

IV dalis

Turing'o mašina

Šita kurso dalis yra iš dalies paruošta pagal dr. Valdo Dičiūno diskrečiosios matematikos paskaitų konspektus. Autorius norėtų jam padėkoti už suteiktą galimybę pasinaudoti išėities tektais ruošiant šiuos konspektus.

“Algoritmas” yra pirminė matematikos bei informatikos sąvoka, panašiai kaip sąvokos “skaičius”, “aibė” ir kitos. Neformaliai (intuityviai) algoritmas suprantamas kaip baigtinis taisyklių arba komandų rinkinys (dar vadinamas programa), nusakantis, kaip vienus objektus (duomenis), atlikus baigtinį skaičių nesudėtingų operacijų, perdirbti į kitus objektus (rezultatus).

Norint formaliai nagrinėti algoritmus kaip matematinius objektus, buvo pasiūlyti įvairūs skaičiavimo modeliai: Turing'o mašinos (beje, apibrėžtos gerokai anksčiau už realių kompiuterių atsiradimą), RAM (tiesioginės kreipties į atmintį) mašinos, dalinės rekursyvosios funkcijos, Markovo algoritmai. Šioje dalyje mes nagrinėsime paprasčiausią Turing'o mašinų variantą — vienajuostę determinuotą Turing'o mašiną.

Apibrėžimas. Turing'o mašina (*arba* 1-juoste determinuota Turing'o mašina) *vadiname rinkinį*

$$M = \langle \Sigma, Q, \delta, q_0, F \rangle,$$

kur:

- $\Sigma = \{a_1, \dots, a_n\}$ yra *baigtinė aibė*, vadinama abėcėle;
- $Q = \{q_0, q_1, \dots, q_k\}$ yra *baigtinė būsenų aibė*;
- $\delta: Q \times \Sigma \rightarrow Q \times \Sigma \times \{K, N, D\}$ yra *dalinė (ne visur apibrėžta) pėrėjimų funkcija*;
- $q_0 \in Q$ – pradinė būsena; *ir*
- $F \subseteq Q$ yra galutinių būsenų aibė.

Determinuota Turing'o mašina interpretuojama kaip mašina, turinti begalinę juostą, suskirstytą į ląsteles, ir palei juostą slenkančią skaitymo bei rašymo galvutę, kuri fiksuotu laiko momentu mato vieną juostos ląstelę ir gali perskaityti, koks abėcėlės Σ ženklas ten įrašytas, vietoje jo įrašyti kitą ženklą bei pasislinkti per vieną ląstelę į kairę arba į dešinę, arba likti stovėti toje pat vietoje. Aibė $\{K, N, D\}$ yra galimų galvutės postūmių aibė, kur K atitinka postūmį į kairę, D atitinka postūmį į dešinę, ir N reiškia, kad galvutė niekur nejuda.

Vienas iš Turing'o mašinos abėcėlės ženklų vadinamas *tuščiu* (“blank”) ir žymimas b ($b \in \Sigma$). Jis yra skirtas tuščioms ląstelėms žymėti ir negali būti naudojamas pradinių duomenų bei tarpinių rezultatų, su kuriais dirba Turing'o mašina, kodavimui. Duomenys tarp savęs atskiriami vienu ar daugiau tuščių ženklų ar kitais sutartiniais abėcėlės ženklais. Dešiniau nuo paskutinių duomenų visa juosta vėl užpildyta tuščiais ženklais. Taigi, kiekvienu laiko momentu tik baigtinė juostos sritis gali turėti netuščias ląsteles (kadangi po baigtinio žingsnių skaičiaus galvutė bus pasiekusi tik baigtinį ląstelių skaičių į kairę ir į dešinę nuo pradinės ląstelės). Ši netuščia sritis (pradedant pirmąja iš kairės netuščia ląstele ir baigiant paskutine iš dešinės netuščia ląstele) dar yra vadinama *reikšmine juostos dalimi*.

Pradiniu laiko momentu mašina visada yra pradinėje būsenoje q_0 ir mato pirmą iš kairės netuščią ląstelę. Darbo eigoje mašina keičia būsenas, nuskaito ir įrašo simbolius į juostą ir juda palei juostą. Ką

jinai turi daryti, nurodo perėjimų funkcijos δ reikšmės. Tarkime, kažkuriuo laiko momentu mašina yra būsenoje $q_{i_1} \in Q$, o jos skaitymo ir rašymo galvutė mato simbolį $a_{j_1} \in \Sigma$. Tada mašina elgiasi pagal tai, kokia yra reikšmė $\delta(q_{i_1}, a_{j_1})$. Tarkime, $\delta(q_{i_1}, a_{j_1}) = (q_{i_2}, a_{j_2}, J)$, kur $q_{i_2} \in Q$, $a_{j_2} \in \Sigma$ ir $J \in \{K, N, D\}$. Tai reiškia, kad mašina turi pereiti į būseną q_{i_2} , vietoje simbolio a_{j_1} juostoje užrašyti a_{j_2} , ir pasislinkti į kurią nors pusę per vieną poziciją priklausomai nuo J reikšmės (arba nejudėti, jei $J = N$). Užrašas $\delta(q_{i_1}, a_{j_1}) = (q_{i_2}, a_{j_2}, J)$ vadinamas *komanda* (arba *instrukcija*).

