

Registar mašine

N. Ikodinović

March 5, 2018

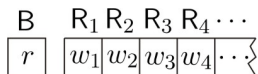
Pregled predavanja

- 1 (Ne)formalni opis Registar mašina
- 2 RM-programi za unarni ulazni alfabet $\Sigma = \{1\}$
- 3 Odnos Tjuringovih mašina i registar mašina

Pregled predavanja

- 1 (Ne)formalni opis Registar mašina
- 2 RM-programi za unarni ulazni alfabet $\Sigma = \{1\}$
- 3 Odnos Tjuringovih mašina i registar mašina

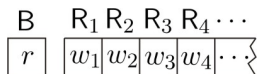
Registar mašina



$R_1, R_2, R_3, \dots$ je niz *registara* u koje može biti upisana **čitava reč** unapred izabranog alfabeta $\Sigma = \{s_1, \dots, s_m\}$,

B je *brojač*, u koji se unosi redni broj instrukcije RM-programa.

Registar mašina



$R_1, R_2, R_3, \dots$ je niz *registara* u koje može biti upisana **čitava reč** unapred izabranog alfabeta $\Sigma = \{s_1, \dots, s_m\}$,

B je *brojač*, u koji se unosi redni broj instrukcije RM-programa.

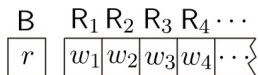
RM-program je konačan niza instrukcija redom numerisanih brojevima od 1 do nekog n : $I_1, \dots, I_n$, pri čemu je svaka instrukcija I_i sledećeg oblika:

$$i. \quad R_k^{+s} \mid \ell \quad (s \in \Sigma, \ell \geq 1)$$

ili

$$i. \quad R_k^- \mid \begin{array}{l} \varepsilon \quad \ell_0 \\ s_1 \quad \ell_1 \\ \vdots \\ s_m \quad \ell_m \end{array} \quad (\ell_0, \ell_1, \dots, \ell_m \geq 1)$$

Konfiguracije



Konfiguracija nad Σ je niz:

$$(C) \quad r; w_1, w_2, w_3, \dots$$

čiji je prvi član prirodni broj različit od nule (upisan u brojač), a ostali članovi su reči iz Σ^* upisane redom u registre $R_1, R_2, \dots$

Konfiguracije

B	R ₁	R ₂	R ₃	R ₄	...
r	w_1	w_2	w_3	w_4	$\cdots$

Konfiguracija nad Σ je niz:

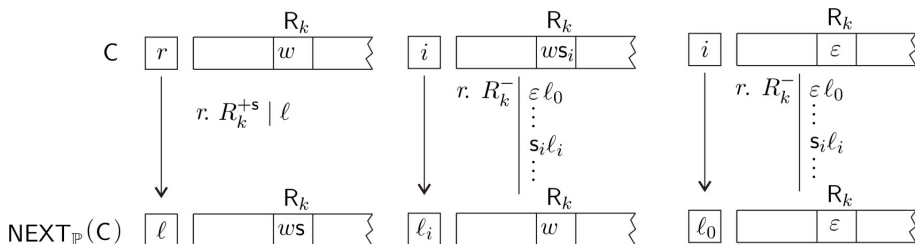
$$(C) \quad r; w_1, w_2, w_3, \dots$$

čiji je prvi član prirodni broj različit od nule (upisan u brojač), a ostali članovi su reči iz Σ^* upisane redom u registre $R_1, R_2, \dots$.

Konfiguracija (C) je **završna konfiguracija** programa $\mathbb{P}$ ako $\mathbb{P}$ ne sadrži instrukciju čiji je redni broj r , tj. ako je $r > n$.

NEXT _{$\mathbb{P}$}

Ako (C) nije završna konfiguracija programa $\mathbb{P}$, onda je ovim programom određen jedinstveni sledbenik NEXT _{$\mathbb{P}$} (C):



Početne konfiguracije

Rad RM-programa $\mathbb{P}$: $l_1, l_2, \dots, l_n$ posmatraćemo isključivo za **početne konfiguracije** u kojima je u brojaču upisan broj 1 (izvršavanje programa počinje instrukcijom koja je prva navedena u $\mathbb{P}$) i u konačno mnogo registara su upisane neke reči alfabeta Σ , a u svim ostalim je automatski upisana prazna reč:

$$(C_0) \quad 1; w_1, \dots, w_k, \varepsilon, \varepsilon, \varepsilon, \dots$$

(k naziva *dubina ulaza*)

Izračunavanje

Svaki RM-program $\mathbb{P}$ za ulaz $(w_1, \dots, w_k)$ generiše jedan niz konfiguracija:

$$C_0 \rightarrow \text{NEXT}_{\mathbb{P}}(C_0) \rightarrow \text{NEXT}_{\mathbb{P}}(\text{NEXT}_{\mathbb{P}}(C_0)) \rightarrow \dots$$

koji je:

- *konačan* ako se u nizu pojavi završna konfiguracija za program $\mathbb{P}$; tada pišemo $\mathbb{P}(w_1, \dots, w_k) \downarrow$ i kažemo *program $\mathbb{P}$ se zaustavlja (konverigra) za ulaz $(w_1, \dots, w_k)$* ; ili
- *beskonačan* ako se u nizu nikada ne pojavljuje završna konfiguracija; tada pišemo $\mathbb{P}(w_1, \dots, w_k) \uparrow$ i kažemo *program $\mathbb{P}$ se ne zaustavlja (divergira) za ulaz $(w_1, \dots, w_k)$* .

