

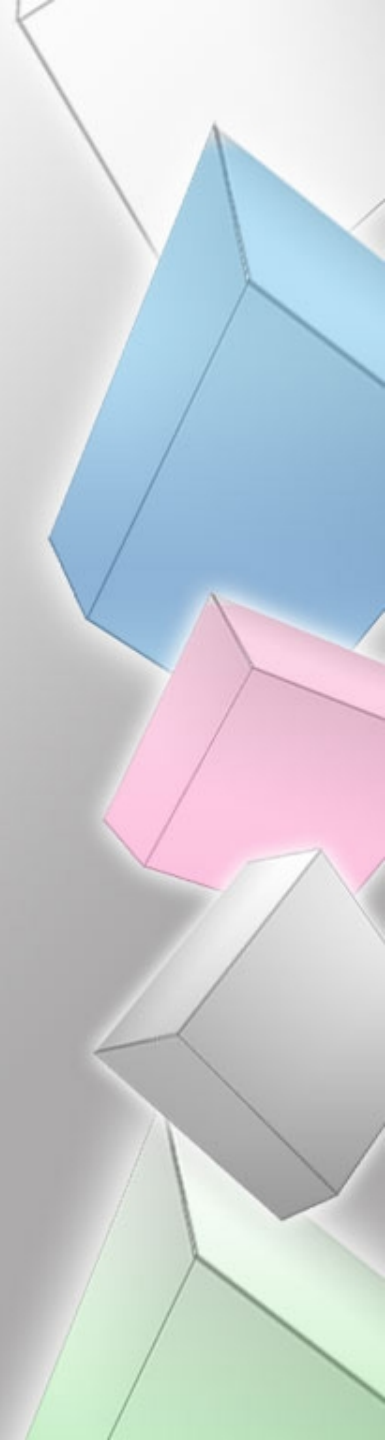
Nikola Stojić

Formalni Opis Tjuringove

Mašine

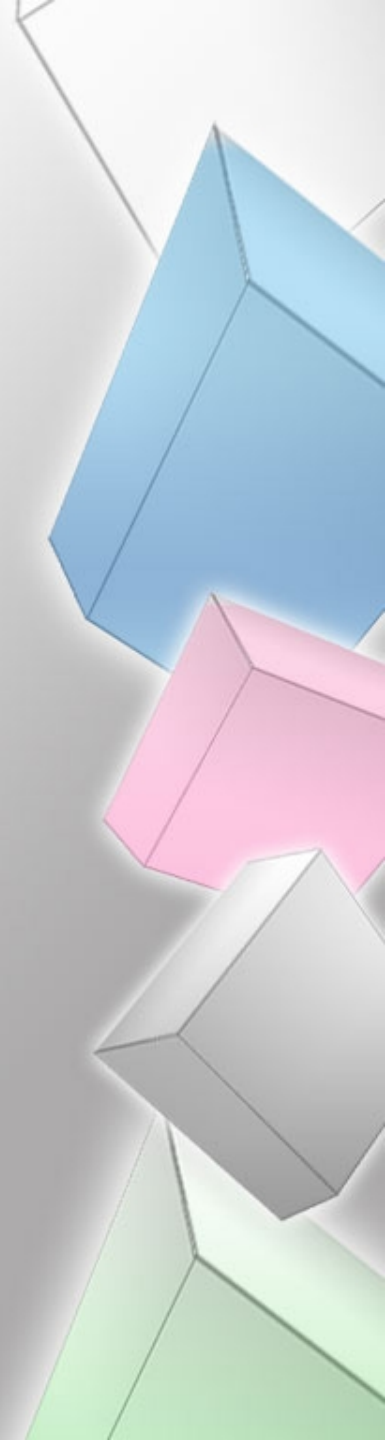
Profesor: Zoran Ognjanović

◆ Neka su:



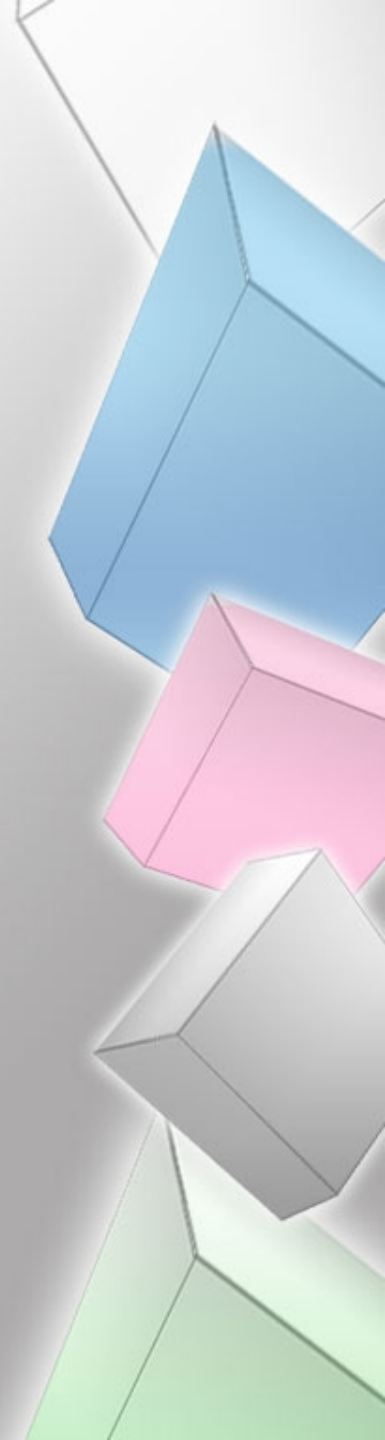
◆ Neka su:

- q_0 početno stanje, $q_0 \in S$



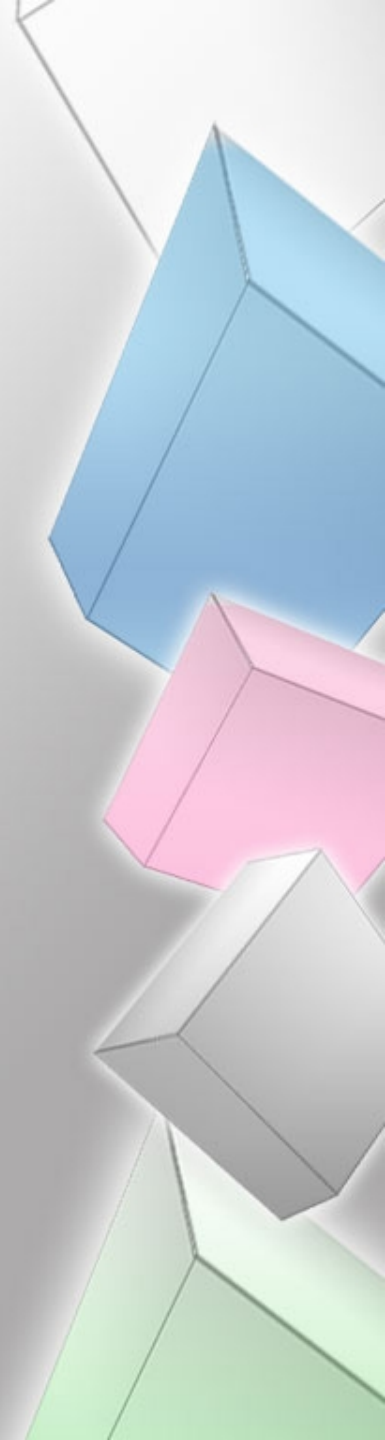
◆ Neka su:

- q_0 početno stanje, $q_0 \in S$
- q_z je završno stanje, $q_z \in S$



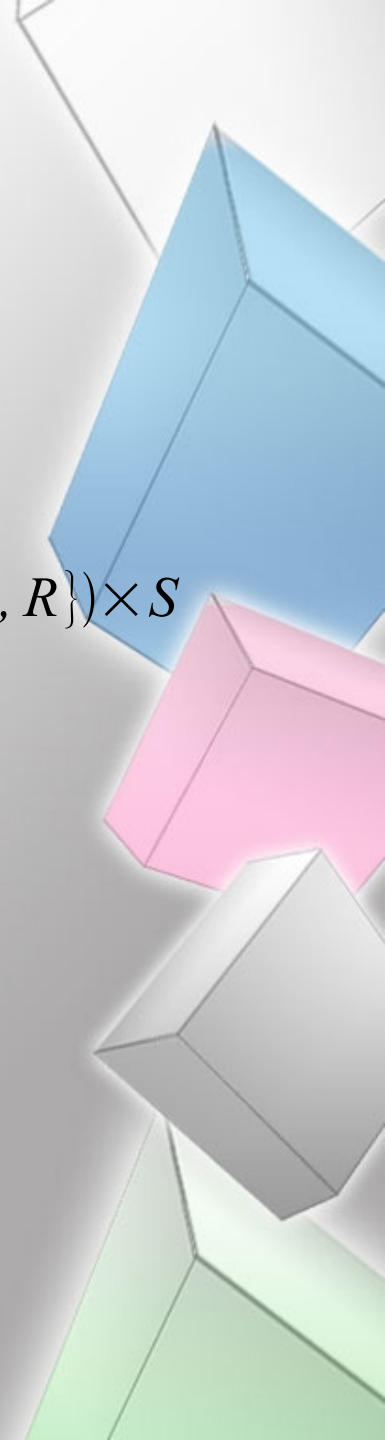
♦ Neka su:

- q_0 početno stanje, $q_0 \in S$
- q_z je završno stanje, $q_z \in S$
- $A = \{1, 0\}$ alfabet



◆ Neka su:

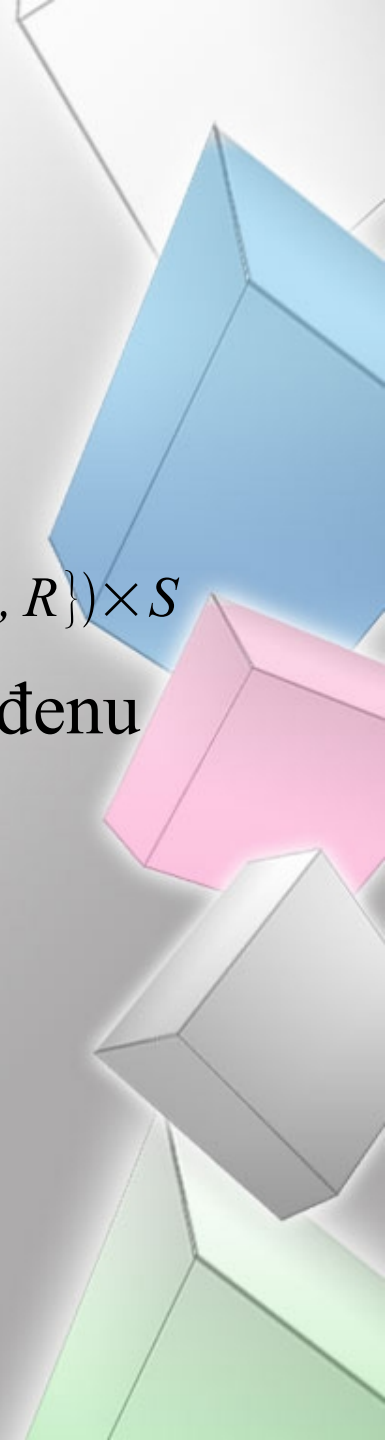
- q_0 početno stanje, $q_0 \in S$
- q_z je završno stanje, $q_z \in S$
- $A = \{1, 0\}$ alfabet
- δ preslikavanje, takvo da $\delta : (S \setminus \{q_z\}) \times A \rightarrow (A \cup \{L, R\}) \times S$



♦ Neka su:

- q_0 početno stanje, $q_0 \in S$
- q_z je završno stanje, $q_z \in S$
- $A = \{1, 0\}$ alfabet
- δ preslikavanje, takvo da $\delta : (S \setminus \{q_z\}) \times A \rightarrow (A \cup \{L, R\}) \times S$

♦ Tada *Tjuringovu mašinu* definišemo kao uređenu petorku (S, q_0, q_z, A, δ)



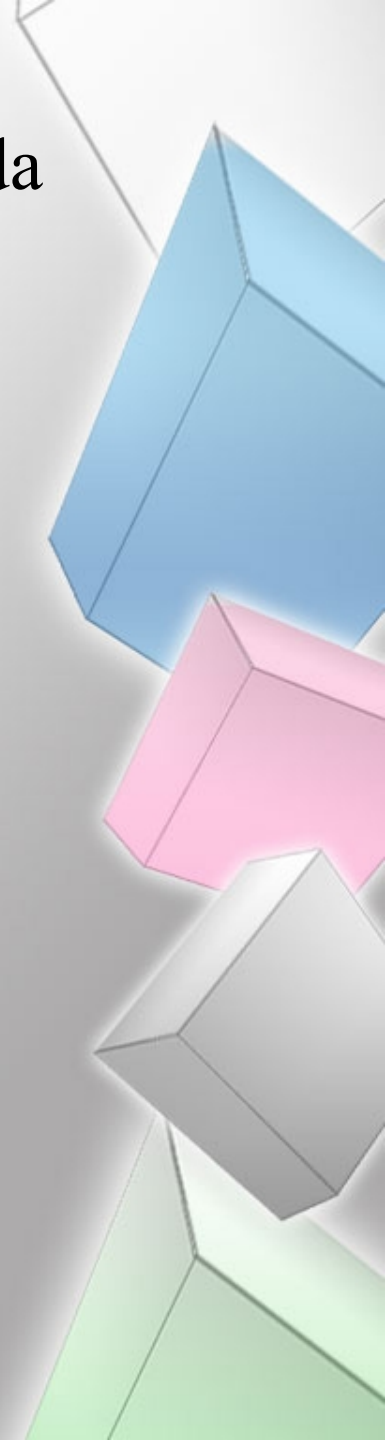
♦ Neka su:

- q_0 početno stanje, $q_0 \in S$
- q_z je završno stanje, $q_z \in S$
- $A = \{1, 0\}$ alfabet
- δ preslikavanje, takvo da $\delta : (S \setminus \{q_z\}) \times A \rightarrow (A \cup \{L, R\}) \times S$

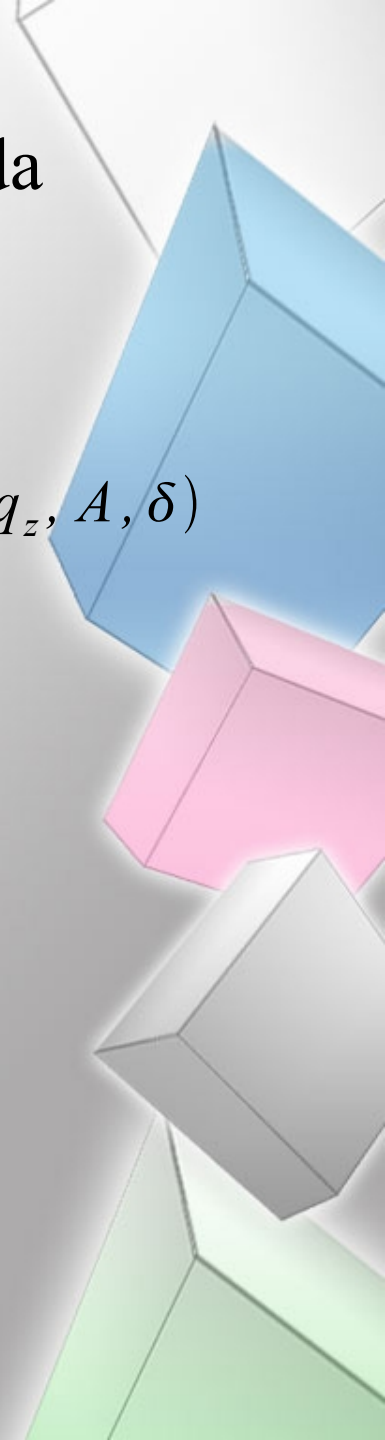
♦ Tada *Tjuringovu mašinu* definišemo kao uređenu petorku (S, q_0, q_z, A, δ)

♦ Čelija trake su numerisane celim brojevima, tako da je glava inicijalno postavljena na nultu, levo od nje su ćelije sa indeksima -1, -2, ..., a desno od nje sa 1, 2, ...

