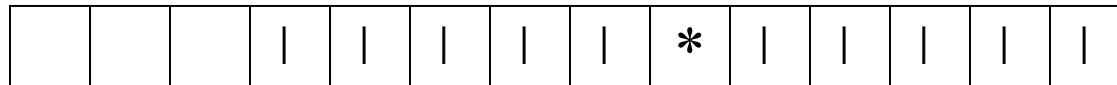
q_1



24. konfiguracija



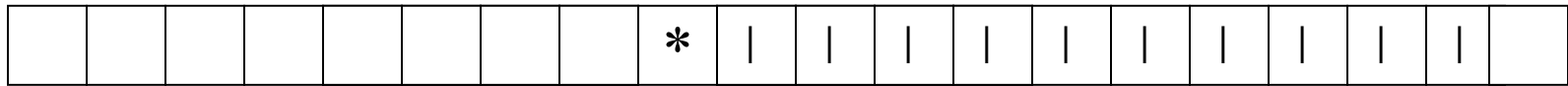
q_1



25. konfiguracija

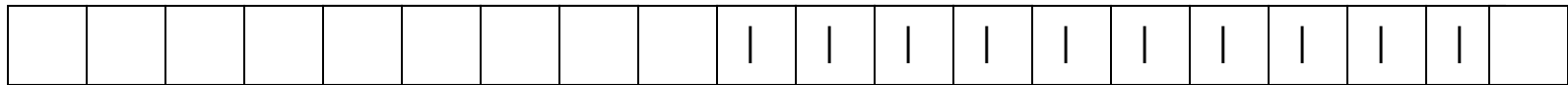


q_0



Predzadnja konfiguracija: q_0

Zadnja konfiguracija:



q_f

Primjer 2:

Inkrementiranje broja predodčenog u dekadnom brojevnom sustavu

$/n \rightarrow n + 1/$:

Zadan je dekadski zapisan broj n , treba naći dekadni zapis broja $n+1$

Vanjska abeceda: $0, 1, 2, \dots, 9, b$

Unutarnja stanja stroja: q_0 (početno i ujedno radno stanje) i q_f – konačno stanje stroja.

Rješenje: Funkcionalna shema stroja

	q_0
0	!N1
1	!N2
2	!N3
3	!N4
4	!N5
5	!N6
6	!N7
7	!N8
8	!N9
9	q_0L0
???	????