Izračunavanje

Ako $\mathbb{P}(w_1, \dots, w_k) \downarrow$, konačan niz konfiguracija $C_0, C_1, \dots, C_d$, gde je:

- C_0 početna konfiguracija određena ulazom $(w_1, \dots, w_k)$,
- $C_{i+1} = \text{NEXT}_{\mathbb{P}}(C_i)$, $1 \leq i < d$; par susednih konfiguracija nazivamo *računskim korakom*; i
- C_d završna konfiguracija za $\mathbb{P}$,

nazivamo **izračunavanjem** programa $\mathbb{P}$ za ulaz $(w_1, \dots, w_k)$ i pišemo $C_0 \rightarrow_{\mathbb{P}}^* C_d$.

Izračunavanje

$\|\mathbb{P}\|$ – najmanji prirodan broj takav da se u programu $\mathbb{P}$ pominju samo registri R_i , $i \leq \|\mathbb{P}\|$, naziva se *dubina programa* $\mathbb{P}$;

Sve dostižne konfiguracije tokom rada programa $\mathbb{P}$ možemo označavati konačnim nizom $i, w_1, \dots, w_d$, $d = \max\{n, \|\mathbb{P}\|\}$, gde je n *dubina ulaza*, a $\|\mathbb{P}\|$ *dubina programa*.

Primer 1

Neka je $\mathbb{P}$ program nad alfabetom $\{0,1\}$:

$$\begin{array}{lll}
 1. & R_2^- & \left| \begin{array}{cc} \varepsilon & 4 \\ 0 & 2 \\ 1 & 3 \end{array} \right. \\
 2. & R_1^{+0} & \mid 1 \\
 3. & R_1^{+1} & \mid 1
 \end{array}$$

Primer 1

Neka je $\mathbb{P}$ program nad alfabetom $\{0,1\}$:

$$\begin{array}{lcl}
 1. & R_2^- & \left| \begin{array}{cc} \varepsilon & 4 \\ 0 & 2 \\ 1 & 3 \end{array} \right. \\
 2. & R_1^{+0} & \left| 1 \right. \\
 3. & R_1^{+1} & \left| 1 \right.
 \end{array}$$

Posmatramo izračunavanje ovog programa za ulaz (001,110).

Primer 1

Neka je $\mathbb{P}$ program nad alfabetom $\{0,1\}$:

$$\begin{array}{lll}
 1. & R_2^- & \left| \begin{array}{cc} \varepsilon & 4 \\ 0 & 2 \\ 1 & 3 \end{array} \right. & 2. & R_1^{+0} & \left| 1 \right. & 3. & R_1^{+1} & \left| 1 \right.
 \end{array}$$

Posmatramo izračunavanje ovog programa za ulaz (001,110).

B	R_1	R_2	
1	001	110	...
2	001	11	...
1	0010	11	...
3	0010	1	...
1	00101	1	...
3	00101		...
1	001011		...
4	001011		...
STOP			

Primer 1

Neka je $\mathbb{P}$ program nad alfabetom $\{0,1\}$:

$$\begin{array}{lll}
 1. & R_2^- & \left| \begin{array}{cc} \varepsilon & 4 \\ 0 & 2 \\ 1 & 3 \end{array} \right. & 2. & R_1^{+0} & \left| 1 \right. & 3. & R_1^{+1} & \left| 1 \right.
 \end{array}$$

Posmatramo izračunavanje ovog programa za ulaz (001,110).

$1; 01, 00101, 11 \rightarrow^* ?$

B	R ₁	R ₂	
1	001	110	...
2	001	11	...
1	0010	11	...
3	0010	1	...
1	00101	1	...
3	00101		...
1	001011		...
4	001011		...
STOP			

Primer 1

Neka je $\mathbb{P}$ program nad alfabetom $\{0,1\}$:

$$\begin{array}{lll}
 1. & R_2^- & \left| \begin{array}{cc} \varepsilon & 4 \\ 0 & 2 \\ 1 & 3 \end{array} \right. & 2. & R_1^{+0} & \left| 1 \right. & 3. & R_1^{+1} & \left| 1 \right.
 \end{array}$$

Posmatramo izračunavanje ovog programa za ulaz (001,110).

B	R ₁	R ₂	
1	001	110	⋯
2	001	11	⋯
1	0010	11	⋯
3	0010	1	⋯
1	00101	1	⋯
3	00101		⋯
1	001011		⋯
4	001011		⋯
STOP			

1;01,00101,11 $\rightarrow^*$? 1;1101 $\rightarrow^*$?