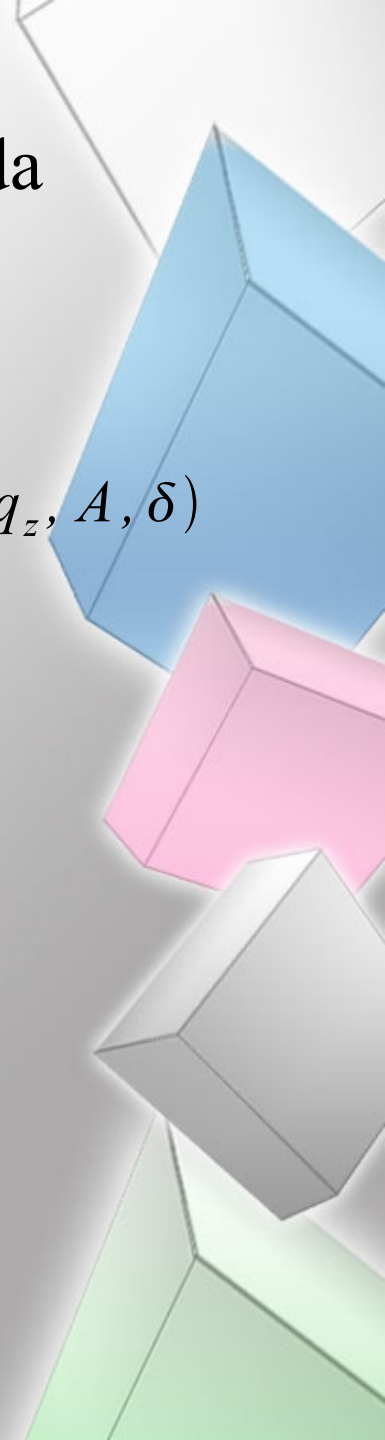
- ♦ *Opis trake* je preslikavanje $F: \mathbb{Z} \rightarrow A$, tako da ako je u ćeliji sa indeksom z upisan znak 1, odnosno 0, $F(z)=1, F(z)=0$



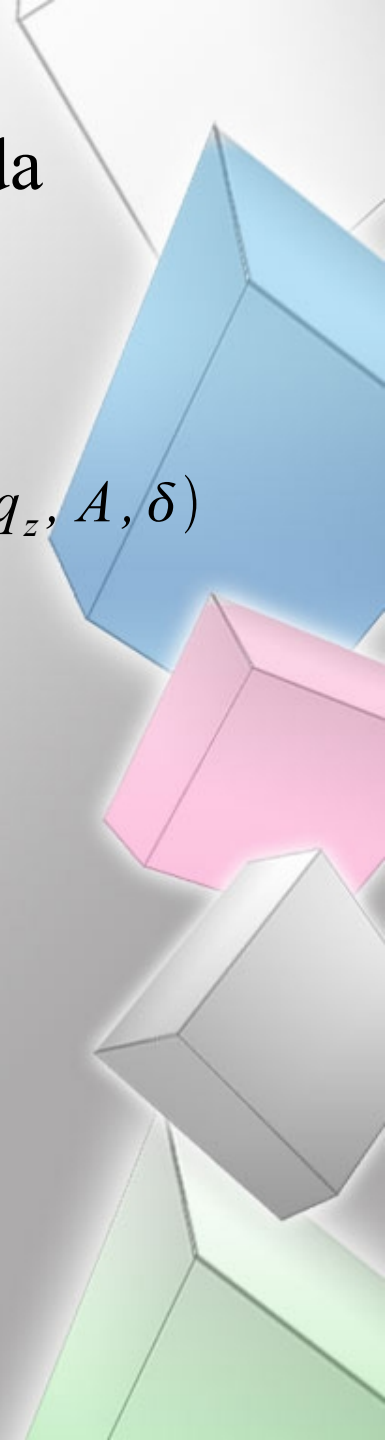
- ♦ *Opis trake* je preslikavanje $F: \mathbb{Z} \rightarrow A$, tako da ako je u ćeliji sa indeksom z upisan znak 1, odnosno 0, $F(z)=1, F(z)=0$
- ♦ *Konfiguracija* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ je trojka (F, q, e) gde su



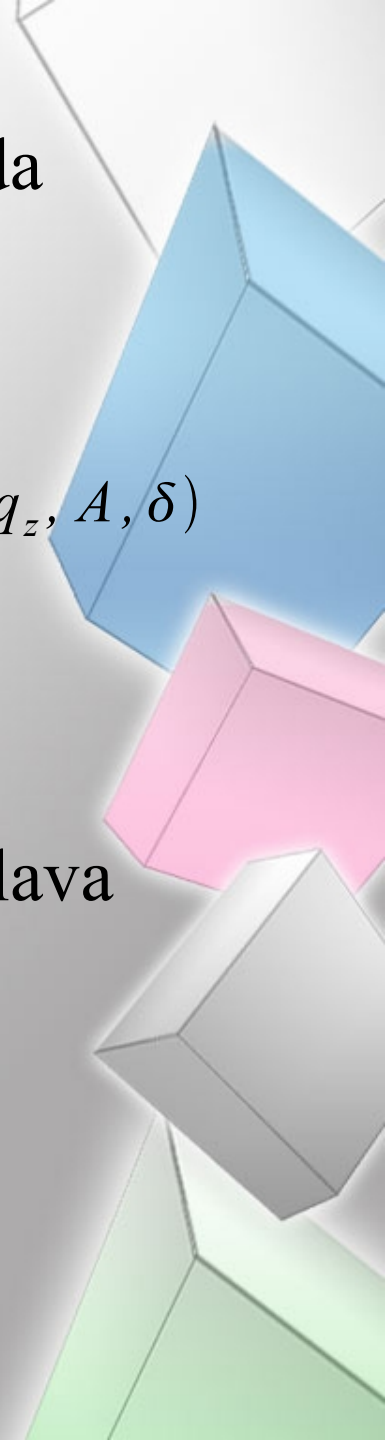
- ♦ *Opis trake* je preslikavanje $F: \mathbb{Z} \rightarrow A$, tako da ako je u ćeliji sa indeksom z upisan znak 1, odnosno 0, $F(z)=1, F(z)=0$
- ♦ *Konfiguracija* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ je trojka (F, q, e) gde su
 - F opis trake



- ♦ *Opis trake* je preslikavanje $F: \mathbb{Z} \rightarrow A$, tako da ako je u ćeliji sa indeksom z upisan znak 1, odnosno 0, $F(z)=1, F(z)=0$
- ♦ *Konfiguracija* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ je trojka (F, q, e) gde su
 - F opis trake
 - $q \in S$, tekuće stanje



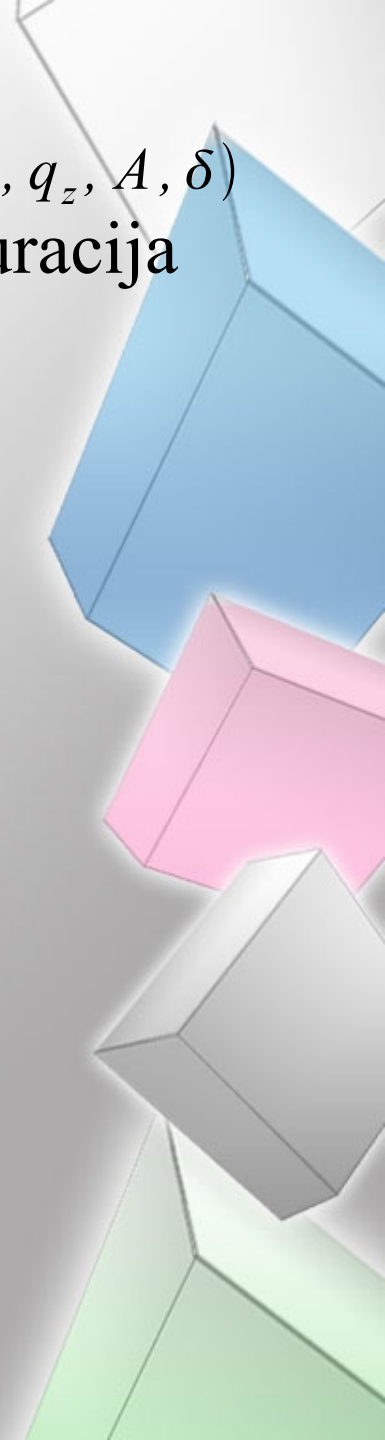
- ♦ *Opis trake* je preslikavanje $F : \mathbb{Z} \rightarrow A$, tako da ako je u ćeliji sa indeksom z upisan znak 1, odnosno 0, $F(z)=1, F(z)=0$
- ♦ *Konfiguracija* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ je trojka (F, q, e) gde su
 - F opis trake
 - $q \in S$, tekuće stanje
 - $e \in \mathbb{Z}$, broj ćelije nad kojom se nalazi glava



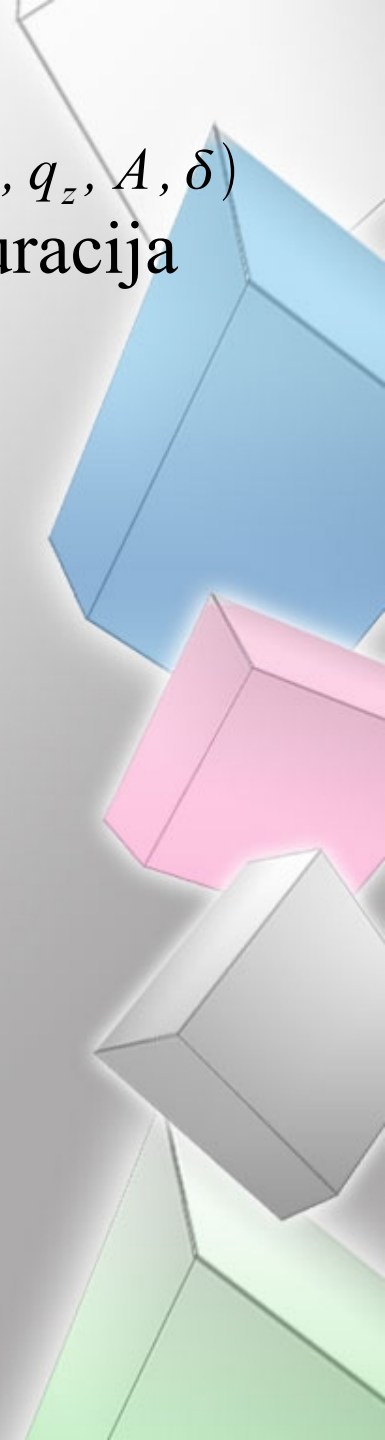
- ♦ *Opis trake* je preslikavanje $F: \mathbb{Z} \rightarrow A$, tako da ako je u ćeliji sa indeksom z upisan znak 1, odnosno 0, $F(z)=1, F(z)=0$
- ♦ *Konfiguracija* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ je trojka (F, q, e) gde su
 - F opis trake
 - $q \in S$, tekuće stanje
 - $e \in \mathbb{Z}$, broj ćelije nad kojom se nalazi glava
- ♦ *Naredba* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ je četvorka (q_i, s, o, q_j) gde je $\delta(q_i, s)=(o, q_j)$. Preslikavanje δ se naziva program.

- ♦ *Opis trake* je preslikavanje $F: \mathbb{Z} \rightarrow A$, tako da ako je u ćeliji sa indeksom z upisan znak 1, odnosno 0, $F(z)=1, F(z)=0$
- ♦ *Konfiguracija* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ je trojka (F, q, e) gde su
 - F opis trake
 - $q \in S$, tekuće stanje
 - $e \in \mathbb{Z}$, broj ćelije nad kojom se nalazi glava
- ♦ *Naredba* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ je četvorka (q_i, s, o, q_j) gde je $\delta(q_i, s)=(o, q_j)$. Preslikavanje δ se naziva program.
- ♦ Važi $q_i = q_i', s = s' \Rightarrow \delta(q_i, s) = \delta(q_i', s')$

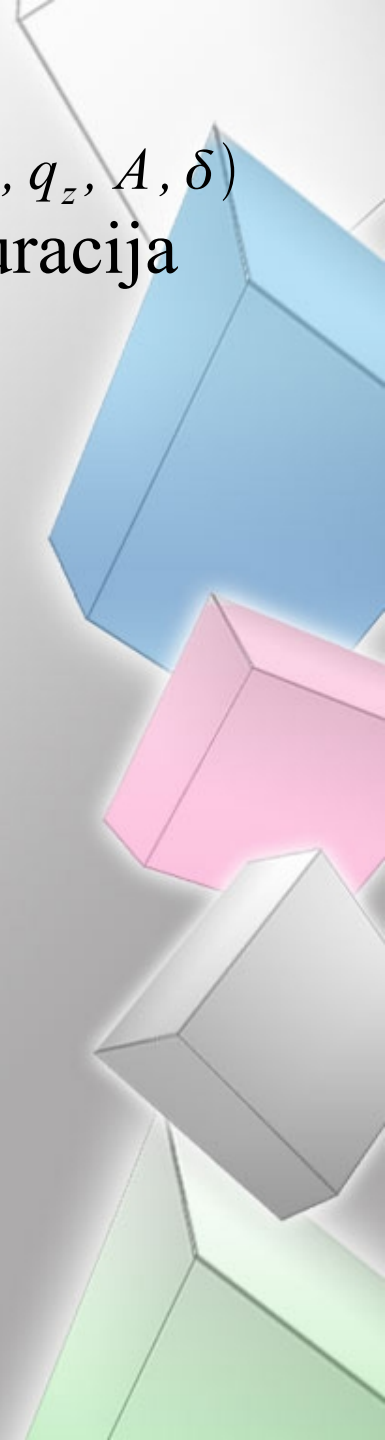
- ♦ *Računski korak* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ za naredbu $I=(q_i, s, o, q_j)$ je svaki par konfiguracija $((F', q', e'), (F'', q'', e''))$ za koje važi