	3	2	9	9				
--	---	---	---	---	--	--	--	--



q_0

	3	2	9	0				
--	---	---	---	---	--	--	--	--



q_0

	3	2	0	0				
--	---	---	---	---	--	--	--	--



q_0

	3	3	0	0				
--	---	---	---	---	--	--	--	--

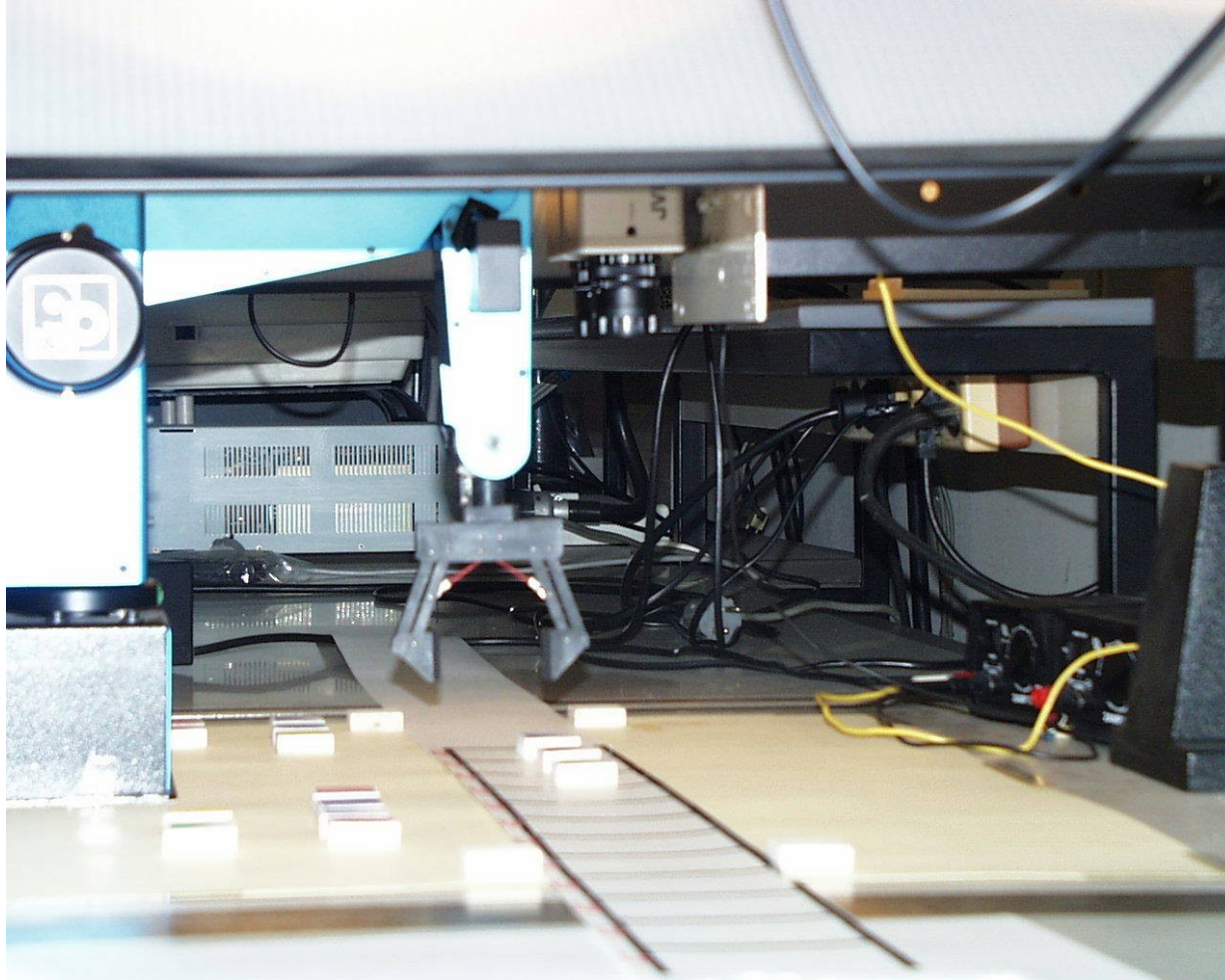


!