Primer 2

Program $\mathbb{P}$, nad $\{a, b\}$ sadrži samo jednu instrukciju :

1.	R_1^-	ε	1
		a	1
		b	2

Primer 2

Program $\mathbb{P}$, nad $\{a, b\}$ sadrži samo jednu instrukciju :

1.	R_1^-	ε	1
		a	1
		b	2

$1; aaa \rightarrow^* ?$

Primer 2

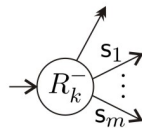
Program $\mathbb{P}$, nad $\{a, b\}$ sadrži samo jednu instrukciju :

1.	R_1^-		ε	1
			a	1
			b	2

$1; aaa \rightarrow^*?$ $1; aaabaab \rightarrow^*?$

Grafički prikaz RM-programa

 $R_k^{+s} \mid \ell$

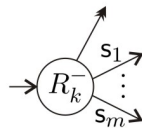
 $R_k^- \mid \begin{array}{c} \varepsilon \\ s_1 \\ \vdots \\ s_m \end{array} \mid \begin{array}{c} \ell_0 \\ \ell_1 \\ \vdots \\ \ell_m \end{array}$


Grafički prikaz RM-programa

$$R_k^{+s} \mid \ell$$



$$R_k^- \mid \begin{array}{cc} \varepsilon & \ell_0 \\ s_1 & \ell_1 \\ \vdots & \\ s_m & \ell_m \end{array}$$



Neka je $\mathbb{P}$ program nad alfabetom $\{0,1\}$:

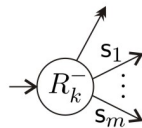
$$\begin{array}{lll} 1. & R_2^- \mid \begin{array}{cc} \varepsilon & 4 \\ 0 & 2 \\ 1 & 3 \end{array} & 2. & R_1^{+0} \mid 1 & 3. & R_1^{+1} \mid 1 \end{array}$$

Grafički prikaz RM-programa

$$R_k^{+s} \mid \ell$$

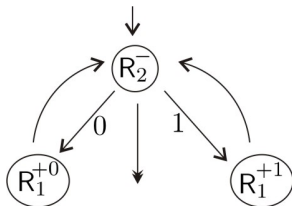


$$R_k^- \mid \begin{array}{ll} \varepsilon & \ell_0 \\ s_1 & \ell_1 \\ \vdots & \\ s_m & \ell_m \end{array}$$



Neka je $\mathbb{P}$ program nad alfabetom $\{0,1\}$:

$$\begin{array}{lll} 1. & R_2^- \mid \begin{array}{ll} \varepsilon & 4 \\ 0 & 2 \\ 1 & 3 \end{array} & 2. \quad R_1^{+0} \mid 1 \quad 3. \quad R_1^{+1} \mid 1 \end{array}$$



Nekoliko primera nad $\Sigma = \{s_1, \dots, s_m\}$

$$R_i := \varepsilon$$

Nekoliko primera nad $\Sigma = \{s_1, \dots, s_m\}$

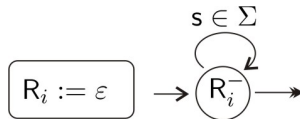
$$R_i := \varepsilon$$

$$1. \quad R_i^- \left| \begin{array}{ll} \varepsilon & 2 \\ s_1 & 1 \\ \vdots & \\ s_m & 1 \end{array} \right.$$

Nekoliko primera nad $\Sigma = \{s_1, \dots, s_m\}$

$$R_i := \varepsilon$$

$$1. \quad R_i^- \mid \begin{array}{ll} \varepsilon & 2 \\ s_1 & 1 \\ \vdots & \\ s_m & 1 \end{array}$$



Nekoliko primera nad $\Sigma = \{s_1, \dots, s_m\}$

$$R_j := R_j R_i$$

Nekoliko primera nad $\Sigma = \{s_1, \dots, s_m\}$

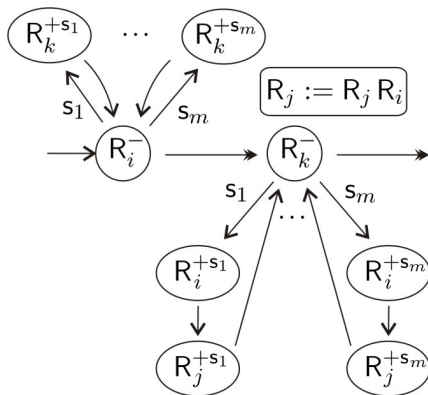
$$R_j := R_j R_i$$

Na sadržaj registra R_j dopisuje se (zdesna) sadržaj registra R_i , uz pomoć registra R_k u kome je prazna reč i na početku i na kraju. Nakon izvršavanja ovog programa, sadržaj registra R_i ostaje isti kao što je bio na početku.

Nekoliko primera nad $\Sigma = \{s_1, \dots, s_m\}$

$$R_j := R_j R_i$$

Na sadržaj registra R_j dopisuje se (zdesna) sadržaj registra R_i , uz pomoć registra R_k u kome je prazna reč i na početku i na kraju. Nakon izvršavanja ovog programa, sadržaj registra R_i ostaje isti kao što je bio na početku.



Nekoliko primera nad $\Sigma = \{s_1, \dots, s_m\}$

$R_j := R_i$

Nekoliko primera nad $\Sigma = \{s_1, \dots, s_m\}$

$$R_j := R_i$$

Sadržaj registra R_i se kopira u R_j uz pomoć registra R_k u kome je prazna reč i na početku kopiranja i nakon kopiranja; posle kopiranja, sadržaj registra R_i ostaje isti kao što je bio na početku.

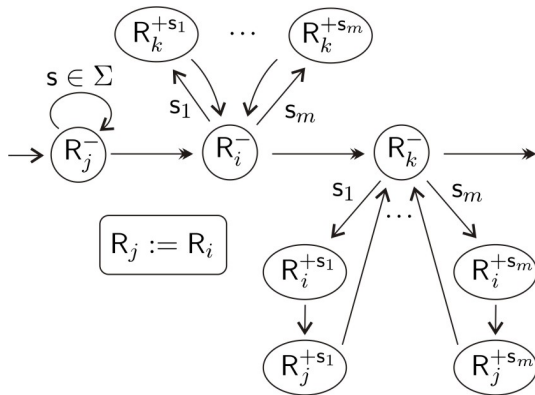
$R_j := R_i$ dobijamo *nadovezivanjem* programa, $R_j := \varepsilon$ i $R_j := R_j R_i$.