- ♦ *Računski korak* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ za naredbu $I=(q_i, s, o, q_j)$ je svaki par konfiguracija $((F', q', e'), (F'', q'', e''))$ za koje važi
 - $q_i = q'$

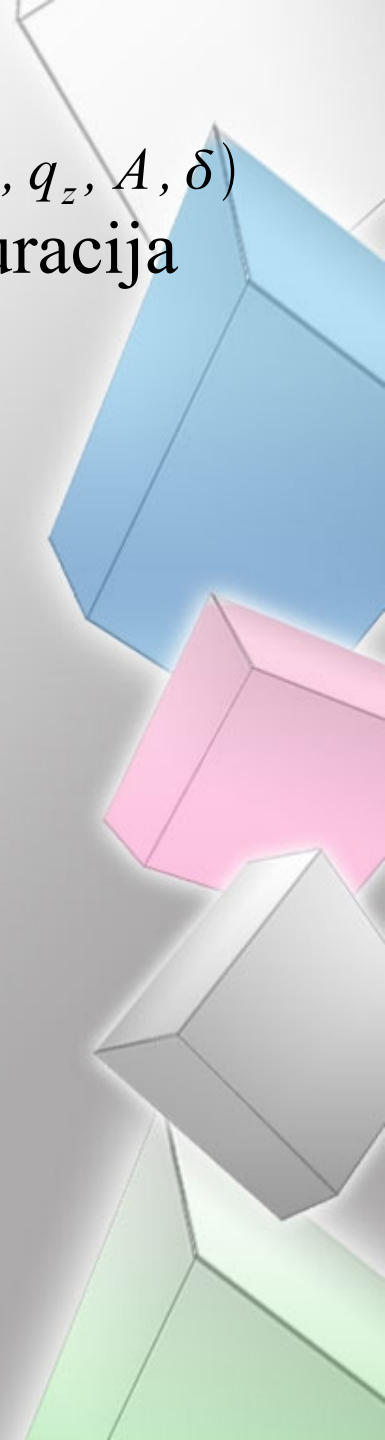


- ♦ *Računski korak* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ za naredbu $I=(q_i, s, o, q_j)$ je svaki par konfiguracija $((F', q', e'), (F'', q'', e''))$ za koje važi
 - $q_i = q'$
 - $F'(e') = s$



♦ *Računski korak* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ za naredbu $I=(q_i, s, o, q_j)$ je svaki par konfiguracija $((F', q', e'), (F'', q'', e''))$ za koje važi

- $q_i = q'$
- $F'(e') = s$
- $q_j = q''$



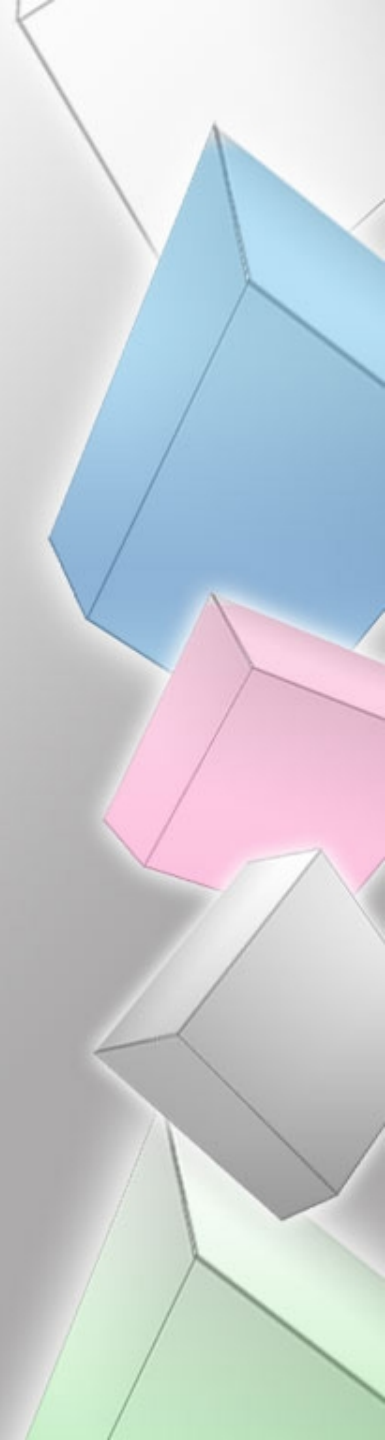
♦ *Računski korak* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ za naredbu $I=(q_i, s, o, q_j)$ je svaki par konfiguracija $((F', q', e'), (F'', q'', e''))$ za koje važi

- $q_i = q'$
- $F'(e') = s$
- $q_j = q''$
- Ako je operacija $o = 1$, važi
 - $e' = e''$
 - F' i F'' se poklapaju, osim u e'' , gde $F''(e'') = 1$

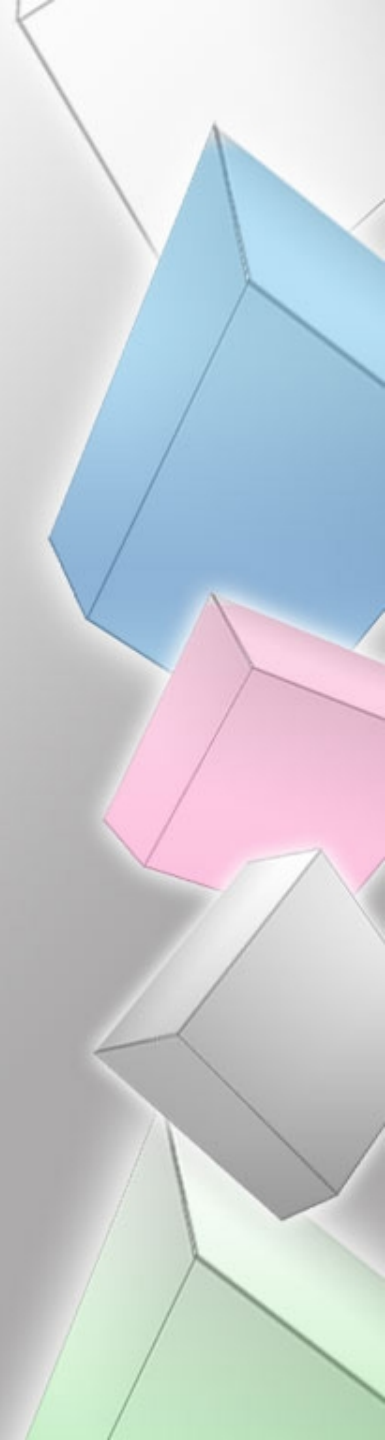
♦ *Računski korak* Tjuringove mašine $M=(S, q_0, q_z, A, \delta)$ za naredbu $I=(q_i, s, o, q_j)$ je svaki par konfiguracija $((F', q', e'), (F'', q'', e''))$ za koje važi

- $q_i = q'$
- $F'(e') = s$
- $q_j = q''$
- Ako je operacija $o=1$, važi
 - $e' = e''$
 - F' i F'' se poklapaju, osim u e'' , gde $F''(e'')=1$
- Ako je operacija $o=0$, važi
 - $e' = e''$
 - F' i F'' se poklapaju, osim u e'' , gde $F''(e'')=0$

- Ako je operacija $o=L$, važi
 - $e' - 1 = e''$
 - F' i F'' se poklapaju



- Ako je operacija $o = L$, važi
 - $e' - 1 = e''$
 - F' i F'' se poklapaju
- Ako je operacija $o = R$, važi
 - $e' + 1 = e''$
 - F' i F'' se poklapaju

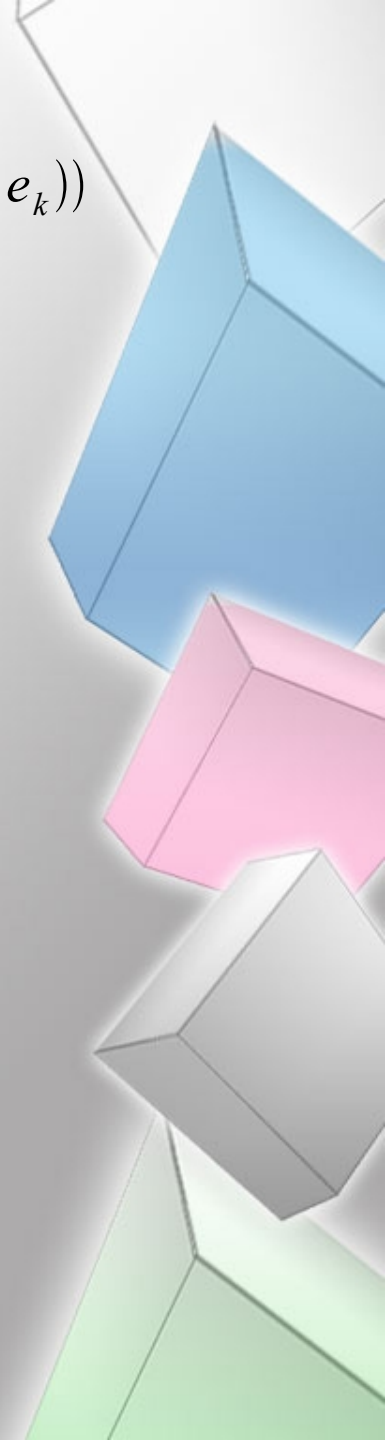


- Ako je operacija $o = L$, važi
 - $e' - 1 = e''$
 - F' i F'' se poklapaju
- Ako je operacija $o = R$, važi
 - $e' + 1 = e''$
 - F' i F'' se poklapaju
- ♦ Da je $((F', q', e'), (F'', q'', e''))$ računski korak naredbe I , označavamo sa $(F', q', e') \vdash_I (F'', q'', e'')$

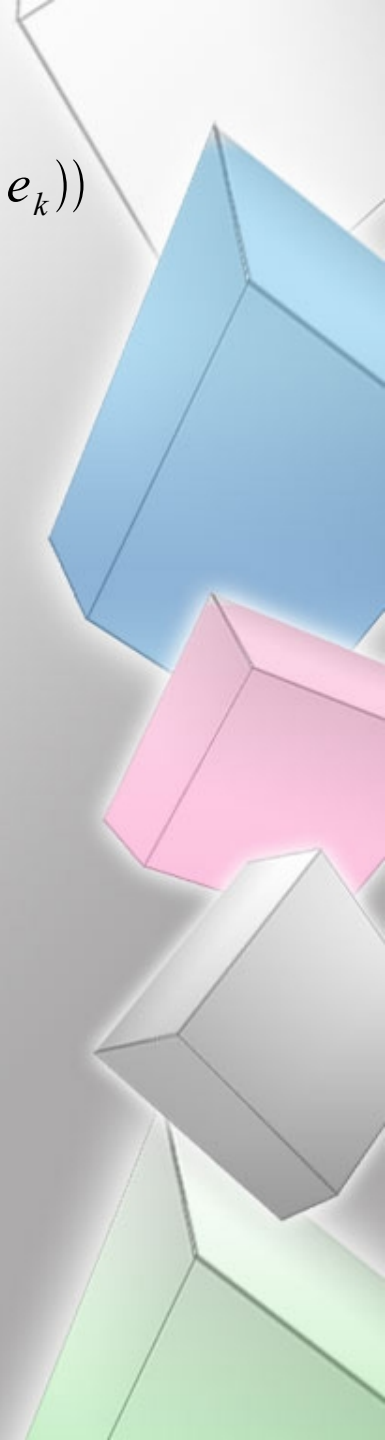
- Ako je operacija $o = L$, važi
 - $e' - 1 = e''$
 - F' i F'' se poklapaju
- Ako je operacija $o = R$, važi
 - $e' + 1 = e''$
 - F' i F'' se poklapaju
- ♦ Da je $((F', q', e'), (F'', q'', e''))$ računski korak naredbe I , označavamo sa $(F', q', e') \vdash_I (F'', q'', e'')$
- ♦ *Izračunavanje* za Tjuringovu mašinu $M = (S, q_0, q_z, A, \delta)$ i program P opisan f-jom δ je niz konfiguracija $(F_0, q^0, e_0), (F_1, q^1, e_1), \dots, (F_m, q^m, e_m)$ tako da

- Ako je operacija $o = L$, važi
 - $e' - 1 = e''$
 - F' i F'' se poklapaju
- Ako je operacija $o = R$, važi
 - $e' + 1 = e''$
 - F' i F'' se poklapaju
- ♦ Da je $((F', q', e'), (F'', q'', e''))$ računski korak naredbe I , označavamo sa $(F', q', e') \vdash_I (F'', q'', e'')$
- ♦ *Izračunavanje* za Tjuringovu mašinu $M = (S, q_0, q_z, A, \delta)$ i program P opisan f-jom δ je niz konfiguracija $(F_0, q^0, e_0), (F_1, q^1, e_1), \dots, (F_m, q^m, e_m)$ tako da
 - $(F_0, q^0, e_0) \vdash_{I_1} (F_1, q^1, e_1)$, za neku naredbu I_1 (programa P) i stanje $q^0 = q_0$

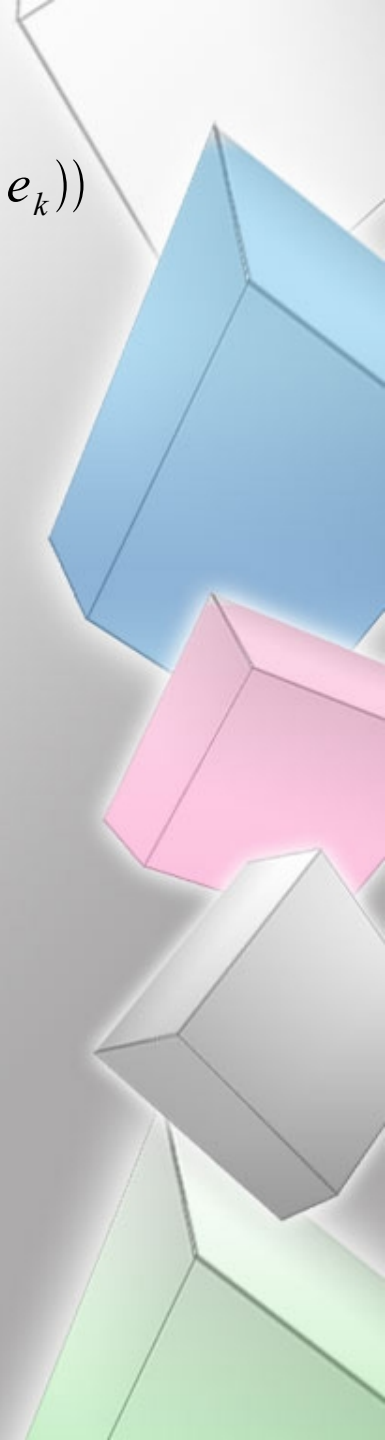
- $(\forall k \in \mathbb{N})(0 < k < m-1)((F_{k-1}, q^{k-1}, e_{k-1}) \vdash_{I_k} (F_k, q^k, e_k))$



