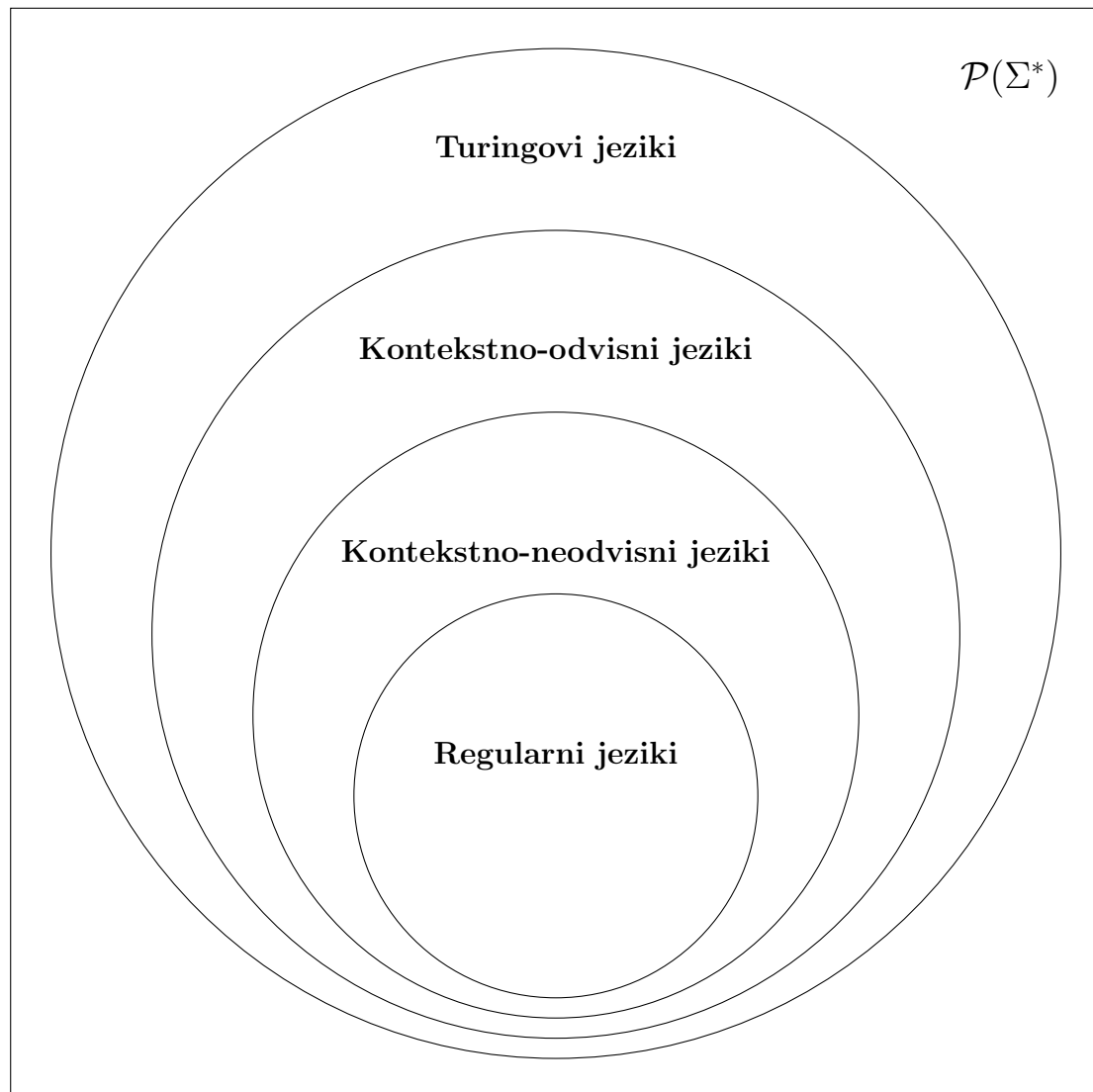


Teoretične osnove računalništva

23. marec 2011



This work is licensed under a Creative Commons
Attribution-NonCommercial-ShareAlike 3.0
Unported License

Kazalo

1	Vaja 1 (4. 3. 2011)	2
1.1	Turingovi stroji	2
1.1.1	Trenutni opis	2
2	Vaja 2 (7. 3. 2011)	4
2.1	Turingovi stroji - nadaljevanje	4
2.1.1	Relacija \vdash	6
2.2	Jezik Turingovega stroja	6
2.2.1	Ugotavljanje pripadnosti besed Turingovemu jeziku	7
2.2.2	Turingov stroj kot računalnik funkcij	7
2.2.3	Lažja konstrukcija Turingovih strojev	8