Nekoliko primera nad $\Sigma = \{s_1, \dots, s_m\}$

$$R_j := R_i$$

Sadržaj registra R_i se kopira u R_j uz pomoć registra R_k u kome je prazna reč i na početku kopiranja i nakon kopiranja; posle kopiranja, sadržaj registra R_i ostaje isti kao što je bio na početku.

$R_j := R_i$ dobijamo *nadovezivanjem* programa, $R_j := \varepsilon$ i $R_j := R_j R_i$.



Standardna forma RM-programa

RM-program $\mathbb{P} = l_1, \dots, l_n$ je u **standardnoj formi** ukoliko za svaki parametar prelaza ℓ koji se pojavljuje u nekoj instrukciji l_i , $1 \leq i \leq n$, važi $1 \leq \ell \leq n+1$.

Standardna forma RM-programa

RM-program $\mathbb{P} = l_1, \dots, l_n$ je u **standardnoj formi** ukoliko za svaki parametar prelaza ℓ koji se pojavljuje u nekoj instrukciji l_i , $1 \leq i \leq n$, važi $1 \leq \ell \leq n+1$.

Dva RM-programa su **ekvivalentna** ako za sve ulaze $(w_1, \dots, w_k)$, $k \geq 1$, generišu ista izračunavanja.

Standardna forma RM-programa

RM-program $\mathbb{P} = l_1, \dots, l_n$ je u **standardnoj formi** ukoliko za svaki parametar prelaza ℓ koji se pojavljuje u nekoj instrukciji l_i , $1 \leq i \leq n$, važi $1 \leq \ell \leq n+1$.

Dva RM-programa su **ekvivalentna** ako za sve ulaze $(w_1, \dots, w_k)$, $k \geq 1$, generišu ista izračunavanja.

Svaki RM-program se može preraditi u ekvivalentan RM-program koji je u standardnoj formi.

Nadovezivanje RM-programa

Ako su $\mathbb{P}_1$ i $\mathbb{P}_2$ dva RM-programa u standardnoj formi:

$$\mathbb{P}_1 : 1. l_1 \cdots n_1. l_{n_1}; \quad \mathbb{P}_2 : 1. J_1 \cdots n_2. J_{n_2},$$

onda **nadovezivanjem** $\mathbb{P}_2$ na $\mathbb{P}_1$ dobijamo program

$$\mathbb{P}_1 \mathbb{P}_2 : 1. l_1 \cdots n_1. l_{n_1} \quad n_1 + 1. J'_1 \cdots n_2 + n_1. J'_{n_2},$$

pri čemu je svaka instrukcija J'_i dobijena iz J_i tako što se parametri prelaza uvećavaju za n_1 .

Nadovezivanje RM-programa

Ako su $\mathbb{P}_1$ i $\mathbb{P}_2$ dva RM-programa u standardnoj formi:

$$\mathbb{P}_1 : 1. l_1 \cdots n_1. l_{n_1}; \quad \mathbb{P}_2 : 1. J_1 \cdots n_2. J_{n_2},$$

onda **nadovezivanjem** $\mathbb{P}_2$ na $\mathbb{P}_1$ dobijamo program

$$\mathbb{P}_1\mathbb{P}_2 : 1. l_1 \cdots n_1. l_{n_1} \quad n_1 + 1. J'_1 \cdots n_2 + n_1. J'_{n_2},$$

pri čemu je svaka instrukcija J'_i dobijena iz J_i tako što se parametri prelaza uvećavaju za n_1 .

Analogno definišemo nadovezivanje tri i više programa. Npr. $\mathbb{P}_1\mathbb{P}_2\mathbb{P}_3$ je program dobijen nadovezivanjem programa $\mathbb{P}_3$ na $\mathbb{P}_1\mathbb{P}_2$.

Pregled predavanja

- 1 (Ne)formalni opis Registar mašina
- 2 RM-programi za unarni ulazni alfabet $\Sigma = \{1\}$
- 3 Odnos Tjuringovih mašina i registar mašina

Neka je $\Sigma = \{1\}$...

Reči alfabeta $\Sigma = \{1\}$ identifikujemo sa skupom prirodnih brojeva $\mathbb{N}$:

reč $1^n = \underbrace{11 \cdots 1}_n$ identifikujemo sa n .

Neka je $\Sigma = \{1\}$...

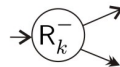
Reči alfabeta $\Sigma = \{1\}$ identifikujemo sa skupom prirodnih brojeva $\mathbb{N}$:

reč $1^n = \underbrace{11 \dots 1}_n$ identifikujemo sa n .

Instrukcije nad $\Sigma = \{1\}$:

$$R_k^+ \mid \ell$$

$$R_k^- \mid \begin{array}{cc} \varepsilon & \ell_0 \\ 1 & \ell_1 \end{array}$$



NEXT _{$\mathbb{P}$}

Za alfabet $\{1\}$, konfiguracije su zapravo nizovi prirodnih brojeva čiji su svi članovi počev od nekog jednaki nuli.