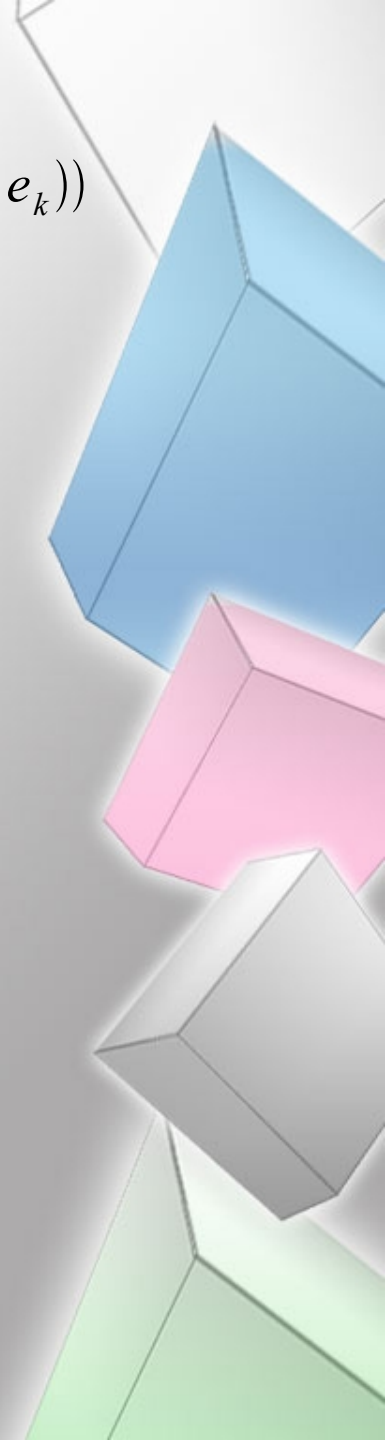
- $(\forall k \in \mathbb{N})(0 < k < m-1)((F_{k-1}, q^{k-1}, e_{k-1})|_{I_k}(F_k, q^k, e_k))$
- $(F_{m-1}, q^{m-1}, e_{m-1})|_{I_m}(F_m, q^m, e_m)$, za $q^m = q_m$



- $(\forall k \in \mathbb{N})(0 < k < m-1)((F_{k-1}, q^{k-1}, e_{k-1})|_{-I_k}(F_k, q^k, e_k))$
- $(F_{m-1}, q^{m-1}, e_{m-1})|_{-I_m}(F_m, q^m, e_m), \text{ za } q^m = q_m$
- $(\forall k = \overline{1, m-1}) q^k \neq q_z$



- $(\forall k \in \mathbb{N})(0 < k < m-1)((F_{k-1}, q^{k-1}, e_{k-1})|_{I_k}(F_k, q^k, e_k))$
- $(F_{m-1}, q^{m-1}, e_{m-1})|_{I_m}(F_m, q^m, e_m), \text{ za } q^m = q_m$
- $(\forall k = \overline{1, m-1}) q^k \neq q_z$
- $e_0 = 0, (\forall i < 0)(F_0(i) = 0)$

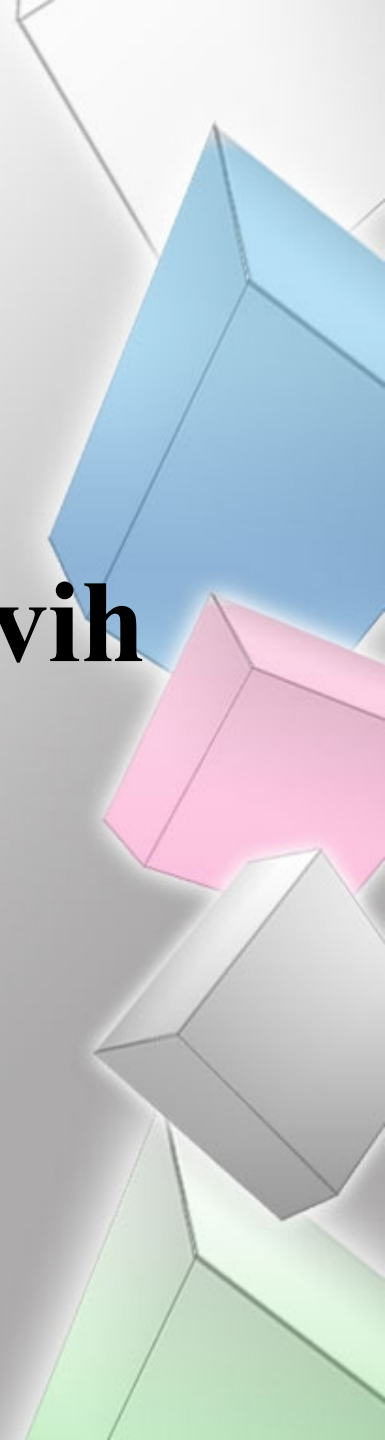


- $(\forall k \in \mathbb{N})(0 < k < m-1)((F_{k-1}, q^{k-1}, e_{k-1})|_{-I_k}(F_k, q^k, e_k))$
- $(F_{m-1}, q^{m-1}, e_{m-1})|_{-I_m}(F_m, q^m, e_m)$, za $q^m = q_m$
- $(\forall k = \overline{1, m-1}) q^k \neq q_z$
- $e_0 = 0, (\forall i < 0)(F_0(i) = 0)$
- $(\exists n > 0)(\exists k_1, \dots, k_n) e_0 \leq k_1 < k_2 < \dots < k_n$, tako da važi
 - $(\forall i, e_0 \leq i \leq k_1) F_0(i) = 1, F_0(k_1 + 1) = 0$
 - $(\forall i, k_1 + 1 \leq i \leq k_2) F_0(i) = 1, F_0(k_2 + 1) = 0$
 - ...
 - $(\forall i, k_{n-1} + 1 \leq i \leq k_n) F_0(i) = 1$
 - $(\forall i, k_n < i) F_0(i) = 0$

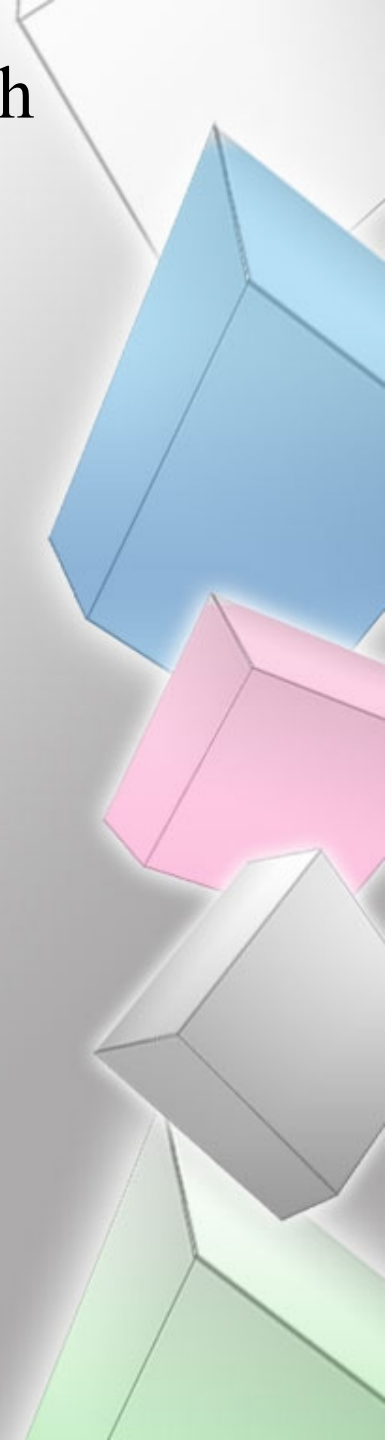
- $(\forall k \in \mathbb{N})(0 < k < m-1)((F_{k-1}, q^{k-1}, e_{k-1})|_{-I_k}(F_k, q^k, e_k))$
- $(F_{m-1}, q^{m-1}, e_{m-1})|_{-I_m}(F_m, q^m, e_m)$, za $q^m = q_m$
- $(\forall k = \overline{1, m-1}) q^k \neq q_z$
- $e_0 = 0, (\forall i < 0)(F_0(i) = 0)$
- $(\exists n > 0)(\exists k_1, \dots, k_n) e_0 \leq k_1 < k_2 < \dots < k_n$, tako da važi
 - $(\forall i, e_0 \leq i \leq k_1) F_0(i) = 1, F_0(k_1 + 1) = 0$
 - $(\forall i, k_1 + 1 \leq i \leq k_2) F_0(i) = 1, F_0(k_2 + 1) = 0$
 - ...
 - $(\forall i, k_{n-1} + 1 \leq i \leq k_n) F_0(i) = 1$
 - $(\forall i, k_n < i) F_0(i) = 0$
- $F_m(e_m) = 1$, i $(\forall i < e_m) F_m(i) = 0$

- $(\forall k \in \mathbb{N})(0 < k < m-1)((F_{k-1}, q^{k-1}, e_{k-1})|_{-I_k}(F_k, q^k, e_k))$
- $(F_{m-1}, q^{m-1}, e_{m-1})|_{-I_m}(F_m, q^m, e_m)$, za $q^m = q_m$
- $(\forall k = \overline{1, m-1}) q^k \neq q_z$
- $e_0 = 0, (\forall i < 0)(F_0(i) = 0)$
- $(\exists n > 0)(\exists k_1, \dots, k_n) e_0 \leq k_1 < k_2 < \dots < k_n$, tako da važi
 - $(\forall i, e_0 \leq i \leq k_1) F_0(i) = 1, F_0(k_1 + 1) = 0$
 - $(\forall i, k_1 + 1 \leq i \leq k_2) F_0(i) = 1, F_0(k_2 + 1) = 0$
 - ...
 - $(\forall i, k_{n-1} + 1 \leq i \leq k_n) F_0(i) = 1$
 - $(\forall i, k_n < i) F_0(i) = 0$
- $F_m(e_m) = 1$, i $(\forall i < e_m) F_m(i) = 0$
- $\exists k \geq e_m$, tako da
 - $(\forall i, e_m \leq i \leq k) F_m(i) = 1$
 - $(\forall j > k) F_m(j) = 0$

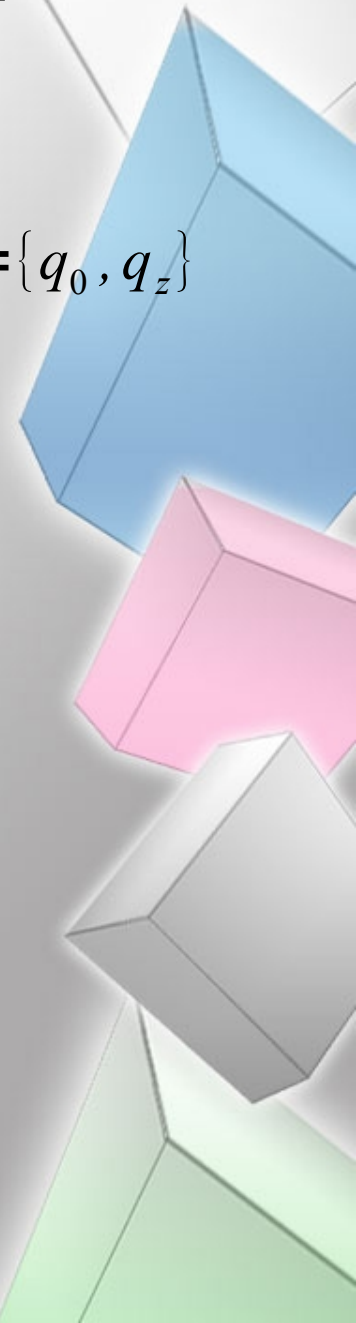
Kombinovanje Tjuringovih Mašina



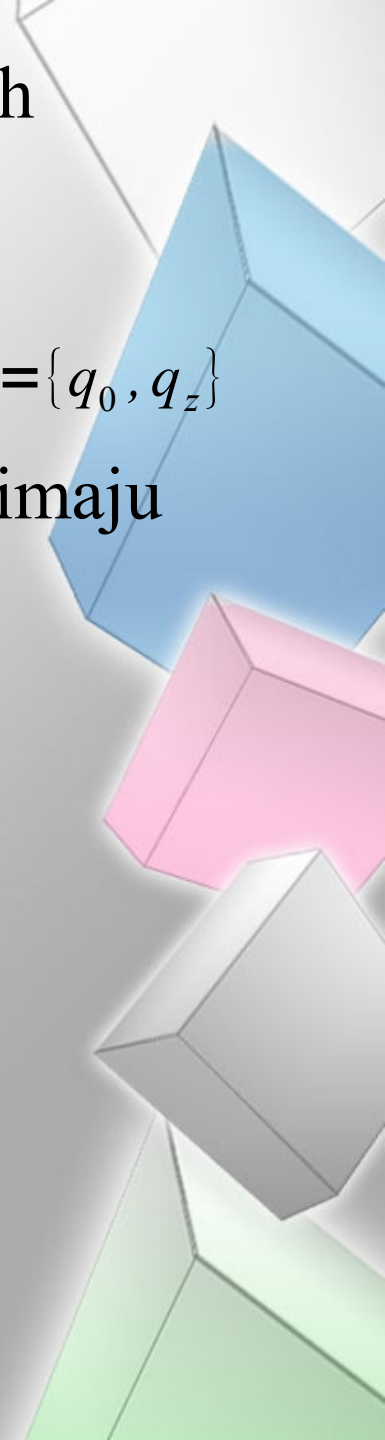
- ♦ Nalik funkcijama – modularnost programskih jezika



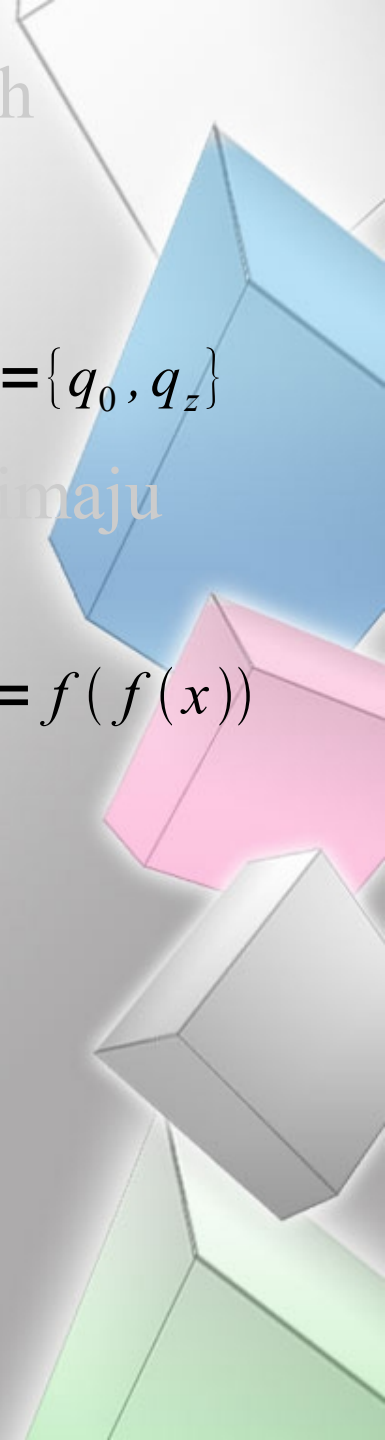
- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$



- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja



- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ★ Neka je $f(x) = x + 1$, $g(x) = x + 2$, tj. $g(x) = (f \circ f)(x) = f(f(x))$



- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ★ Neka je $f(x) = x + 1$, $g(x) = x + 2$, tj. $g(x) = (f \circ f)(x) = f(f(x))$

$$f(x) = x + 1$$

$$q_0 \vdash q_0$$

$$q_0 \vdash q_z$$

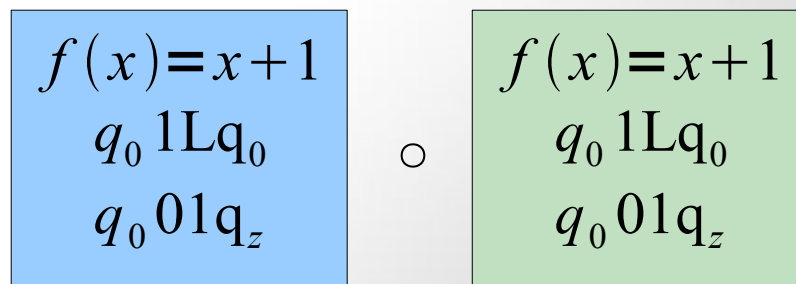
- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ★ Neka je $f(x) = x + 1$, $g(x) = x + 2$, tj. $g(x) = (f \circ f)(x) = f(f(x))$

$f(x) = x + 1$
 $q_0 \rightarrow q_0$
 $q_0 \rightarrow q_z$

○

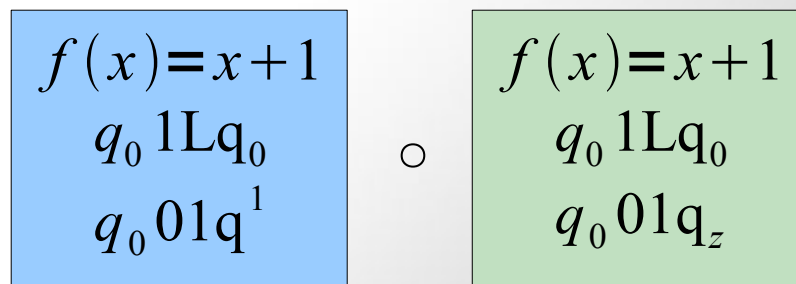
$f(x) = x + 1$
 $q_0 \rightarrow q_0$
 $q_0 \rightarrow q_z$

- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ★ Neka je $f(x) = x + 1$, $g(x) = x + 2$, tj. $g(x) = (f \circ f)(x) = f(f(x))$



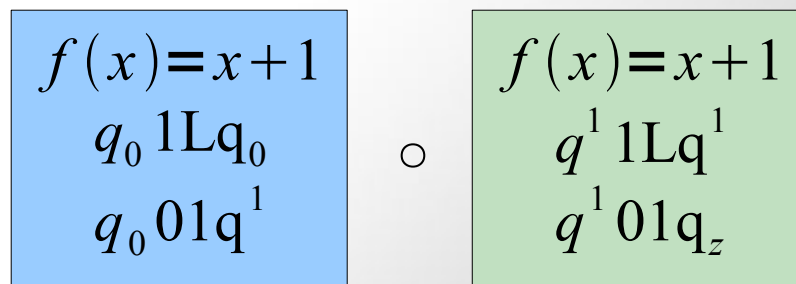
$$q^1 \notin S(\text{blue}) \cup S(\text{green})$$

- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ★ Neka je $f(x) = x + 1$, $g(x) = x + 2$, tj. $g(x) = (f \circ f)(x) = f(f(x))$



$$q^1 \notin S(\text{blue}) \cup S(\text{green})$$

- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ★ Neka je $f(x) = x + 1$, $g(x) = x + 2$, tj. $g(x) = (f \circ f)(x) = f(f(x))$



$$q^1 \notin S(\text{blue}) \cup S(\text{green})$$

- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ★ Neka je $f(x) = x + 1$, $g(x) = x + 2$, tj. $g(x) = (f \circ f)(x) = f(f(x))$

$$\begin{array}{|c|} \hline f(x) = x + 1 \\ q_0 1Lq_0 \\ q_0 01q^1 \\ \hline \end{array} \quad \circ \quad \begin{array}{|c|} \hline f(x) = x + 1 \\ q^1 1Lq^1 \\ q^1 01q_z \\ \hline \end{array} = \begin{array}{|c|} \hline g(x) = x + 2 \\ q_0 1Lq_0 \\ q_0 01q^1 \\ q^1 1Lq^1 \\ q^1 01q_z \\ \hline \end{array}$$

$q^1 \notin S(\text{blue box}) \cup S(\text{green box})$

- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ★ Neka je $f(x) = x + 1$, $g(x) = x + 2$, tj. $g(x) = (f \circ f)(x) = f(f(x))$

$$\begin{array}{l} f(x) = x + 1 \\ q_0 \text{ 1L} q_0 \\ q_0 \text{ 01} q^1 \end{array}$$

○

$$\begin{array}{l} f(x) = x + 1 \\ q^1 \text{ 1L} q^1 \\ q^1 \text{ 01} q_z \end{array}$$

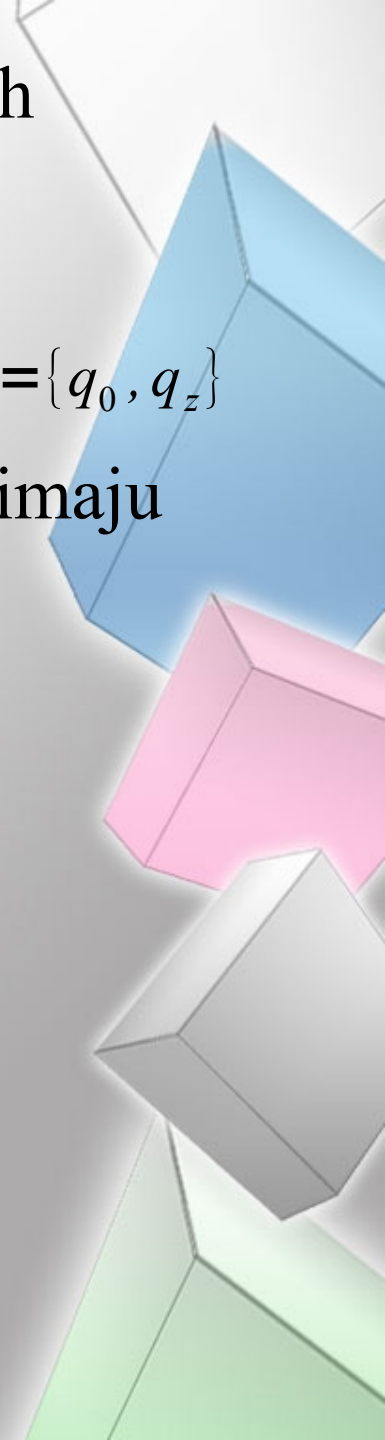
=

$$\begin{array}{l} g(x) = x + 2 \\ q_0 \text{ 1L} q_0 \\ q_0 \text{ 01} q^1 \\ q^1 \text{ 1L} q^1 \\ q^1 \text{ 01} q_z \end{array}$$

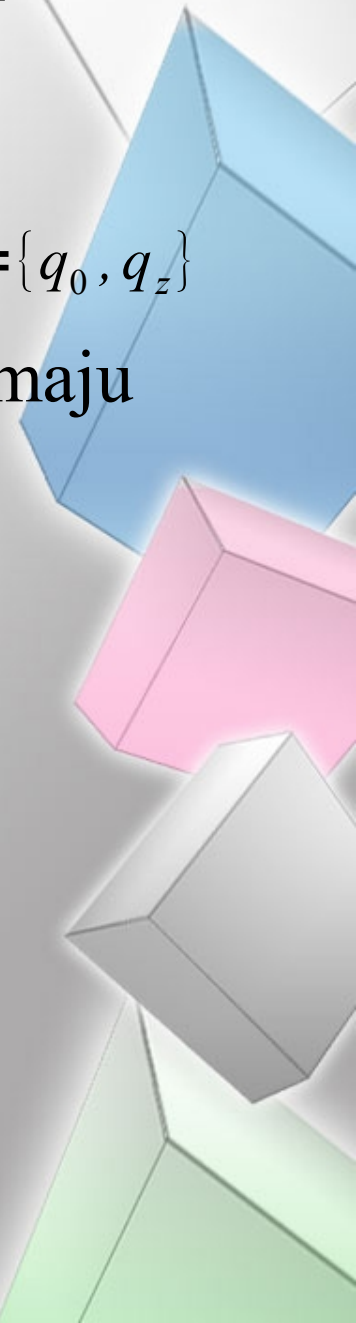
$$q^1 \notin S(\text{blue}) \cup S(\text{green})$$

 nazivamo *kompozicijom*

- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja

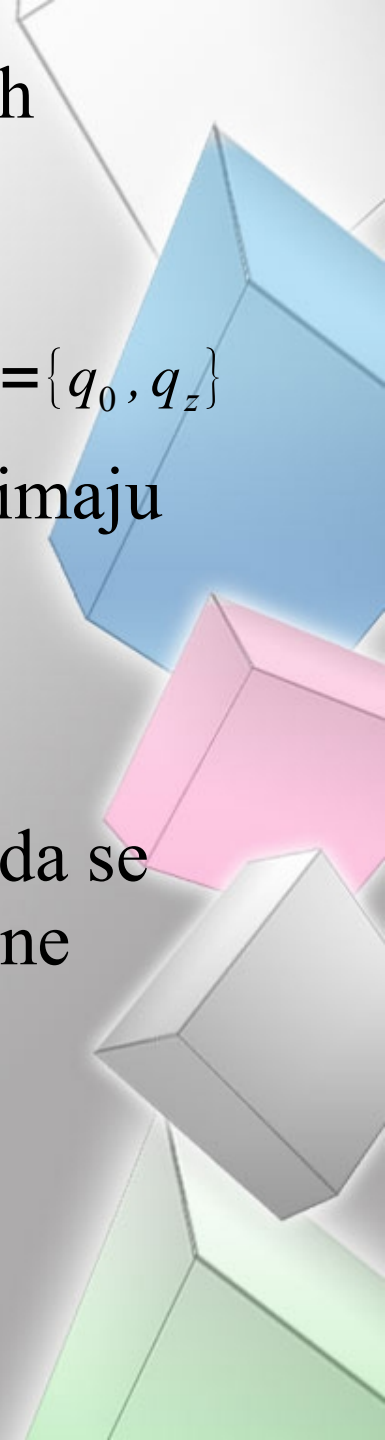


- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Turingove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ♦ Više T mašina možemo iskombinovati i na nešto složeniji način



- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Turingove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ♦ Više T mašina možemo iskombinovati i na nešto složeniji način

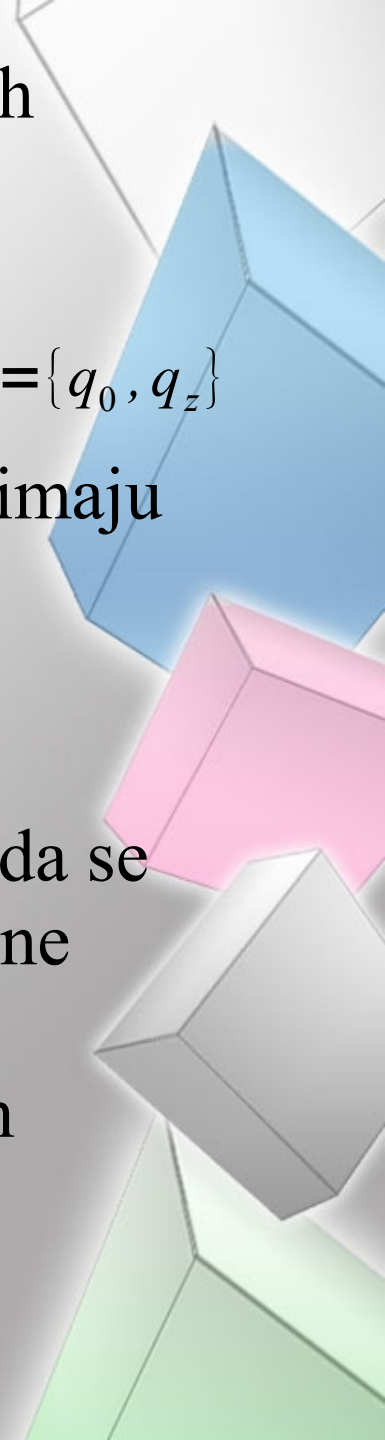
Pri tome, ne uzimamo u obzir “konvenciju” da se glava na početku i na kraju izvršavanja mašine mora nalaziti na prvoj jedinici s' leva



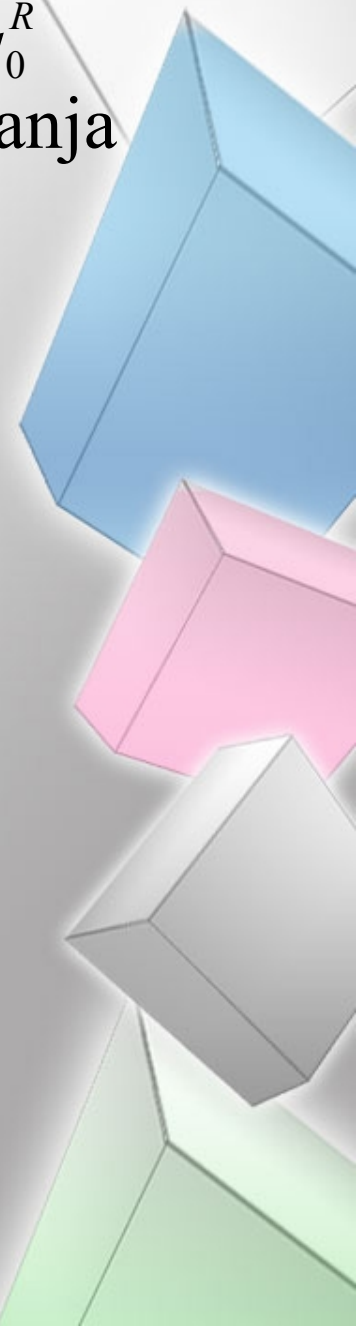
- ♦ Nalik funkcijama – modularnost programskih jezika
- ♦ Dve Tjuringove mašine (programa) imaju disjunktne skupove stanja akko $S(P_1) \cap S(P_2) = \{q_0, q_z\}$
- ♦ Pri kombinovanju obezbediti da sve mašine imaju disjunktne skupove stanja
- ♦ Više T mašina možemo iskombinovati i na nešto složeniji način

Pri tome, ne uzimamo u obzir “konvenciju” da se glava na početku i na kraju izvršavanja mašine mora nalaziti na prvoj jedinici s' leva

Podseća na naredbu grananja u programskim jezicima.