Poglavje 1

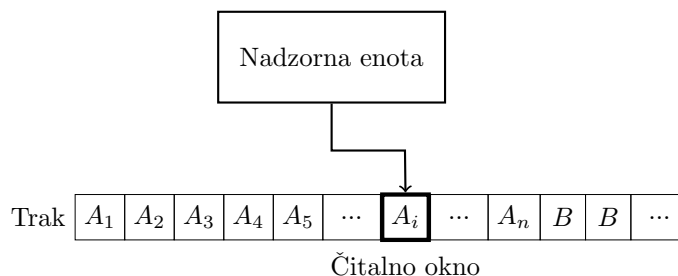
Vaja 1 (4. 3. 2011)

1.1 Turingovi stroji

Turingov stroj se je uveljavil kot uporaben in preprost model računanja, ki zna izračunati vse kar se izračunati da (pod pogojem, da Church-Turingova teza drži). Alan Turing je svoj stroj izpeljal iz razmišljanja o tem, kako človek rešuje miselne probleme na papir. Pri tem je izbral tri sestavne dele:

- Nadzorna enota (glava)
- Čitalno okno (roka in vid)
- Trak (papir)

V postopku formalizacije, pa je zaradi večje preprostosti, zahteval še, da je stroj sestavljen iz končno mnogo elementov, ter da deluje v diskretnih korakih.



Def.: Turingov stroj je definiran kot sedmerka $M = \langle Q, \Sigma, \Gamma, \delta, q_0, B, F \rangle$, kjer je:

- Q končna množica stanj
- Σ končna množica vhodnih simbolov, $Q \cap \Sigma = \emptyset$
- Γ končna množica tračnih simbolov, $\Sigma \subset \Gamma$
- δ funkcija prehodov: $Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, D\}$,
kjer L in D označujeta premik levo ali desno
- q_0 začetno stanje, $q_0 \in Q$
- B prazen simbol, $B \in \Gamma$
- F množica končnih stanj, $F \subseteq Q$

Stroj deluje tako, da v vsakem koraku opravi naslednje:

- preide v neko stanje
- zapiše nov simbol v celico, ki je pod oknom
- okno premakne eno celico levo ali desno

1.1.1 Trenutni opis

Def.: $TO = \Gamma^* \times Q \times \Gamma^*$ je množica vseh trenutnih opisov.

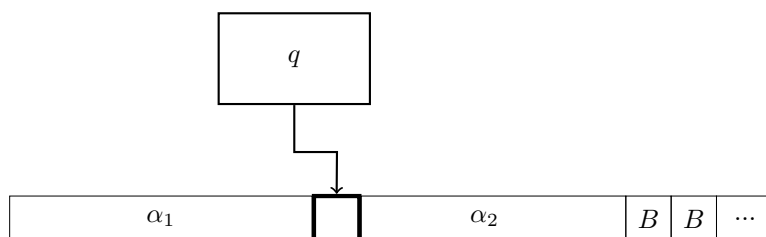
Nek trenutni opis $\langle \alpha_1, q, \alpha_2 \rangle$, ali krajše $\alpha_1 q \alpha_2$ opisuje konfiguracijo Turingovega stroja.

- naredi bolj formalen zapis!

Poglavje 2

Vaja 2 (7. 3. 2011)

2.1 Turingovi stroji - nadaljevanje



Čitalno okno je nad prvim znakom niza α_2 , iz tega lahko razberemo:

- če je $\alpha_1 = \varepsilon$, je okno skrajno levo
- če je $\alpha_2 = \varepsilon$, je okno nad B in so naprej sami B -ji

2.1.1 Relacija \vdash

Def.: Če sta u, v trenutna opisa iz množice TO , ter v neposredno sledi iz u v enem koraku Turingovega stroja, tedaj pišemo $u \vdash v$.

Naj bo $x_1 \dots x_{i-1} q x_i \dots x_n$ trenutni opis:

- če je $\delta(q, x_i) = \langle p, Y, D \rangle$:
 $x_1 \dots x_{i-1} q x_i \dots x_n \vdash x_1 \dots x_{i-1} Y p x_{i+1} \dots x_n$
- če je $\delta(q, x_i) = \langle p, Y, L \rangle$:
 - če je okno na robu ($i = 1$), se Turingov stroj ustavi, ker je trak na levi omejen.
 - če okno ni na robu ($i > 1$), potem: $x_1 \dots x_{i-2} x_{i-1} q x_i \dots x_n \vdash x_1 \dots x_{i-2} p x_{i-1} Y x_{i+1} \dots x_n$

Imamo pa tudi posplošeno relacijo $u \vdash^* v$, ki pove, da trenutni opis v sledi iz u v enem ali več korakih.