NEXT $_{\mathbb{P}}$

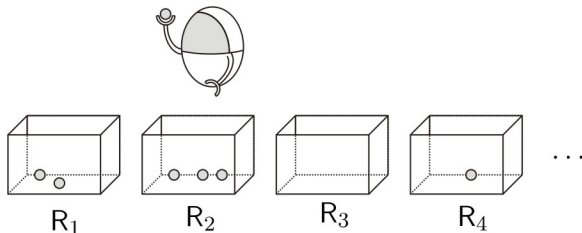
Za alfabet $\{1\}$, konfiguracije su zapravo nizovi prirodnih brojeva čiji su svi članovi počev od nekog jednaki nuli.

$$\text{NEXT}_{\mathbb{P}}(i; r_1, \dots, r_k, \dots, r_d, 0 \dots) = \begin{cases} (\ell; r_1, \dots, r_k + 1, \dots, r_d, 0 \dots), & i\text{-ta instrukcija je } R_k^{+1} \mid \ell, \\ (\ell_0; r_1, \dots, r_k, \dots, r_d, 0 \dots), & i\text{-ta instrukcija je } R_k^{-} \mid \ell_0, \ell_1 \text{ i } r_k = 0, \\ (\ell_1; r_1, \dots, r_k - 1, \dots, r_d, 0 \dots), & i\text{-ta instrukcija je } R_k^{-} \mid \ell_0, \ell_1 \text{ i } r_k \neq 0, \\ (i; r_1, \dots, r_k, \dots, r_d, 0 \dots), & i\text{-ta instrukcija programa ne postoji.} \end{cases}$$

NEXT _{$\mathbb{P}$}

Za alfabet $\{1\}$, konfiguracije su zapravo nizovi prirodnih brojeva čiji su svi članovi počev od nekog jednaki nuli.

$$\text{NEXT}_{\mathbb{P}}(i; r_1, \dots, r_k, \dots, r_d, 0 \dots) = \begin{cases} (\ell; r_1, \dots, r_k + 1, \dots, r_d, 0 \dots), & i\text{-ta instrukcija je } R_k^{+1} \mid \ell, \\ (\ell_0; r_1, \dots, r_k, \dots, r_d, 0 \dots), & i\text{-ta instrukcija je } R_k^{-} \mid \ell_0, \ell_1 \text{ i } r_k = 0, \\ (\ell_1; r_1, \dots, r_k - 1, \dots, r_d, 0 \dots), & i\text{-ta instrukcija je } R_k^{-} \mid \ell_0, \ell_1 \text{ i } r_k \neq 0, \\ (i; r_1, \dots, r_k, \dots, r_d, 0 \dots), & i\text{-ta instrukcija programa ne postoji.} \end{cases}$$



Primer

$$|\cdot, \cdot| : \mathbb{N} \times \mathbb{N} \xrightarrow{\text{na}} \mathbb{N}^+$$

$$|x_1, x_2| = 2^{x_1}(2x_2 + 1)$$

Primer

$$|\cdot, \cdot| : \mathbb{N} \times \mathbb{N} \xrightarrow{\text{na}} \mathbb{N}^+$$

$$|x_1, x_2| = 2^{x_1}(2x_2 + 1)$$

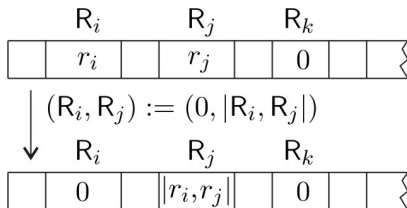
$$(R_i, R_j) := (0, |R_i, R_j|)$$

Primer

$$|\cdot, \cdot| : \mathbb{N} \times \mathbb{N} \xrightarrow{\text{na}} \mathbb{N}^+$$

$$|x_1, x_2| = 2^{x_1}(2x_2 + 1)$$

$$(R_i, R_j) := (0, |R_i, R_j|)$$

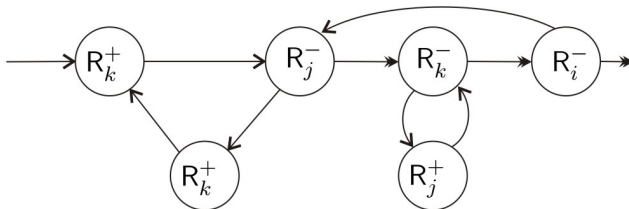
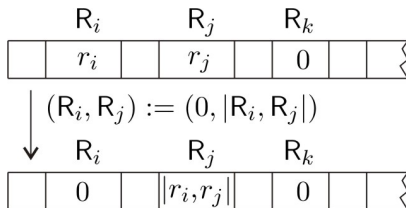


Primer

$$|\cdot, \cdot| : \mathbb{N} \times \mathbb{N} \xrightarrow{\text{na}} \mathbb{N}^+$$