S. Ribarić, Arhitektura računala 2

	q_0
0	!N1
1	!N2
2	!N3
3	!N4
4	!N5
5	!N6
6	!N7
7	!N8
8	!N9
9	q_0L0
b	!N1

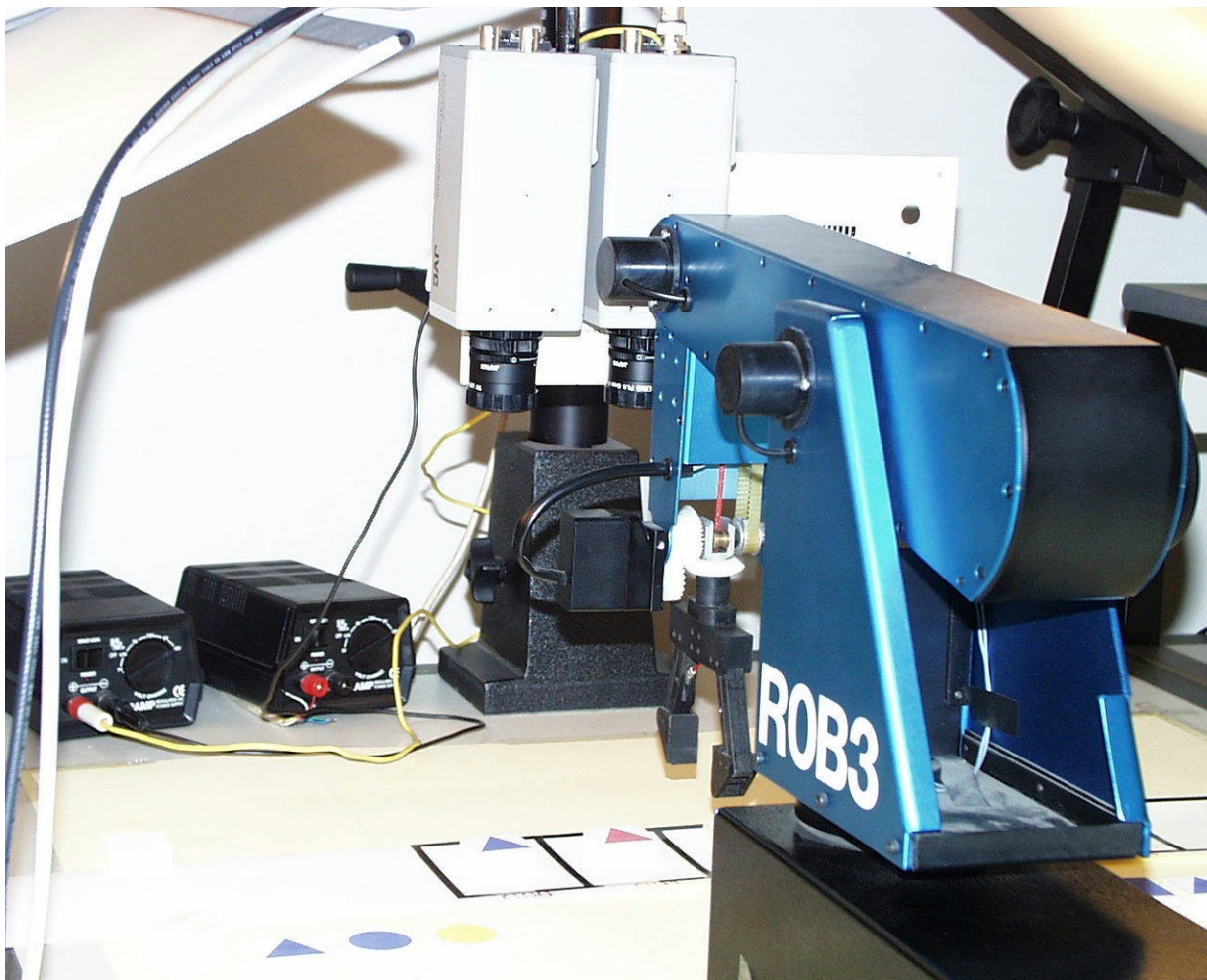
Fizička izvedba Turingovog stroja
(S. Ribarić, D. Krleža, N. Pavešić, “ A Turing Machine with Robot Arm and Eye”,
Proc. of the 5th IEEE Conf. on Intelligent Eng. Systems, INES 2001, Helsinki, 273-276.)