Def.: $u \vdash^* v$, če obstaja tako zaporedje $x_i, (i \in [0, 1, \dots, k], k \geq 0)$, da velja $u = x_0, v = x_k$ in $x_0 \vdash x_1 \wedge x_1 \vdash x_2 \wedge \dots \wedge x_{k-1} \vdash x_k$

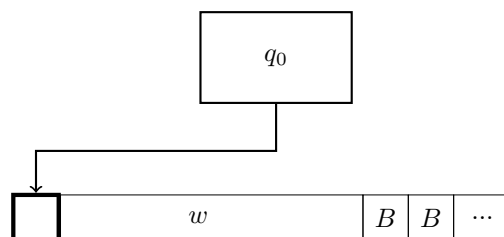
Torej, trenutni opis v sledi iz u , v k korakih Turingovega stroja.

2.2 Jezik Turingovega stroja

Def.: Jezik Turingovega stroja je:

$$L(M) = \{w \mid w \in \Sigma^* \wedge \varepsilon q_0 w \vdash^* \alpha_1 q_F \alpha_2 \wedge \alpha_1, \alpha_2 \in \Gamma^*, q_F \in F\}$$

Z besedami to pomeni, da je jezik Turingovega stroja množica besed, ki če jih damo na vhod stroja, povzročijo, da se ta v končno mnogo korakov znajde v končnem stanju.



Začetna konfiguracija Turingovega stroja.

Def.: Jezik L je Turingov jezik, če obstaja Turingov stroj M , tak, da je $L = L(M)$.

2.2.1 Ugotavljanje pripadnosti besed Turingovemu jeziku

Pri vprašanju ali je neka beseda v jeziku, Turingove jezike ločimo na:

- Odločljive - obstaja algoritem, s katerim se lahko za poljubno besedo odločimo, ali pripada jeziku.
- Neodločljive - v splošnem ni algoritma, ki bi za poljubno vhodno besedo z DA ali NE odgovoril na vprašanje pripadnosti.
 - če je odgovor DA, to ugotovimo v nekem končnem številu korakov.
 - če je odgovor NE, pa ni nujno, da se bo stroj kdaj ustavil.
- - vennov diagram odločljivi jeziki znotraj Turingovih?

Primer: Zapiši Turingov stroj, ki sprejema jezik $L = \{0^n 1^n | n \geq 1\}$

Skica izvajanja stroja:

- $0^n 1^n$ - vhodna beseda
- $X0^{n-1}1^n$ - zamenjamo najbolj levo 0 z X
- $X0^{n-1}Y1^{n-1}$ - premaknemo okno desno do najbolj leve 1 in jo zamenjamo z Y
- $XX0^{n-2}Y1^{n-1}$
- $XX0^{n-2}YY1^{n-2}$ - ponovimo in vidimo, da bomo niz sprejeli, če je prave oblike.

Turingov stroj zapišemo kot $M = \langle Q, \Sigma, \Gamma, \delta, q_0, B, F \rangle$:

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, B, X, Y\}$
- $F = \{q_4\}$
- δ bomo definirali s tabelo

Pomen stanj:

- q_0 - začetno stanje in stanje pred zamenjavo 0 z X
- q_1 - premikanje desno do 1
- q_2 - zamenjava 1 z Y in premikanje levo do X
- q_3 - najde X in se premik desno
- q_4 - končno stanje

Tabela prehajanja stanj:

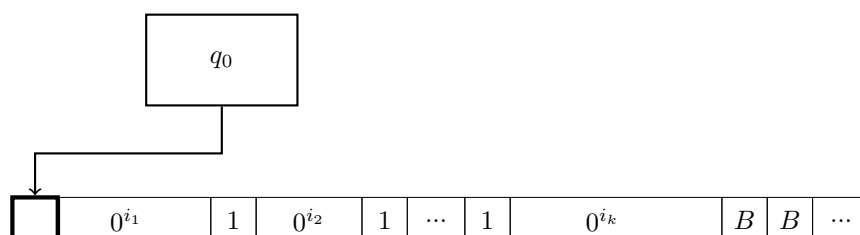
	0	1	B	X	Y
x_0	$\langle q_1, X, D \rangle$	–	–	$\langle q_3, Y, D \rangle$	–
x_1	$\langle q_1, 0, D \rangle$	$\langle q_2, Y, L \rangle$	–	$\langle q_1, Y, D \rangle$	–
x_2	$\langle q_2, 0, D \rangle$	–	$\langle q_0, X, D \rangle$	$\langle q_2, Y, L \rangle$	–
x_3	–	–	–	$\langle q_3, Y, D \rangle$	$\langle q_4, B, D \rangle$
x_4	–	–	–	–	–