$$|x_1, x_2| = 2^{x_1}(2x_2 + 1)$$

$$(R_i, R_j) := (0, |R_i, R_j|)$$



Primer

$$|x_1, x_2| = 2^{x_1}(2x_2 + 1)$$

$\text{hd}(n)$ = prvi član niza čiji je kôd n

$\text{tl}(n)$ = kôd repa niza čiji je kôd n

$$(R_i, R_j) := (\text{hd}(R_j), \text{tl}(R_j))$$

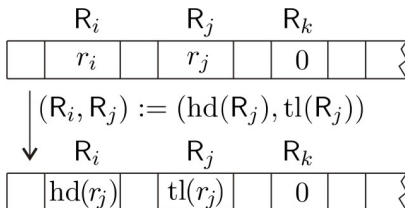
Primer

$$|x_1, x_2| = 2^{x_1}(2x_2 + 1)$$

$\text{hd}(n) =$ prvi član niza čiji je kôd n

$\text{tl}(n) =$ kôd repa niza čiji je kôd n

$$(R_i, R_j) := (\text{hd}(R_j), \text{tl}(R_j))$$



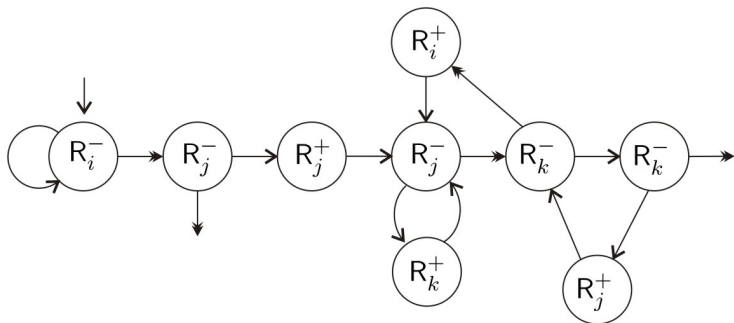
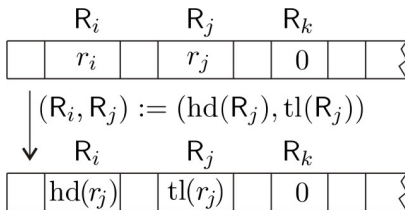
Primer

$$|x_1, x_2| = 2^{x_1}(2x_2 + 1)$$

$\text{hd}(n)$ = prvi član niza čiji je kôd n

$\text{tl}(n)$ = kôd repa niza čiji je kôd n

$$(R_i, R_j) := (\text{hd}(R_j), \text{tl}(R_j))$$



Uslovni prelazak

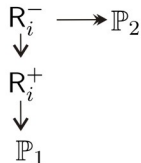
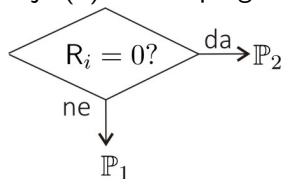
if $R_i = 0$? then $\mathbb{P}_1$ else $\mathbb{P}_2$

Uslovni prelazak

if $R_i = 0?$ then $\mathbb{P}_1$ else $\mathbb{P}_2$

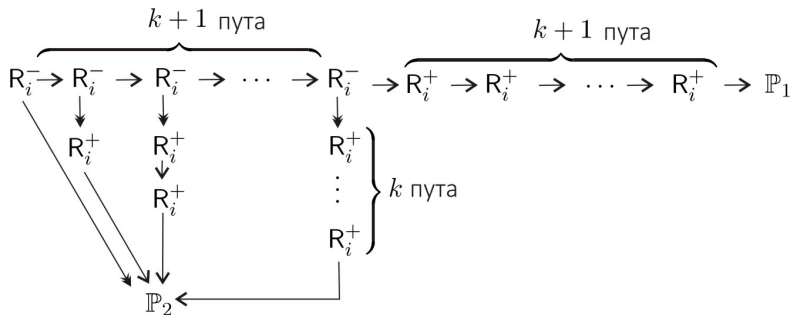
Ako je $\mathbb{P}_1 = I_1, \dots, I_{n+1}$ i $\mathbb{P}_2 = J_1, \dots, J_{n+1}$, onda je (*) sledeći program:

1. $R_i^- \mid n_1 + 3, 2$
2. $R_i^+ \mid 3$
3. I'_1
- $\vdots$
- $n_1 + 2.$ I'_{n_1}
- $n_1 + 3.$ J'_1
- $\vdots$
- $n_1 + n_2 + 3.$ J'_{n_2}



Zadatak

if $R_i > k$? then $\mathbb{P}_1$ else $\mathbb{P}_2$ (k je fiksirani prirodni broj)



RM-izračunljivost/odlučivost

ko se izvršavanje programa $\mathbb{P}$ za ulaz $(w_1, \dots, w_k)$ zaustavlja, tj.
 $\mathbb{P}(w_1, \dots, w_k) \downarrow$, pod **rezultatom** izračunavanja podrazumevamo reč v koja
je sadržaj prvog registra R_1 u završnoj konfiguraciji: $\mathbb{P}(w_1, \dots, w_k) \downarrow v$.

RM-izračunljivost/odlučivost

ko se izvršavanje programa $\mathbb{P}$ za ulaz $(w_1, \dots, w_k)$ zaustavlja, tj.
 $\mathbb{P}(w_1, \dots, w_k) \downarrow$, pod **rezultatom** izračunavanja podrazumevamo reč v koja
 je sadržaj prvog registra R_1 u završnoj konfiguraciji: $\mathbb{P}(w_1, \dots, w_k) \downarrow v$.

Definicija

Funkcija $f : \Sigma^{*k} \rightarrow \Sigma^*$ je **RM-izračunljiva**, ako postoji RM-program $\mathbb{P}$
 takav da za sve $w_1, \dots, w_k \in \Sigma^*$ važi $\mathbb{P}(w_1, \dots, w_k) \downarrow f(w_1, \dots, w_k)$.