Mašina sustoja, kai pereina į kurią nors galutinę būseną, t.y. į būseną iš aibės F . Mašinai sustojus jos *darbo rezultatu* vadinsime žodį, užrašytą netuščiose ląstelėse. Jei mašina visai nesustoja, jos darbo rezultata laikome *neapibrėžtu*.

Turing'o mašina perdirba žodžius abėcėlėje $\Sigma \setminus \{b\}$ į kitus žodžius toje pat abėcėlėje. Kiekvienu laiko momentu Turing'o mašiną galima visiškai apibrėžti nurodant jos būseną, galvutės vietą ir reikšminės juostos dalies turinį. Tokį momentinį Turing'o mašinos aprašą dar vadina *konfigūracija*. Taigi, Turing'o mašinos konfigūracija yra žodis pavidalo

$$K = a_{j_1} \dots a_{j_{l-1}} q_{i_l} a_{j_l} \dots a_{j_m},$$

kuris reiškia, kad duotu momentu juostos reikšminėje dalyje stovi žodis $a_{j_1} \dots a_{j_{l-1}} a_{j_l} \dots a_{j_m}$, mašina yra būsenoje q_{i_l} , o jos skaitymo ir rašymo galvutė mato ženklą a_{j_l} . Tokiu atveju mašina ieško komandos, kurios kairėje pusėje stovi pora (q_{i_l}, a_{j_l}) . Tarkime, mašina rado tokią komandą pavidalo $\delta(q_{i_l}, a_{j_l}) = (q_{i_2}, a_{j_2}, J)$. Tarkime, nagrinėjamoje komandoje judesys $J = D$. Įvykdžius šią komandą mašinos konfigūracija pasikeičia iš konfigūracijos K į naują konfigūraciją

$$K' = a_{j_1} \dots a_{j_{l-1}} a_{j_l} q_{i_2} a_{j_{l+1}} \dots a_{j_m}.$$

Jei nauja mašinos būseną q_{i_2} yra iš aibės F , mašina sustoja. Jei ne, ieško kitos komandos.

Taigi, Turing'o mašinos darbą nuo pradinio momento iki sustojimo (tarkime, po s žingsnių) galima visiškai aprašyti nurodant konfigūracijų seką $K_0 \rightarrow K_1 \rightarrow \dots \rightarrow K_s$, kur K_0 yra pradinė konfigūracija, o K_s yra galutinė konfigūracija po sustojimo.

Tarkime, duota Turing'o mašina M ir žodis (t.y. vektorius) u virš $\Sigma \setminus \{b\}$. Mašinos M darbo rezultata, jai pradėjus savo darbą iš pradinės konfigūracijos $K_0 = q_0 u$, žymėsime $M(u)$. Jei $M(u)$ neapibrėžtas, žymėsime ∞ .

Toliau naudosime abėcėlę $\Sigma = \{0, 1, b\}$ ir apibrėšime, kaip Turing'o mašina skaičiuoja dalines funkcijas $f: \mathbb{N}^n \rightarrow \mathbb{N}$. Iš natūraliųjų skaičių sudarytą vektorių $\mathbf{u} = (u_1, \dots, u_n)$ juostoje vaizduosime žodžiu \tilde{u} , sudarytu iš skaičių u_1, \dots, u_n dvejetainių išraiškų, atskirtų tuščiais ženklais b . Sakysime, kad mašina M *skaičiuoja funkciją* $f: \mathbb{N}^n \rightarrow \mathbb{N}$, jei $\forall \mathbf{u} \in \mathbb{N}^n$:

- (1) $f(\mathbf{u})$ reikšmė apibrėžta tada ir tik tada, kai $M(\tilde{u})$ apibrėžtas, ir
- (2) jei $f(\mathbf{u})$ apibrėžta, tai $M(\tilde{u})$ yra $f(\mathbf{u})$ dvejetainė išraiška.

Dalinę funkciją $f: \mathbb{N}^n \rightarrow \mathbb{N}$ vadiname *apskaičiuojama (pagal Turing'ą)*, jei egzistuoja Turing'o mašina M , skaičiuojanti f .

Panagrinėkime porą Turing'o mašinos pavyzdžių.