Izvajanje stroja s trenutnimi opisi:

$$q_0 0011 \vdash Xq_1 011 \vdash X0q_1 11 \vdash Xq_2 0Y1 \vdash \dots$$

2.2.2 Turingov stroj kot računalnik funkcij

Imamo Turingov stroj, ki ima na traku neko število ničel, ki predstavljajo pozitivna naravna števila, ločena z enicami:



Recimo, da se stroj po nekem številu korakov ustavi in ima na traku skupino ničel 0^m , na levi in desni strani skupine pa same B -je. S tem je stroj morda izračunal neko funkcijo:

$$f^{(k)} : \mathbb{N}_+^k \rightarrow \mathbb{N}_+ \quad \text{oz.} \quad f(i_1, i_2, \dots, i_k) = m$$

Funkcija f ni nujno definirana za vsako k -terico iz \mathbb{N}_+^k , torej je parcialna funkcija, kadar pa je definirana povsod, pravimo da je totalna. Stroj se pri nedefiniranih k -tericah pač na neki točki ustavi in pri tem na traku ne pusti le ene skupine ničel, ali pa se sploh ne ustavi. Isti turingov stroj hkrati računa več funkcij: $f^{(1)}, f^{(2)}, \dots, f^{(k)}$.

Parcialna rekurzivna funkcija

Def.: Vsaka funkcija $f^{(k)} : \mathbb{N}_+^k \rightarrow \mathbb{N}$, ki jo lahko izračuna nek Turingov stroj, je parcialna rekurzivna funkcija. Če je $f^{(k)}$ definirana za vse k -terice, jo imenujemo totalna rekurzivna funkcija (včasih samo rekurzivna funkcija)

Vse običajne aritmetične funkcije so parcialne ali celo totalne rekurzivne funkcije. V primerih si bomo pogledali nekaj primerov, tu pa jih nekaj naštejmo: $m + n$, $m * n$, $n!$, 2^n , $\lceil \log(n) \rceil$, m^n , \dots

Primeri:

Primer 1: Ali je $f(m, n) = m + n$ (parcialno) rekurzivna?

Skica stroja, ki računa $m + n$:

- $0^m 10^m$ - vhodna beseda
- $B0^{m-1}10^m$ - izbriši prvo ničlo
- $B0^{m+n}$ - premakni se do 1 in jo zamenjaj z 0

Primer 2: Ali je $f(m, n) = m * n$ (parcialno) rekurzivna?

Skica stroja, ki računa $m * n$:

- $0^m 10^n$ - vhodna beseda
- $0^m 10^n 1$ - premakni se na konec in zapiši 1 (ločnica za rezultat)
- $B0^{m-1}10^n 1$ - premakni se na začetek in izbriši 0
- $B0^{m-1}10^m 10^n$ - prekopiraj n ničel za ločnico (in ničle)
- $B^m 10^m 10^{m*n}$ - ponavljaj tadva koraka, dokler ni več ničel pred prvo 1
- $B^{m+n+2}0^{m*n}$ - izbriši del, ki ne spada v rezultat

2.2.3 Lažja konstrukcija Turingovih strojev

Obstaja nekaj tehnik, ki poenostavijo in pohitijo sestavljanje Turingovih strojev.

Nadzorna enota kot pomnilnik

Vsako stanje stroja, je sestavljeno iz dveh delov – stanja avtomata, ter shrambe za tračne znake. Novo množico stanj zapišemo kot $Q = K \times \Gamma$, kjer je K stara množica stanj in Γ tračna abeceda.