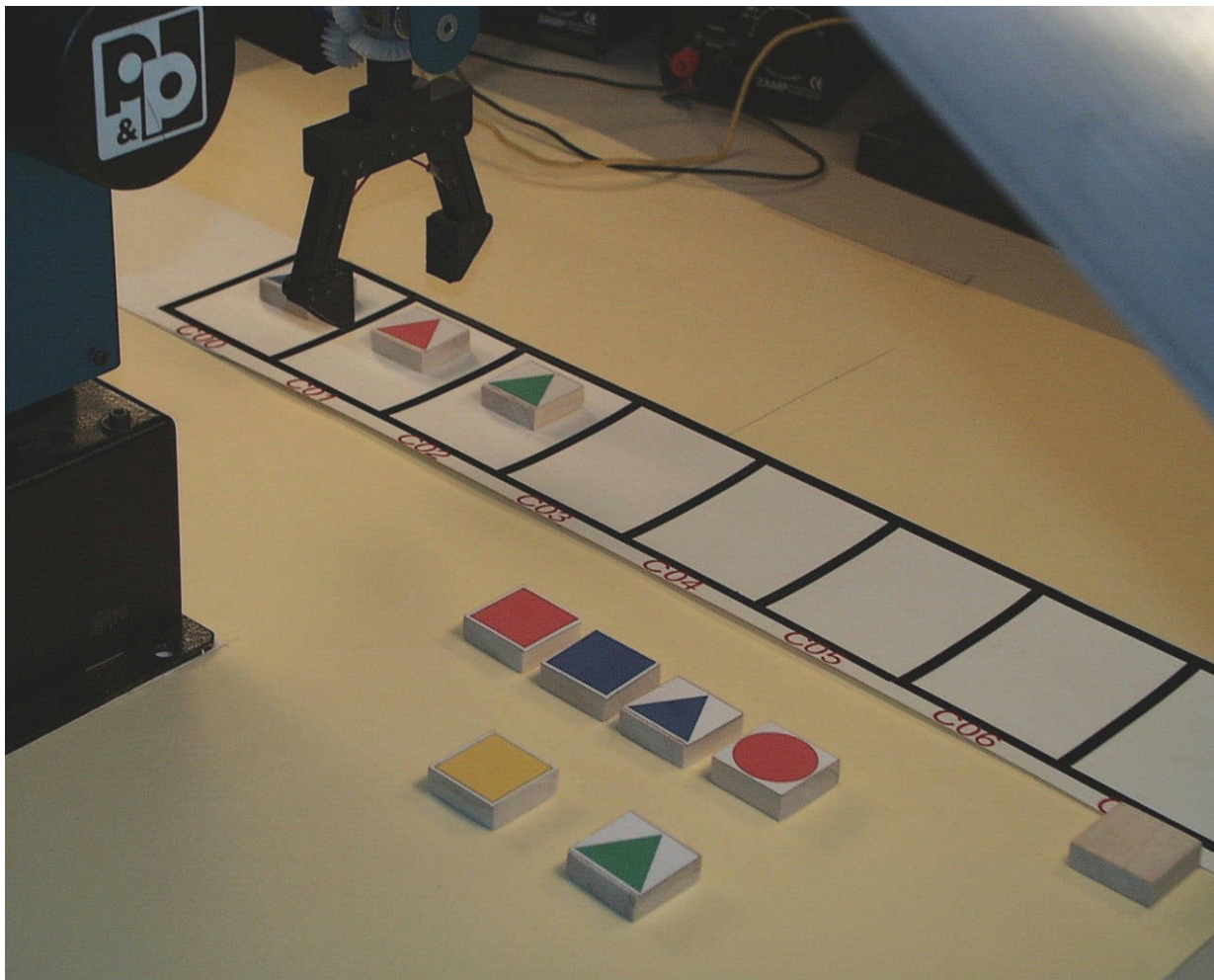


S. Ribarić, Arhitektura računala 2

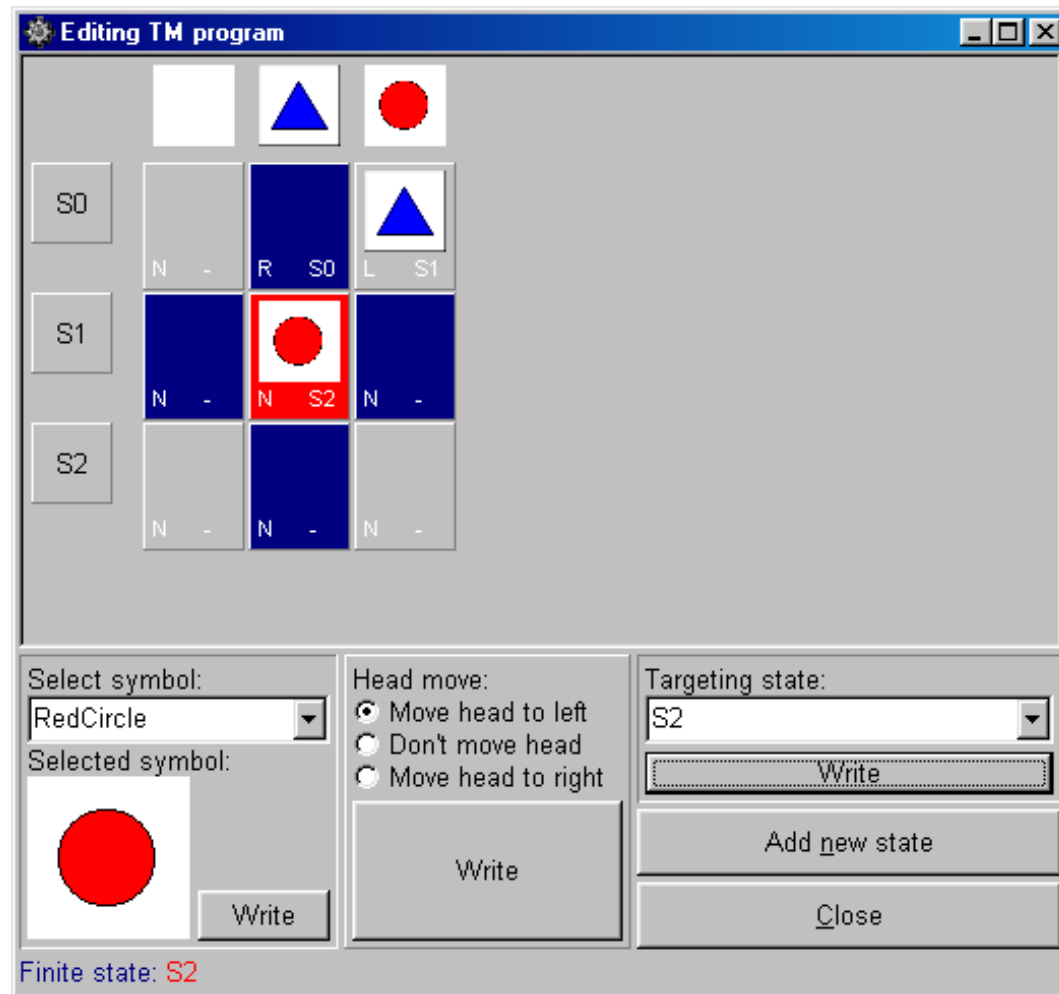
U fizičkoj izvedbi Turingovog stroja koriste se zasadi *raspoznavanja uzoraka, robotike i umjetne inteligencije*

- Glava za čitanje i pisanje: kombinacija robotske ruke i TV kamere;
- Vanjska memorija: Papirna vrpca s označenim poljima;
- Umjesto pomaka glave za čitanje i pisanje pomiče se vrpca;
- Simboli vanjske abecede su pločice s geometrijskim likovima u boji;
- Logički blok ostvaren osobnim računalom;
- Postojanje “poola” simbola vanjske abecede;





Izgled korisničkog sučelja za programiranje TS-a:



Zadaci:

5. Napisati program za TS koji pretvara binarni zapis broja u potpuni ili dvojni komplement. Glava za čitanje i pisanje (R/W) nalazi se početno na krajnje lijevoj binarnoj znamenci niza.
6. Napisati program za TS koji inkrementira vrijednost broja predodčenog u oktalnom sustavu. Glava R/W nalazi se početno na krajnje desnoj znamenci niza.
7. Napisati program za inkrementiranje rimskih brojeva u rasponu od I – III (elementi vanjske abecede su b, I, II, III, IV). Glava za čitanje i pisanje (R/W) nalazi se početno na krajnje lijevoj binarnoj znamenci niza. Odrediti 4. konfiguraciju stroja za početni niz III.

8. Napisati program za TS koji pronalazi zapis na traci u slučaju kada nije poznata početna pozicija glave R/W niti njen relativni položaj u odnosu na zapisani niz.