Definicija

Skup $A \subseteq \mathbb{N}^k$ je **RM-odlučiv** ako je RM-izračunljiva njegova
 karakteristična funkcija $\chi_A : \mathbb{N}^k \rightarrow \{0, 1\}$,

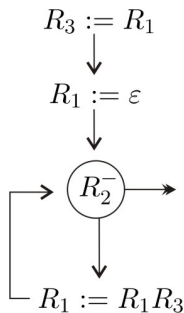
$$\chi_A(n_1, \dots, n_k) = \begin{cases} 1, & (n_1, \dots, n_k) \in A, \\ 0, & (n_1, \dots, n_k) \notin A. \end{cases}$$

Primer

Množenje $\cdot : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ je RM-izračunljiva funkcija.

Primer

Množenje $\cdot : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ je RM-izračunljiva funkcija.



Pregled predavanja

- 1 (Ne)formalni opis Registar mašina
- 2 RM-programi za unarni ulazni alfabet $\Sigma = \{1\}$
- 3 Odnos Tjuringovih mašina i registar mašina

Svaki RM-program može se simulirati nekom TM.

Za proizvoljan RM-program $\mathbb{P} = (l_1, \dots, l_n)$ nad $\Sigma = \{s_1, \dots, s_m\}$,

Svaki RM-program može se simulirati nekom TM.

Za proizvoljan RM-program $\mathbb{P} = (l_1, \dots, l_n)$ nad $\Sigma = \{s_1, \dots, s_m\}$, postoji TM $\mathbb{T}$, sa ulaznim alfabetom Σ , alfabetom trake $\Gamma = \Sigma \cup \{1, \sqcup\}$ (pri čemu $1, \sqcup \notin \Sigma$), početnim stanjem q_0 i jednim završnim stanjem q_{stop} takva da za sve $k \geq 0$ i sve reči $w_1, \dots, w_k \in \Sigma^*$, važe sledeći uslovi:

Svaki RM-program može se simulirati nekom TM.

Za proizvoljan RM-program $\mathbb{P} = (l_1, \dots, l_n)$ nad $\Sigma = \{s_1, \dots, s_m\}$, postoji TM $\mathbb{T}$, sa ulaznim alfabetom Σ , alfabetom trake $\Gamma = \Sigma \cup \{1, \sqcup\}$ (pri čemu $1, \sqcup \notin \Sigma$), početnim stanjem q_0 i jednim završnim stanjem q_{stop} takva da za sve $k \geq 0$ i sve reči $w_1, \dots, w_k \in \Sigma^*$, važe sledeći uslovi:

(1) ako $\mathbb{P}(w_1, \dots, w_k) \uparrow$, onda $\mathbb{T}(w_1, \dots, w_k) \uparrow$,

Svaki RM-program može se simulirati nekom TM.

Za proizvoljan RM-program $\mathbb{P} = (l_1, \dots, l_n)$ nad $\Sigma = \{s_1, \dots, s_m\}$, postoji TM $\mathbb{T}$, sa ulaznim alfabetom Σ , alfabetom trake $\Gamma = \Sigma \cup \{1, \sqcup\}$ (pri čemu $1, \sqcup \notin \Sigma$), početnim stanjem q_0 i jednim završnim stanjem q_{stop} takva da za sve $k \geq 0$ i sve reči $w_1, \dots, w_k \in \Sigma^*$, važe sledeći uslovi:

- (1) ako $\mathbb{P}(w_1, \dots, w_k) \uparrow$, onda $\mathbb{T}(w_1, \dots, w_k) \uparrow$,
- (2) ako $\mathbb{P}(w_1, \dots, w_k) \downarrow$, onda $\mathbb{T}(w_1, \dots, w_k) \downarrow$; štaviše ako

$$1; w_1, \dots, w_k \rightarrow_{\mathbb{P}}^* n+1; v_1, \dots, v_d, \text{ gde je } d = \max\{k, \|\mathbb{P}\|\},$$

$$\text{onda } q_0 w_1 \sqcup w_2 \sqcup \dots \sqcup w_k \rightarrow_{\mathbb{T}}^* q_{\text{stop}} v_1 \sqcup \dots \sqcup v_d.$$

Svaka TM može se simulirati RM-programom.

Uz pogodno kodiranje, za svaku Turingovu mašinu

$\mathbb{T} = (Q, q_0, F, \Sigma, \sqcup, \Gamma, \tau)$ postoji RM-program nad unarnim alfabetom $\{1\}$ koji simulira rad mašine $\mathbb{T}$.

Svaka TM može se simulirati RM-programom.

Uz pogodno kodiranje, **za svaku Tjuringovu mašinu**

$\mathbb{T} = (Q, q_0, F, \Sigma, \sqcup, \Gamma, \tau)$ **postoji** RM-program nad unarnim alfabetom $\{1\}$ koji simulira rad mašine $\mathbb{T}$.

Stanja mašine $\mathbb{T}$ su $q_0, \dots, q_k, q_{k+1}, \dots, q_n$, pri čemu je q_0 početno stanje, $q_{k+1}, \dots, q_n$ su završna stanja.

Svaka TM može se simulirati RM-programom.

Uz pogodno kodiranje, **za svaku Turingovu mašinu**

$\mathbb{T} = (Q, q_0, F, \Sigma, \sqcup, \Gamma, \tau)$ **postoji** RM-program nad unarnim alfabetom $\{1\}$ koji simulira rad mašine $\mathbb{T}$.