Primer: Sestavi Turingov stroj za razpoznavanje besed, pri katerih se prvi znak ne ponovi:

Stroj $M = \langle Q, \Sigma, \Gamma, \delta, q_0, B, F \rangle$ zapišemo kot:

- $M = \langle Q, \{0, 1\}, \{0, 1, B\}, \delta, \langle q_0, B \rangle, B, F \rangle$
- $Q = \{q_0, q_1\} \times \{0, 1, B\} = \{\langle q_0, 0 \rangle, \langle q_0, 1 \rangle, \langle q_0, B \rangle, \langle q_1, 0 \rangle, \langle q_1, 1 \rangle, \langle q_1, B \rangle\}$
- $F = \{\langle q_1, B \rangle\}$
- δ zapišemo kot:
 - Shrani prvi znak besede v stanje stroja:

$$\delta(\langle q_0, B \rangle, 0) = \langle \langle q_1, 0 \rangle, 0, D \rangle$$

$$\delta(\langle q_0, B \rangle, 1) = \langle \langle q_1, 1 \rangle, 1, D \rangle$$
 - Premakni okno v desno do prvega znaka, enakega shranjenemu:

$$\delta(\langle q_1, 0 \rangle, 1) = \langle \langle q_1, 0 \rangle, 1, D \rangle$$

$$\delta(\langle q_1, 1 \rangle, 0) = \langle \langle q_1, 1 \rangle, 0, D \rangle$$

- Če prebereš B , pojdi v končno stanje:
 $\delta(\langle q_1, 0 \rangle, B) = \langle \langle q_1, B \rangle, karkoli \rangle$
 $\delta(\langle q_1, 1 \rangle, B) = \langle \langle q_1, B \rangle, karkoli \rangle$
- Sicer se ustavi. To dosežemo tako, da ne definiramo prehodov:
 $\delta(\langle q_1, 0 \rangle, 0)$ in $\delta(\langle q_1, 1 \rangle, 1)$

Večsledni trak

Na traku imamo več kot eno sled, kar pomeni, da s traku beremo k -terice tračnih znakov, kar zapišemo kot: $\Gamma = \Gamma_1 \times \Gamma_2 \times \dots \times \Gamma_k$.

Primer: Sestavi Turingov stroj, ki preveri, ali je vhodno število praštevilo.

Skica stroja:

- Trak ima tri sledi:
 - na prvi sledi je vhodno število
 - na drugi sledi je števec, ki na začetku hrani število 2
 - tretjo sled uporabimo za delovno sled, na začetku je lahko prazna.
- Stroj deluje tako:
 - prepriši število s prve sledi na tretjo sled
 - odštevaj število iz druge sledi od števila na tretji sledi
 - če se odštevanje konča z 0, se ustavi (ni praštevilo)
 - sicer število na drugi sledi povečaj za 1
 - če je število na drugi sledi enako tistemu na prvi, sprejmemo (je praštevilo)
 - sicer, ponovimo postopek

Prestavljanje vsebine traku

Recimo, da bi s traku radi vzeli nekaj zaporednih znakov tako, kot da bi jih izrezali iz traku in nato trak zlepili nazaj skupaj, izrezane simbole pa bi si pri tem seveda radi nekako zapomnili. Tudi to metodo realiziramo s pomočjo shrambe za tračne simbole v nadzorni enoti, a moramo pri tem paziti, da je funkcija prehodov pravilno napisana.

- slika "gube" na traku in slika nadzorne enote

Primer: Sestavi Turingov stroj, ki premakne vsebino traku za 2 celici v desno.

Skica stroja:

- Q vsebuje stanja oblike: $\langle q, A_1, A_2 \rangle$; $q \in \{q_1, q_2\}$, $A_1, A_2 \in \Gamma$
- Γ poleg ostalih znakov, vsebuje še poseben znak X , ki označuje izpraznjeno celico na traku
- $F = \{q_2\}$
- δ zapišemo kot:
 - Prva koraka – zapomni si in izprazni prvi in drugi znak:
 $\delta(\langle q_1, B, B \rangle, A_1) = \langle \langle q_1, B, A_1 \rangle, X, D \rangle$
 $\delta(\langle q_1, B, A_1 \rangle, A_2) = \langle \langle q_1, A_1, A_2 \rangle, X, D \rangle$
 - Zapomni si nov znak in prvega iz shrambe zapiši na trak:
 $\delta(\langle q_1, A_i, A_{i+1} \rangle, A_{i+2}) = \langle \langle q_1, A_{i+1}, A_{i+2} \rangle, A_i, D \rangle$
 - Zadnja koraka – zapiši vsebino shrambe na trak:
 $\delta(\langle q_1, A_{n-1}, A_n \rangle, B) = \langle \langle q_1, A_n, B \rangle, A_{n-1}, D \rangle$
 $\delta(\langle q_1, A_n, B \rangle, B) = \langle \langle q_2, B, B \rangle, A_n, L \rangle$

Podprogrami

-