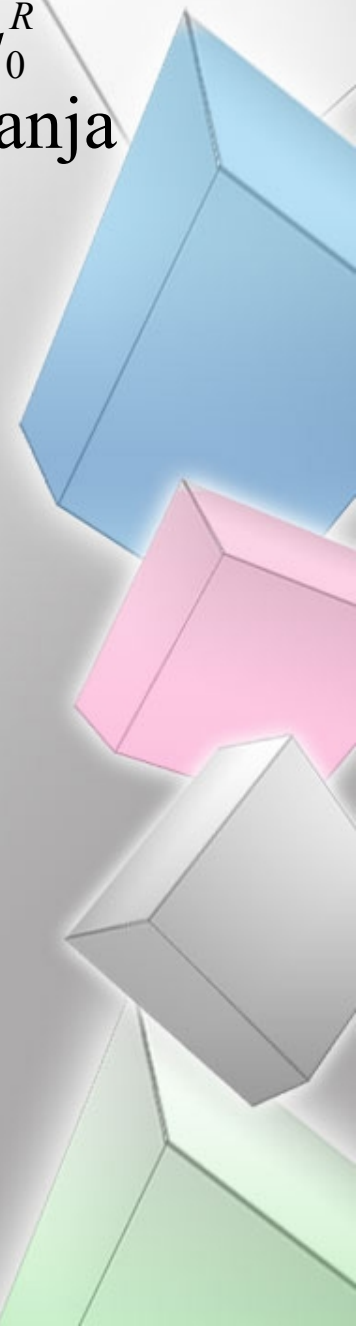


- ♦ Neka su P , Q i R Tjuringove mašine, takve da $S(P) \cap S(Q) \cap S(R) = \{q_0, q_z\}$ i neka su q_z^P, q_0^Q, q_0^R redom – završno stanje mašine P i početna stanja mašina Q i R .



- ♦ Neka su P , Q i R Tjuringove mašine, takve da $S(P) \cap S(Q) \cap S(R) = \{q_0, q_z\}$ i neka su q_z^P, q_0^Q, q_0^R redom – završno stanje mašine P i početna stanja mašina Q i R .

Tada, možemo iskombinovati ove mašine jednostavnim nadovezivanjem uz nešto što liči na pitalice u programskim jezicima



- ♦ Neka su P , Q i R Tjuringove mašine, takve da $S(P) \cap S(Q) \cap S(R) = \{q_0, q_z\}$ i neka su q_z^P, q_0^Q, q_0^R redom – završno stanje mašine P i početna stanja mašina Q i R .

Tada, možemo iskombinovati ove mašine jednostavnim nadovezivanjem uz nešto što liči na pitalice u programskim jezicima

```
If ( !(p = P()) )  
    Q();  
else if ( p == 1 )  
    R();
```

~

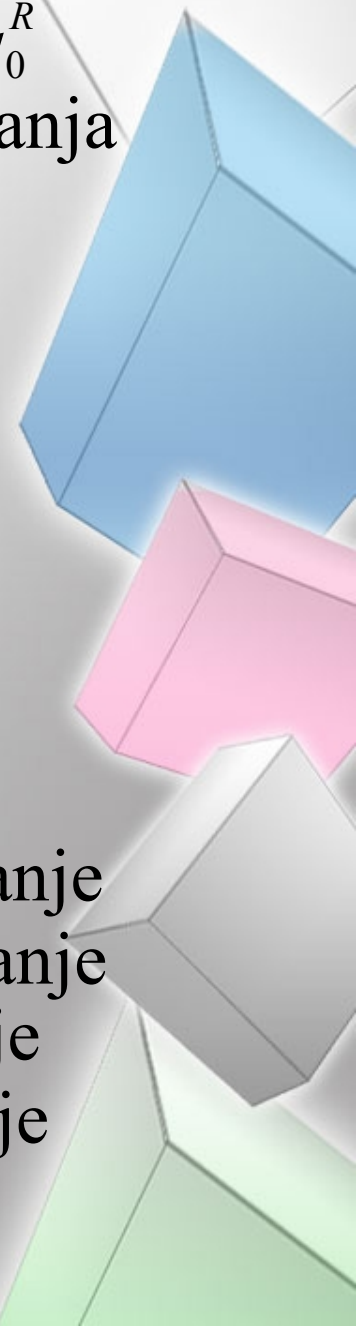
P
q_z^P 00 q_0^Q
q_z^P 11 q_0^R
Q
R

- ♦ Neka su P , Q i R Tjuringove mašine, takve da $S(P) \cap S(Q) \cap S(R) = \{q_0, q_z\}$ i neka su q_z^P, q_0^Q, q_0^R redom – završno stanje mašine P i početna stanja mašina Q i R .

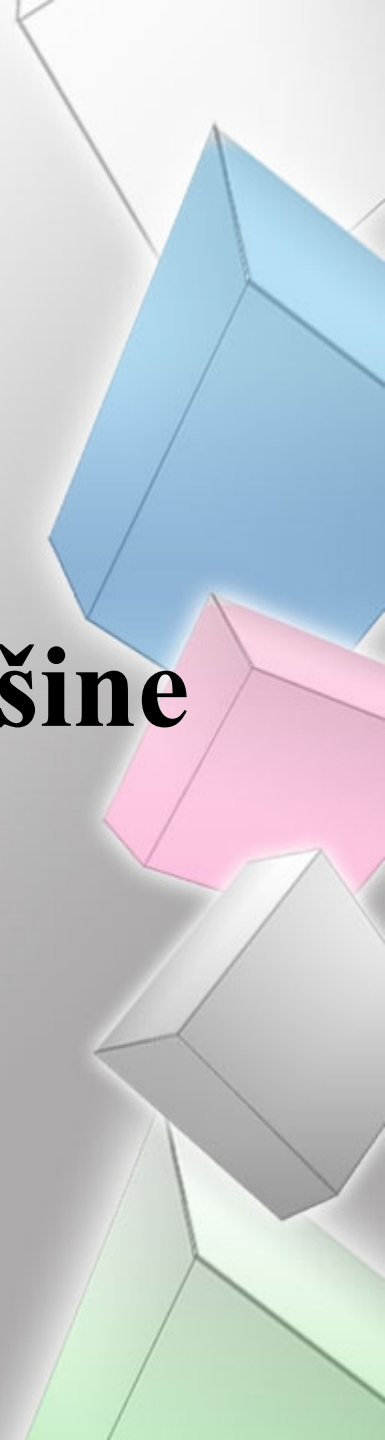
Tada, možemo iskombinovati ove mašine jednostavnim nadovezivanjem uz nešto što liči na pitalice u programskim jezicima

Kompozicijom i grananjem četiri mašine - ekvivalenata osnovnim naredbama - možemo dobiti proizvoljnu T mašinu

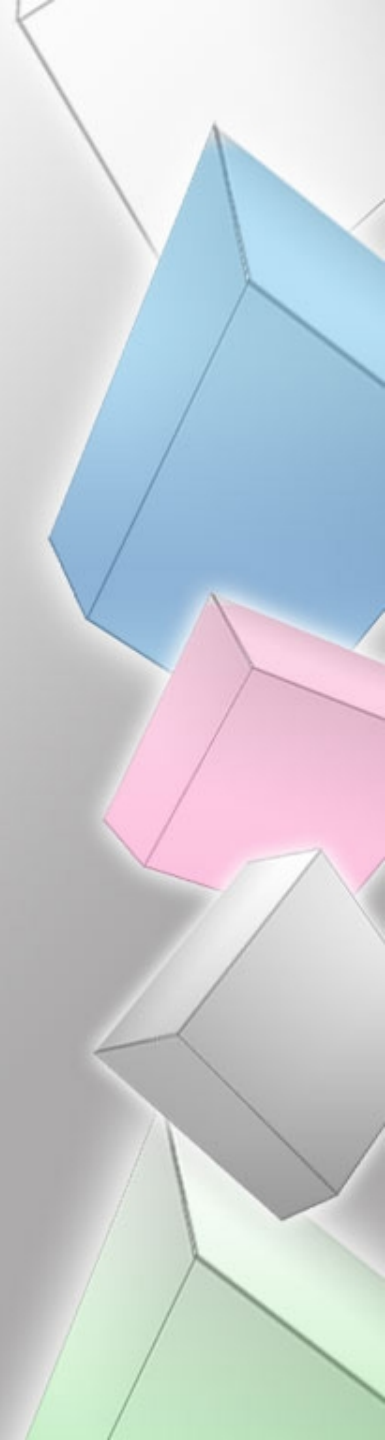
- Upis jedinice u tekuću ćeliju i zaustavljanje
- Upis nule u tekuću ćeliju i zaustavljanje
- Pomeraj glave ulevo za 1 i zaustavljanje
- Pomeraj glave udesno za 1 i zaustavljanje



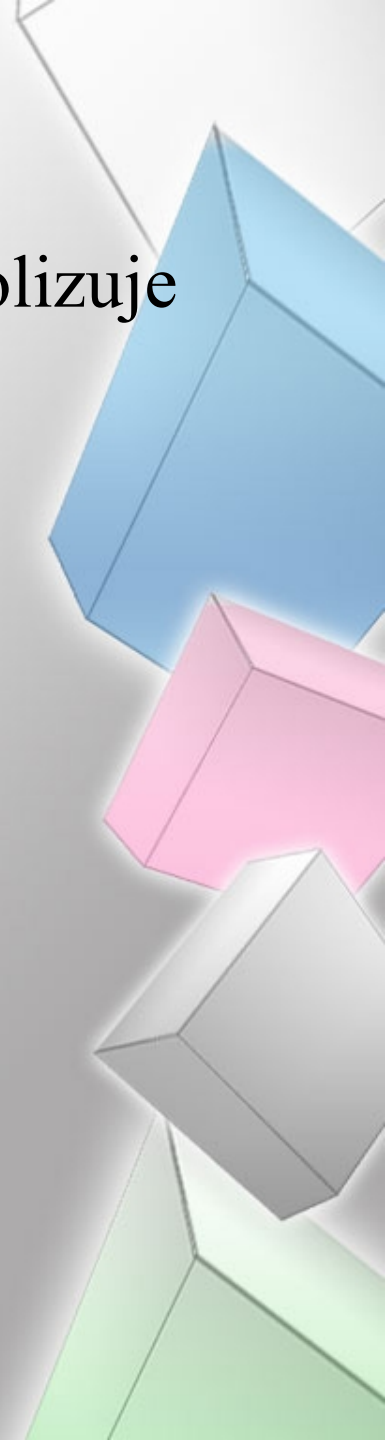
Varijante Tjuringove Mašine



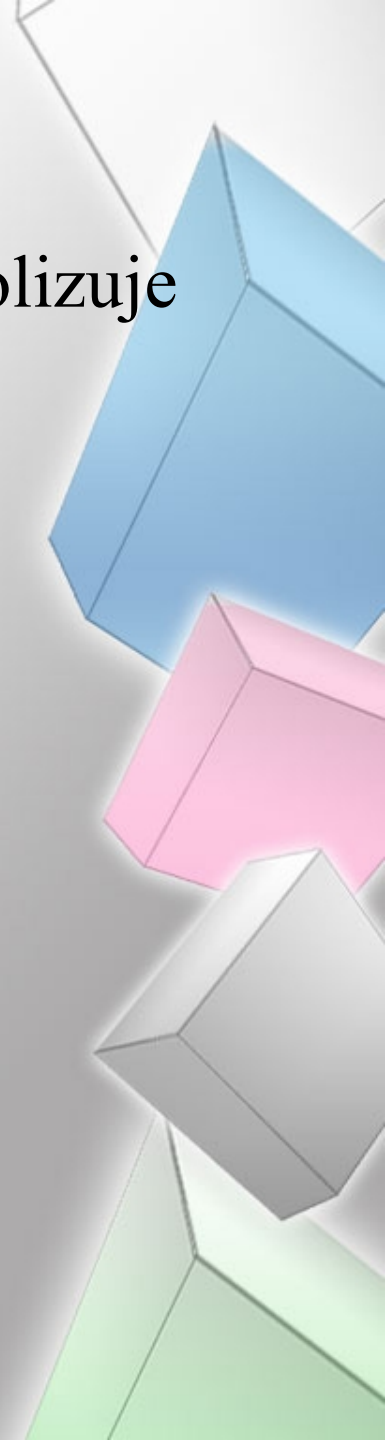
- ♦ TM sa bogatijim alfabetom



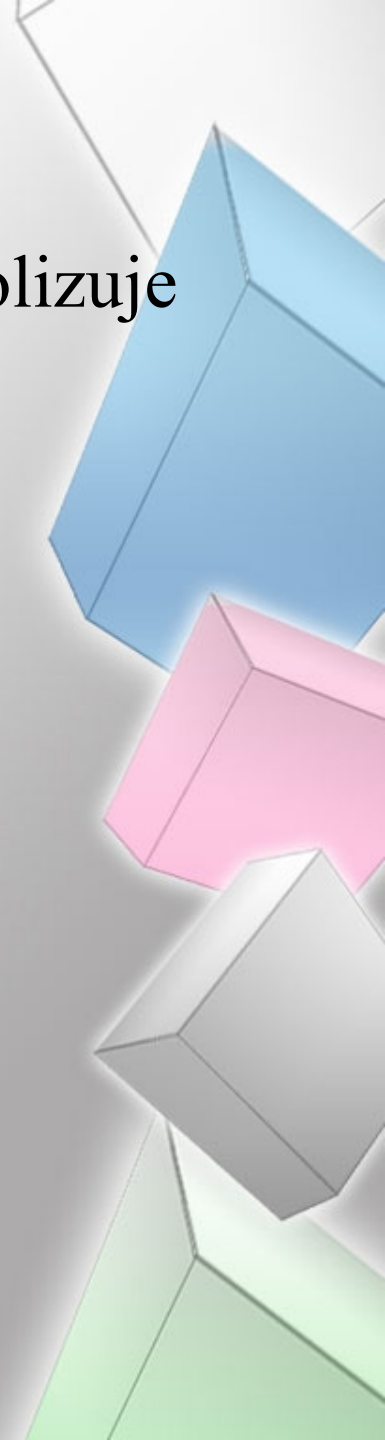
- ♦ TM sa bogatijim alfabetom
- ♦ TM sa više završnih stanja (gde svako simbolizuje neki rezultat rešavanja problema)