Stanja mašine $\mathbb{T}$ su $q_0, \dots, q_k, q_{k+1}, \dots, q_n$, pri čemu je q_0 početno stanje, $q_{k+1}, \dots, q_n$ su završna stanja.

Simboli trake su $s_0, s_1, \dots, s_m$, pri čemu je s_0 blanko znak $\sqcup$.

Svaka TM može se simulirati RM-programom.

Uz pogodno kodiranje, **za svaku Turingovu mašinu**

$\mathbb{T} = (Q, q_0, F, \Sigma, \sqcup, \Gamma, \tau)$ **postoji** RM-program nad unarnim alfabetom $\{1\}$ koji simulira rad mašine $\mathbb{T}$.

Stanja mašine $\mathbb{T}$ su $q_0, \dots, q_k, q_{k+1}, \dots, q_n$, pri čemu je q_0 početno stanje, $q_{k+1}, \dots, q_n$ su završna stanja.

Simboli trake su $s_0, s_1, \dots, s_m$, pri čemu je s_0 blanko znak $\sqcup$.

Svakom stanju i i svakom simbolu pridružuje se kôd: $[q_i] = i, [s_i] = i$.

Svaka TM može se simulirati RM-programom.

Uz pogodno kodiranje, **za svaku Turingovu mašinu**

$\mathbb{T} = (Q, q_0, F, \Sigma, \sqcup, \Gamma, \tau)$ **postoji** RM-program nad unarnim alfabetom $\{1\}$ koji simulira rad mašine $\mathbb{T}$.

Stanja mašine $\mathbb{T}$ su $q_0, \dots, q_k, q_{k+1}, \dots, q_n$, pri čemu je q_0 početno stanje, $q_{k+1}, \dots, q_n$ su završna stanja.

Simboli trake su $s_0, s_1, \dots, s_m$, pri čemu je s_0 blanko znak $\sqcup$.

Svakom stanju i svakom simbolu pridružuje se kôd: $[q_i] = i, [s_i] = i$.

Svaku konfiguraciju mašine $\mathbb{T}$, tj. reč iz $\Gamma^* Q \Gamma^+$,

$$C : u_\ell \cdots u_1 q_i v_1 \cdots v_r, \ell \geq 0, r > 0, u_j, v_k \in \Gamma,$$

kodiramo brojem $[C] = [i, \lceil [u_1], \dots, [u_\ell] \rceil, \lceil [v_1], \dots, [v_r] \rceil]$.

Svaka TM može se simulirati RM-programom.

Ako je $\bar{w}$ neki konačan niz reči ulaznog alfabeta mašine $\mathbb{T}$, početnu konfiguraciju određenu ovim ulazom označavamo $C_{\bar{w}}$.

Svaka TM može se simulirati RM-programom.

Ako je $\bar{w}$ neki konačan niz reči ulaznog alfabeta mašine $\mathbb{T}$, početnu konfiguraciju određenu ovim ulazom označavamo $C_{\bar{w}}$.

Treba sastaviti RM-program takav da za sve $k \geq 0$ i sve reči $w_1, \dots, w_k \in \Sigma^*$, važe sledeći uslovi:

- (1) ako $\mathbb{T}(w_1, \dots, w_k) \uparrow$, onda $\mathbb{P}([C_{\bar{w}}]) \uparrow$,
- (2) ako $\mathbb{T}(w_1, \dots, w_k) \downarrow$, onda $\mathbb{P}([C_{\bar{w}}]) \downarrow$; štaviše ukoliko $C_{\bar{w}} \xrightarrow{*}_{\mathbb{T}} C_{\text{stop}}$, onda je $[C_{\text{stop}}]$ sadržaj prvog registra u završnoj konfiguraciji izračunavanja programa $\mathbb{P}$ za ulaz $[C_{\bar{w}}]$.

Svaka TM može se simulirati RM-programom.

Važni registri:

C – oznaka za registar R_1 u kome se čuvaju kôdovi konfiguracija izračunavanja mašine $\mathbb{T}$ za ulaz $\bar{w}$;

U – za kôd reči levo od ćelije koja se čita, bez te ćelije;

V – za kôd reči desno od ćelije koja se čita, sa tom ćelijom;

Q – za kôd tekućeg stanja;

S – za kôd tekućeg simbola (koji čita glava);

D – za kôd pomeraja glave (vrednosti ovih registara mogu biti $[R] = 0$, $[P] = 1$, $[L] = 2$);

B – pomoćni registar.

Svaka TM može se simulirati RM-programom.

Važni registri:

C – oznaka za registar R_1 u kome se čuvaju kôdovi konfiguracija izračunavanja mašine $\mathbb{T}$ za ulaz $\bar{w}$;

U – za kôd reči levo od ćelije koja se čita, bez te ćelije;

V – za kôd reči desno od ćelije koja se čita, sa tom ćelijom;

Q – za kôd tekućeg stanja;

S – za kôd tekućeg simbola (koji čita glava);

D – za kôd pomeraja glave (vrednosti ovih registara mogu biti $[R] = 0$, $[P] = 1$, $[L] = 2$);

B – pomoćni registar.

Za svaku petorku $qss'Dq'$ programa τ , može se sastaviti RM-program $(Q, S, D) := \tau(Q, S)$ koji obavlja sledeći zadatak: ako su sadržaji registara Q i S jednaki $[q]$ i $[s]$, onda u registre Q, S, D upiši $[q']$, $[s']$ i $[D]$.

Svaka TM može se simulirati RM-programom.