Pavyzdys. Parodysime, kad funkcija

$$f(x) = \begin{cases} 1, & x \text{ lyginis,} \\ 0, & \text{priešingu atveju} \end{cases}$$

yra apskaičiuojama pagal Turing'ą. Nesunku įsitikinti, kad šią funkciją skaičiuos tokia mašina:

$$\begin{aligned}\delta(q_0, 0) &= (q_0, 0, D) \\ \delta(q_0, 1) &= (q_0, 1, D) \\ \delta(q_0, b) &= (q_1, b, K) \\ \delta(q_1, 0) &= (q_2, 1, K) \\ \delta(q_1, 1) &= (q_2, 0, K) \\ \delta(q_2, 0) &= (q_2, b, K) \\ \delta(q_2, 1) &= (q_2, b, K) \\ \delta(q_2, b) &= (q_3, b, N)\end{aligned}$$

kurios galutinių būsenų aibė yra $F = \{q_3\}$.

Patogiausia komandas užrašyti lentelė, kurioje stulpeliai sunumeruoti abėcėlės Σ simboliais, o eilutės — būsenomis:

	0	1	b
q_0	$q_0, 0, D$	$q_0, 1, D$	q_1, b, K
q_1	$q_2, 1, K$	$q_2, 0, K$	—
q_2	q_2, b, K	q_2, b, K	q_3, b, N

Iš tikro, būsenoje q_0 galvutė tiesiog važiuoja į dešinę, kol pasiekia žodžio x pabaigą, t.y. nuskaito b. Nuskaičiusi b, galvutė pasislenka per vieną poziciją į kairę, t.y. atsistoja ties paskutiniu netuščiu simboliu, ir pakeičia būseną į q_1 . Būdama būsenoje q_1 mašina patikrina, ar tas paskutinis simbolis yra 0, ar 1. Jei 0, tai skaičius x yra lyginis, todėl atsakymas turėtų būti 1. Užrašome jį ir pereiname į būseną q_2 . Analogiškai darome, jei paskutinis simbolis yra 1. Būdama būsenoje q_2 , galvutė viską trina, važiuodama į kairę. Ištrynusi viską, t.y. radusi b, pereina į galutinę būseną, t.y. baigia darbą. Juostoje lieka užrašytas tik atsakymas — 0 ar 1.

Beje, šio pavyzdžio funkciją apskaičiuoti galima ir kitu būdu: iš karto viską triname, važiuodami į dešinę ir įsimindami paskutinio nuskaityto simbolio reikšmę. Kaip įsimename? Ogi keisdami būseną. Pavyzdžiui, jei nuskaitome 0, keičiame būseną į q_1 , o jei 1, tai į q_2 . Kai viską ištrynę pasiekiamo žodžio pabaigą, užrašome atsakymą priklausomai nuo to, kurioje būsenoje esame (t.y. priklausomai nuo to, koks buvo paskutinis nuskaitytas simbolis, o tuo pačiu ir paskutinis žodžio simbolis), ir baigiame darbą. Tai daro tokia Turing'o mašina:

	0	1	b
q_0	q_0, b, D	q_1, b, D	$q_2, 1, N$
q_1	q_0, b, D	q_1, b, D	$q_2, 0, N$

kurios galutinių būsenų aibė yra $F = \{q_2\}$. □

Pavyzdys. Sudarykime Turing'o mašiną, skaičiuojančią funkciją $f(x) = x + 1$, čia x — natūralusis skaičius, užrašytas dvejetainiu pavidalu.

Kaip ta mašina turėtų veikti? Kad būtų lengviau įsivaizduoti, imkime pavyzdį. Tegų $x = 1100111$. Tada $x + 1 = 1101000$. Matome, kad funkcijos f skaičiavimo algoritmas galėtų būti toks: visus dešinėje skaičiaus pusėje esančius vienetus verčiame nuliais, pirmą iš dešinės nulį verčiame vienetu, o kitkas lieka nepakitę. Toks būtų bendras algoritmas. Bet nepamirškime, kad būna išskirtinių atvejų, kai bendras algoritmas netinka. Šiuo atveju bendras algoritmas netinka, jei nėra jokio pirmo iš dešinės nulio, t.y. jei skaičius sudarytas vien iš vienetų, pavyzdžiui, $x = 1111$. Tokiu atveju atsakymas turėtų būti $x + 1 = 10000$. Taigi, vienetu verčiame pirmą iš dešinės nulį, o jei tokio nėra, tai pirmą tuščią simbolį. Apibendrinkim: galvutė iš pradžių turi nuvažiuoti iki skaičiaus pabaigos (nes pradžioje ji stovi ties pirmu

iš kairės netuščiu simboliu) — tai vyksta būsenoje q_0 , o tada, važiuodama iš dešinės į kairę, visus vienetų verčia nuliais, o pirmą sutiktą nulį arba tuščią simbolį verčia vienetu ir sustoja. Tai daro tokia mašina: $F = \{q_2\}$,

	0	1	b
q_0	$q_0, 0, D$	$q_0, 1, D$	q_1, b, K
q_1	$q_2, 1, N$	$q_1, 0, K$	$q_2, 1, N$