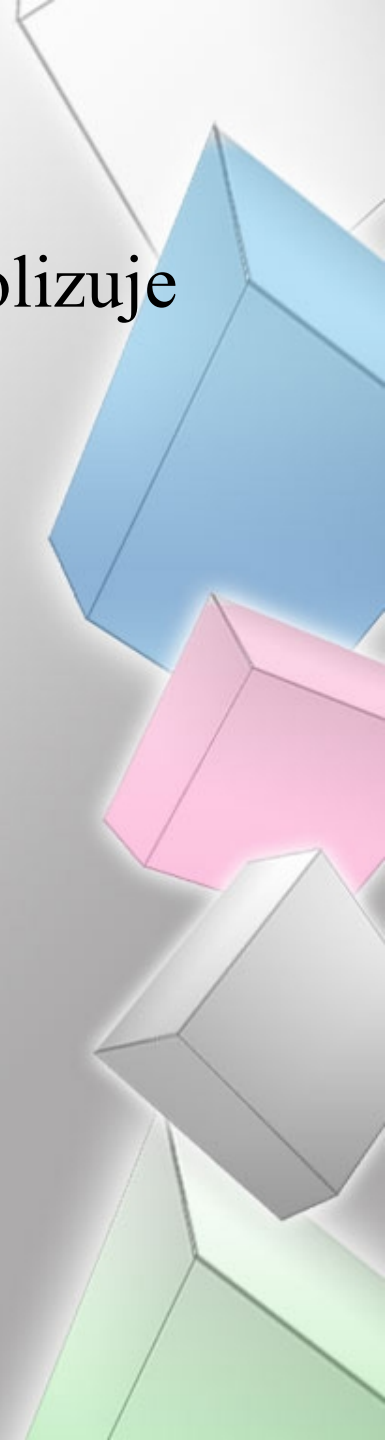
- ♦ TM sa bogatijim alfabetom
- ♦ TM sa više završnih stanja (gde svako simbolizuje neki rezultat rešavanja problema)
- ♦ TM sa levo ograničenom trakom



- ♦ TM sa bogatijim alfabetom
- ♦ TM sa više završnih stanja (gde svako simbolizuje neki rezultat rešavanja problema)
- ♦ TM sa levo ograničenom trakom
- ♦ TM sa više traka i glavom za svaku



- ♦ TM sa bogatijim alfabetom
- ♦ TM sa više završnih stanja (gde svako simbolizuje neki rezultat rešavanja problema)
- ♦ TM sa levo ograničenom trakom
- ♦ TM sa više traka i glavom za svaku
- ♦ TM sa jednom trakom i više glava nad njom



- ♦ TM sa bogatijim alfabetom
- ♦ TM sa više završnih stanja (gde svako simbolizuje neki rezultat rešavanja problema)
- ♦ TM sa levo ograničenom trakom
- ♦ TM sa više traka i glavom za svaku
- ♦ TM sa jednom trakom i više glava nad njom
- ♦ TM sa 2D trakom beskonačnoj po obe dimenzije



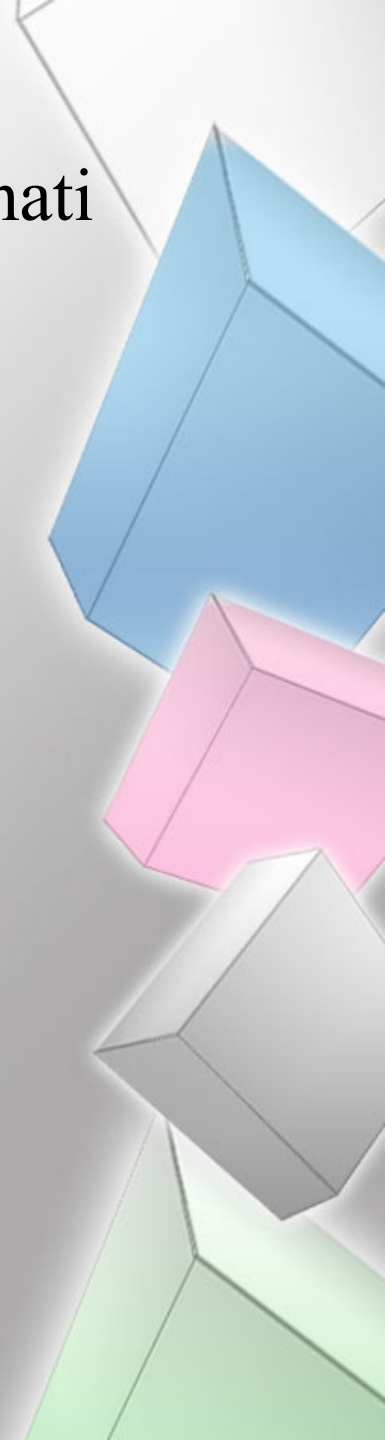
- ♦ TM sa bogatijim alfabetom
- ♦ TM sa više završnih stanja (gde svako simbolizuje neki rezultat rešavanja problema)
- ♦ TM sa levo ograničenom trakom
- ♦ TM sa više traka i glavom za svaku
- ♦ TM sa jednom trakom i više glava nad njom
- ♦ TM sa 2D trakom beskonačnoj po obe dimenzije
- ♦ TM koje dozvoljavaju u istoj naredbi upis u ćeliju i pomeraj glave.



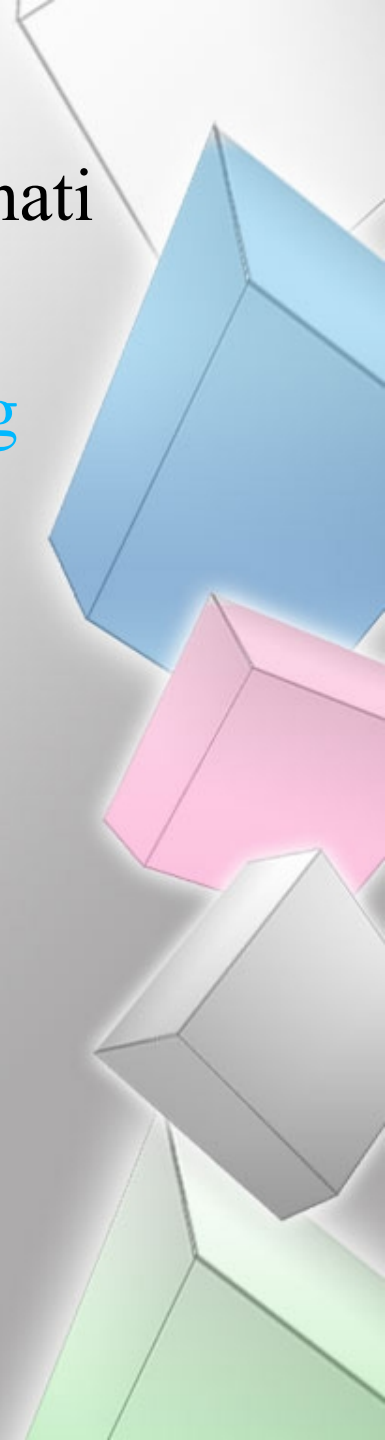
- ♦ TM sa bogatijim alfabetom
- ♦ TM sa više završnih stanja (gde svako simbolizuje neki rezultat rešavanja problema)
- ♦ TM sa levo ograničenom trakom
- ♦ TM sa više traka i glavom za svaku
- ♦ TM sa jednom trakom i više glava nad njom
- ♦ TM sa 2D trakom beskonačnoj po obe dimenzije
- ♦ TM koje dozvoljavaju u istoj naredbi upis u ćeliju i pomeraj glave.
- ♦ Nedeterminističke Tjuringove mašine (δ je relacija a ne funkcija, tj. može postojati za isto stanje i sadržaj ćelije, više različitih naredbi)



- ♦ Ni jedna od ovih varijanti ne proširuje klasu Turing izračunljivih funkcija (mogu izračunati samo ono što može i osnovna verzija TM-a)



- ♦ Ni jedna od ovih varijanti ne proširuje klasu Turing izračunljivih funkcija (mogu izračunati samo ono što može i osnovna verzija TM-a)
- ▶ Izračunljiva f-ja je samo ona koja je Turing izračunljiva

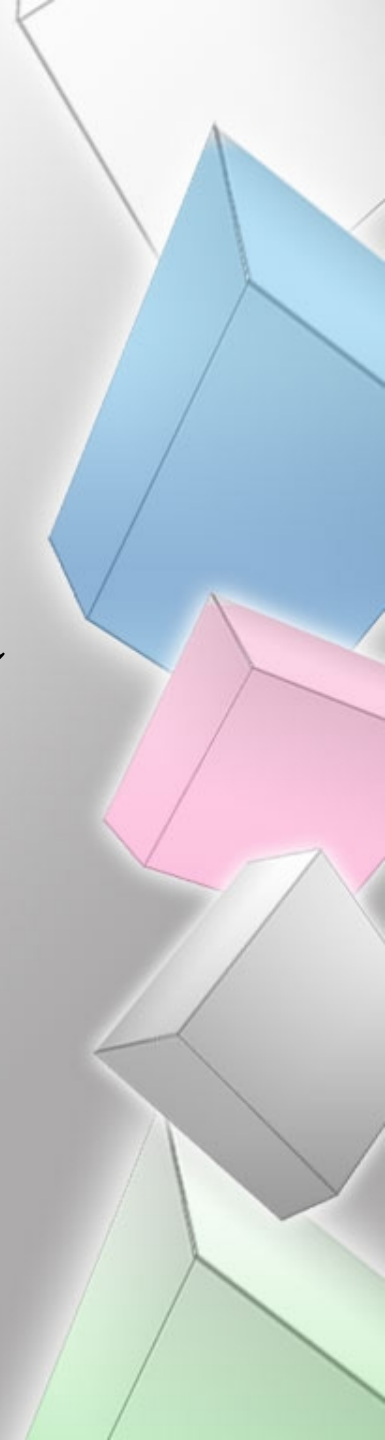


- ♦ Ni jedna od ovih varijanti ne proširuje klasu Turing izračunljivih funkcija (mogu izračunati samo ono što može i osnovna verzija TM-a)
- ▶ Izračunljiva f-ja je samo ona koja je Turing izračunljiva
- ♦ Neke od ovih varijanti ne mogu izračunati sve iz TI klase (npr. mašina sa trakom ograničenom sa jedne strane i unarnim alfabetom, ili mašina sa dva stanja i binarnim alfabetom)



- ♦ Ni jedna od ovih varijanti ne proširuje klasu Turing izračunljivih funkcija (mogu izračunati samo ono što može i osnovna verzija TM-a)
- ▶ Izračunljiva f-ja je samo ona koja je Turing izračunljiva
- ♦ Neke od ovih varijanti ne mogu izračunati sve iz TI klase (npr. mašina sa trakom ograničenom sa jedne strane i unarnim alfabetom, ili mašina sa dva stanja i binarnim alfabetom)
- ♦ Jednakost varijanti T mašina dokazuje se tako što se pokaže da za svaki program jedne od varijanti T mašina postoje programi za druge varijante koji simuliraju njegov rad (izračunavaju isti f-ju)

Tjuringova Mašina sa Bogatijim Alfabetom



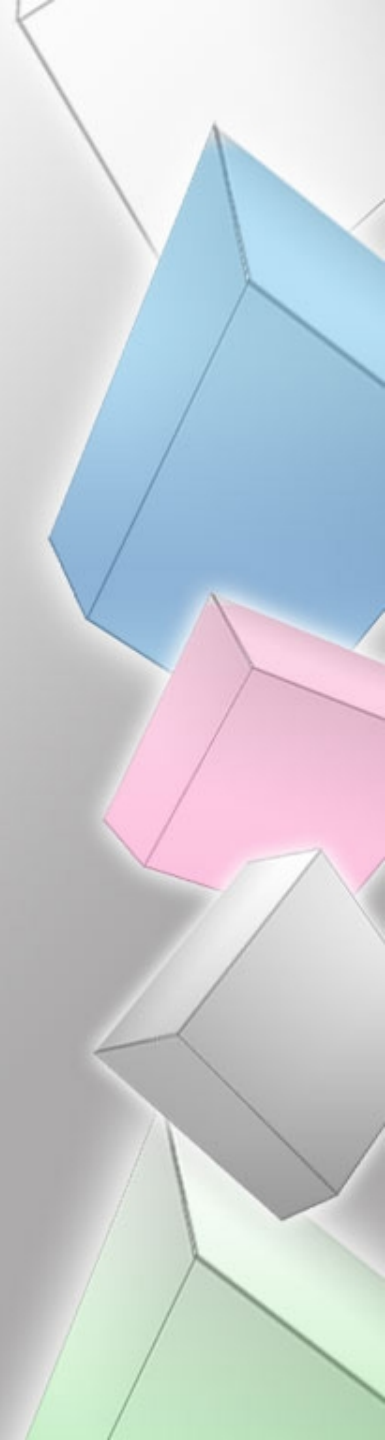
- ♦ Reči bilo kog alfabeta mogu se predstaviti unarnim alfabetom
- ♦ Obično je, što se izračunljivosti tiče, svejedno koji alfabet koristimo (kod mašina sa ni sa jedne strane ograničenom trakom...)
- ♦ Često je broj koraka potrebnih za rešavanje nekog problema manji ukoliko je alfabet veći (npr. što je veća osnova sistema, to je manje ćelija potrebno da se broj zapiše)



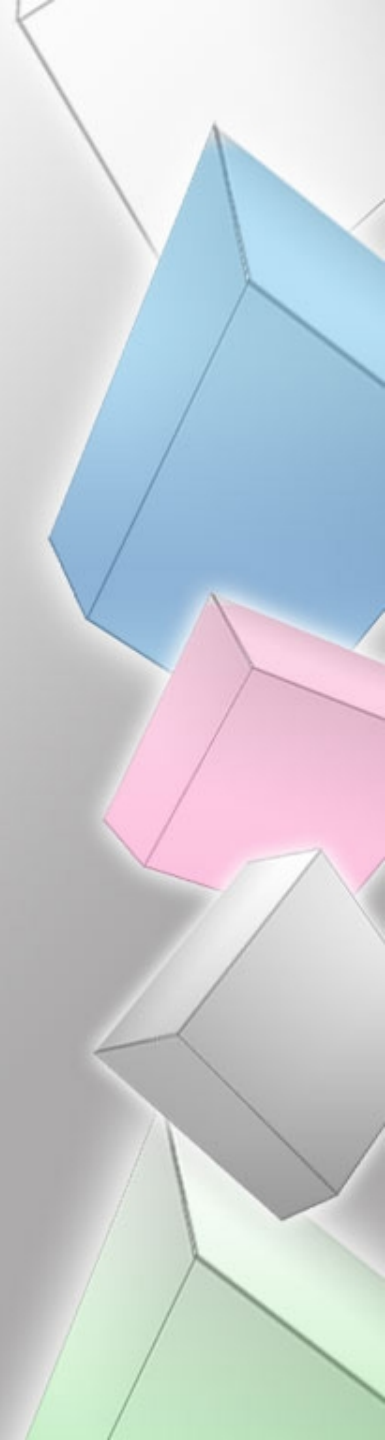
**Tjuringova Mašina sa Levo
Ograničenom Trakom
(sa Trakom sa Početkom na
Levoj Strani)**



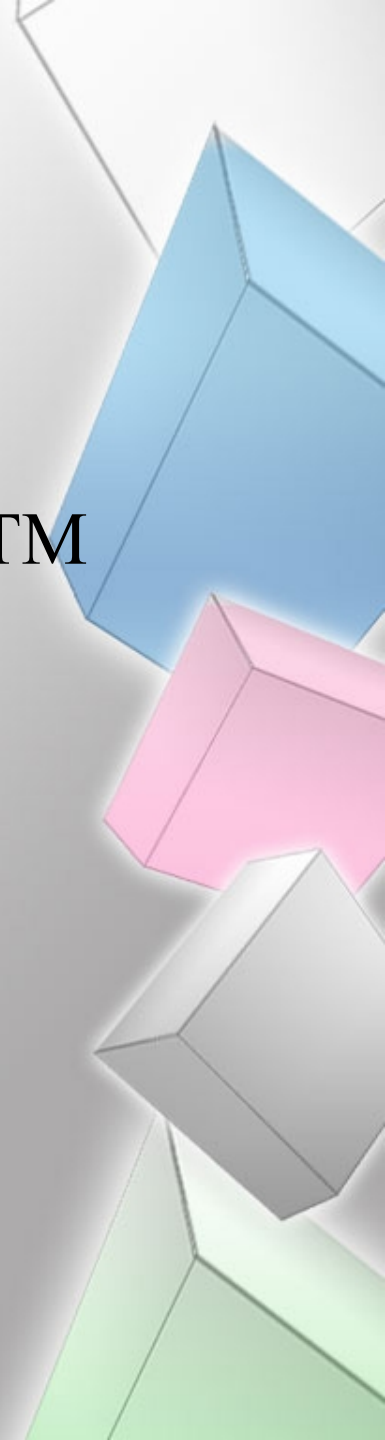
- ♦ U krajnje levoj ćeliji (početku trake) sadrži specijalni znak, npr. →



- ♦ U krajnje levoj ćeliji (početku trake) sadrži specijalni znak, npr. $\rightarrow$
- ♦ Jedina moguća naredba kada je glava iznad krajnje leve ćelije je $q_i \rightarrow R q_j$



- ♦ U krajnje levoj ćeliji (početku trake) sadrži specijalni znak, npr. $\rightarrow$
- ♦ Jedina moguća naredba kada je glava iznad krajnje leve ćelije je $q_i \rightarrow R q_j$
- ♦ Ova varijanta izračunava one f-je koje i std TM (dakle sve izračunljive)

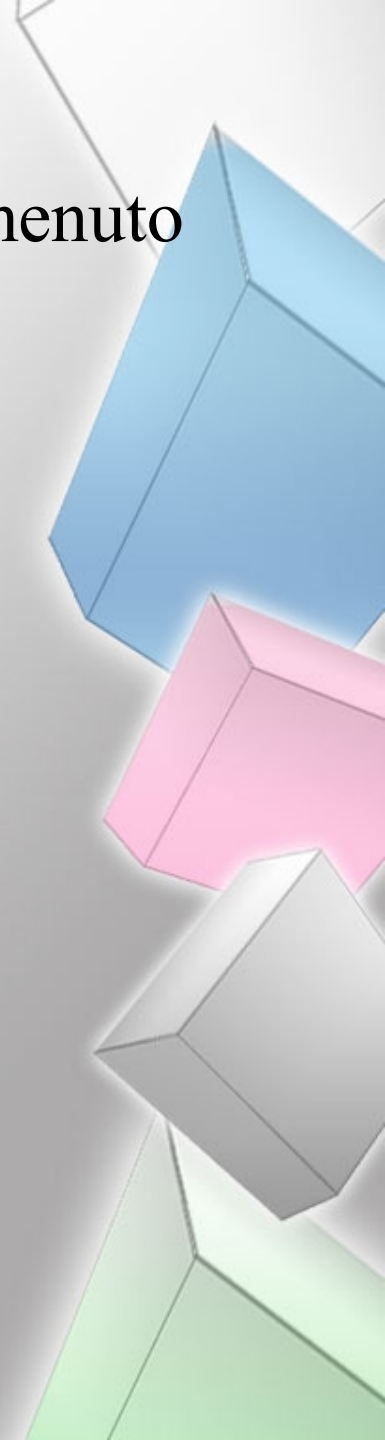


- ♦ U krajnje levoj ćeliji (početku trake) sadrži specijalni znak, npr. $\rightarrow$
- ♦ Jedina moguća naredba kada je glava iznad krajnje leve ćelije je $q_i \rightarrow R q_j$
- ♦ Ova varijanta izračunava one f-je koje i std TM (dakle sve izračunljive)

- Dokaz:

Da je klasa f-ja koju izračunavaju Tjuringove mašine sa levo zatvorenom trakom (LTM) potklasa u klasi Tjuring izračunljivih f-ja – očigledno je (u std verziji jednostavno nećemo koristiti ćelije levo od početka u LTM verziji)

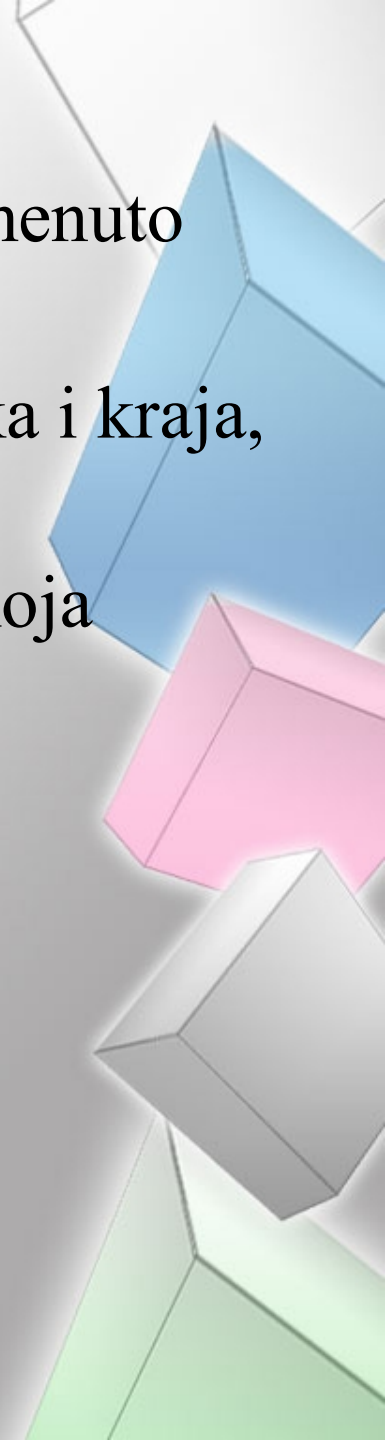
Za suprotan smer, moramo primeniti pomenuto pravilo o dokazivanju jednakosti TM-a



Za suprotan smer, moramo primeniti pomenuto pravilo o dokazivanju jednakosti TM-a

Neka je M Tjuringova mašina bez početka i kraja, i neka M izračunava f -ju f

Potrebno je pokazati da postoji LTM N koja izračunava f



Za suprotan smer, moramo primeniti pomenuto pravilo o dokazivanju jednakosti TM-a

Neka je M Tjuringova mašina bez početka i kraja, i neka M izračunava f -ju f

Potrebno je pokazati da postoji LTM N koja izračunava f

Traka mašine $N \rightarrow \begin{array}{cccc} +0 & +1 & +2 & \dots \\ -1 & -2 & -3 & \dots \end{array}$

Za suprotan smer, moramo primeniti pomenuto pravilo o dokazivanju jednakosti TM-a

Neka je M Tjuringova mašina bez početka i kraja, i neka M izračunava f -ju f

Potrebno je pokazati da postoji LTM N koja izračunava f

Traka mašine $N \rightarrow \begin{array}{cccc} +0 & +1 & +2 & \dots \\ -1 & -2 & -3 & \dots \end{array}$

Za $A(M) = \{0,1\}$

$A(N) = (A(M) \times A(M)) \cup \{\rightarrow\} = \{(0,0), (0,1), (1,0), (1,1), \rightarrow\}$

Za suprotan smer, moramo primeniti pomenuto pravilo o dokazivanju jednakosti TM-a

Neka je M Tjuringova mašina bez početka i kraja, i neka M izračunava f -ju f

Potrebno je pokazati da postoji LTM N koja izračunava f

Traka mašine $N \rightarrow \begin{array}{cccc} +0 & +1 & +2 & \dots \\ -1 & -2 & -3 & \dots \end{array}$

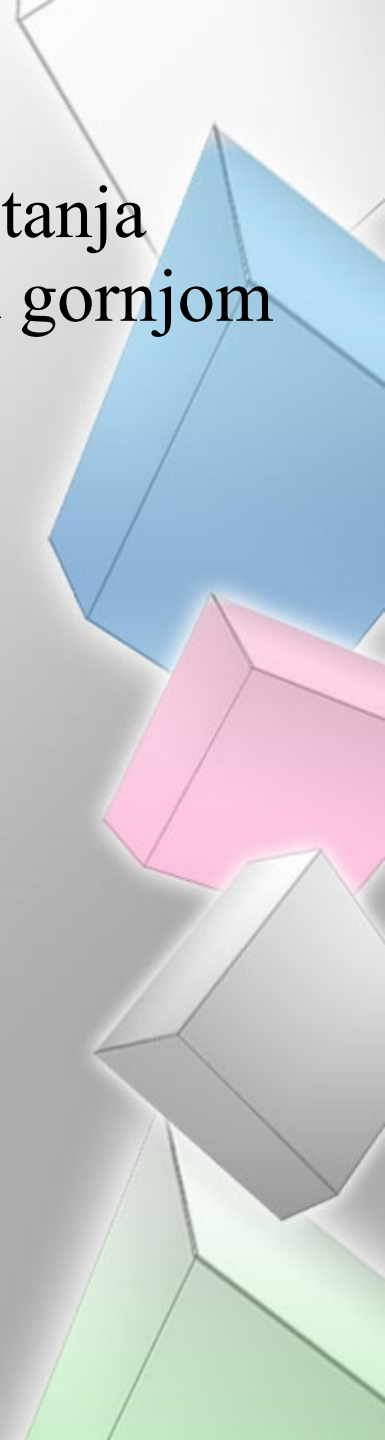
Za $A(M) = \{0, 1\}$

$A(N) = (A(M) \times A(M)) \cup \{\rightarrow\} = \{(0,0), (0,1), (1,0), (1,1), \rightarrow\}$

Ćelije trake od N sadrže parove znakova iz ćelija od M (indeksi: $(i, -i-1), i = \overline{0, \infty}$)

Za svako stanje q mašine M , N ima dva stanja $(q,1)$ i $(q,2)$, tj. stanje u kojem N radi nad gornjom odnosno donjom trakom

$(q_z, 1)$ i $(q_z, 2)$ su završna stanja za N



Za svako stanje q mašine M , N ima dva stanja $(q,1)$ i $(q,2)$, tj. stanje u kojem N radi nad gornjom odnosno donjom trakom

$(q_z, 1)$ i $(q_z, 2)$ su završna stanja za N

Transformacije naredbi:

$q_i s o q_j$ naredba mašine M , tada je

(i) $(q_i, 1)(s, x)\bar{o}(q_j, 1)$
odgovarajuća naredba mašine N , pri čemu

$$(o = L \vee o = R) \Rightarrow \bar{o} = o$$

$$(o = 1 \vee o = 0) \Rightarrow \bar{o} = (o, x), x \in \{0, 1\}$$

Ovom naredbom menjamo položaj glave ili vrednost ćelije u gornjem redu trake od N , tj. desnoj (nenegativnoj) polovini trake od M

(ii) $(q_i, 2)(x, s)\bar{o}(q_j, 2)$
odgovarajuća naredba mašine N, pri čemu

$$o = L \Rightarrow \bar{o} = R$$

$$o = R \Rightarrow \bar{o} = L$$

$$(o = 1 \vee o = 0) \Rightarrow \bar{o} = (x, o), x \in \{0, 1\}$$

Ovom naredbom menjamo položaj glave ili vrednost ćelije u donjem redu trake od N, tj. levoj (negativnoj) polovini trake od M

Kretanje ulevo nad levoj polovini trake od M je kretanje udesno na donjoj traci od N

Za svako stanje q mašine M , u N se dodaju naredbe
 $(q, 1) \rightarrow R(q, 2)$ i $(q, 2) \rightarrow R(q, 1)$

Do ćelije $sa \rightarrow$ može se stići kretanjem ulevo kroz gornji red trake od N , i tada prelazimo u donji, ali sada kretanje ulevo u M je kretanje udesno u M

Analogno reagujemo u slučaju kada kretajući se donjom trakom ulevo dodjemo do početne ćelije

Za svako stanje q mašine M , u N se dodaju naredbe
 $(q, 1) \rightarrow R(q, 2)$ i $(q, 2) \rightarrow R(q, 1)$

Do ćelije $sa \rightarrow$ može se stići kretanjem ulevo kroz gornji red trake od N , i tada prelazimo u donji, ali sada kretanje ulevo u M je kretanje udesno u M

Analogno reagujemo u slučaju kada kretajući se donjom trakom ulevo dodjemo do početne ćelije

- Ovako opisana mašina N ne samo da izračunava f , nego iz nje lako možemo rekonstruisati ponašanje mašine N , tj. simulira je.

Za svako stanje q mašine M , u N se dodaju naredbe
 $(q, 1) \rightarrow R(q, 2)$ i $(q, 2) \rightarrow R(q, 1)$

Do ćelije $sa \rightarrow$ može se stići kretanjem ulevo kroz gornji red trake od N , i tada prelazimo u donji, ali sada kretanje ulevo u M je kretanje udesno u M

Analogno reagujemo u slučaju kada kretajući se donjom trakom ulevo dodjemo do početne ćelije

- Ovako opisana mašina N ne samo da izračunava f , nego iz nje lako možemo rekonstruisati ponašanje mašine N , tj. simulira je.
- Dakle, klasa T -izračunljivih f -ja je podklasa u klasi LT -izračunljivih f -ja, a kako obrnuto važi trivijalno, te dve klase su jednake.

Za svako stanje q mašine M , u N se dodaju naredbe
 $(q, 1) \rightarrow R(q, 2)$ i $(q, 2) \rightarrow R(q, 1)$

Do ćelije $sa \rightarrow$ može se stići kretanjem ulevo kroz gornji red trake od N , i tada prelazimo u donji, ali sada kretanje ulevo u M je kretanje udesno u M

Analogno reagujemo u slučaju kada kretajući se donjom trakom ulevo dodjemo do početne ćelije

- Ovako opisana mašina N ne samo da izračunava f , nego iz nje lako možemo rekonstruisati ponašanje mašine N , tj. simulira je.
- Dakle, klasa T -izračunljivih f -ja je podklasa u klasi LT -izračunljivih f -ja, a kako obrnuto važi trivijalno, te dve klase su jednake.

