



Појам алгоритма

17

- ▶ Алгоритам (ефективни поступак, ефективна процедура) представља један од најзначајнијих математичких појмова.
- ▶ Присутан у математици још од самог њеног настанка.
- ▶ Реч алгоритам долази од латинизираног имена арапског математичара Ал-Хорезмија који је у IX веку дао правила за извођење четири рачунске операције над децималним записима реалних бројева.
- ▶ Већ вековима се појам алгоритма у математици користи у свом неформалном, интуитивном смислу. Интуитивно, алгоритам је коначан скуп строго формулисаних правила за решавање неке класе задатака.

Појам алгоритма

Најпре дајемо описне дефиниције појмова које ћемо касније строго дефинисати.


Алгоритам се неформално може представити на следећи начин:

улаз \longrightarrow израчунавање \longrightarrow излаз

- ▶ Израчунавање је процес којим се из неког почетног скупа података, применом коначног фиксираних правила, добија крајњи резултат.
- ▶ Улаз је почетни низ података на којима треба да се изврши израчунавање.
- ▶ Фиксирани скуп правила је програм.
- ▶ Излаз је резултат израчунавања.

Ако се алгоритмом израчунавају вредности неке функције, онда се за ту функцију каже да је ефективно израчунљива или алгоритамски израчунљива, или само израчунљива.

Пример.

- 
- ▶ Функција $NZD(m, n)$, $m, n \in \mathbb{N}$ (највећи заједнички делилац два природна броја) је израчунљива.
 - ▶ Функција $f : \mathbb{N} \rightarrow \mathbb{N}$ дата са

$$f(n) = \begin{cases} 1, & \text{ако постоји } n \text{ узастопних цифара } 5 \text{ у запису } \sqrt{2} \\ 0, & \text{иначе} \end{cases}$$

је добро дефинисана, али није познат поступак који би за свако $n \in \mathbb{N}$ одређивао постоји ли n узастопних петица у децималном запису броја $\sqrt{2}$.


Особине алгоритама

Следећи општи услови се прихватају као критеријуми за називање неког поступка алгоритмом:

- (1) Сваки алгоритам је коначан низ инструкција.
- (2) Постоји рачунско средство које интерпретира и изводи инструкције алгоритама.
- (3) Израчунавање према датом алгоритму је дискретне природе, дакле изводи се корак по корак и без коришћења непрекидних метода.
- (4) Постоји меморијски простор у којем се чувају, привремено или стално, сви подаци који се јављају приликом израчунавања.
- (5) Израчунавање према датом алгоритму је детерминисано, тј. изводи се без коришћења случајних метода или средстава. Дакле, поновљене примене алгоритама на исте улазне величине производе исте излазне величине.


- (6) Нема ограничења на величину улазних података, број инструкција, величину меморије, као ни дужину рачуна који се изводи за конкретне улазне податке.
- (7) Алгоритам не мора произвести резултат за све улазе. Дакле, могуће је да се за одређене улазне податке израчунавње према неком алгоритму никада не заврши.
- ~~(8)~~ Постоји универзалан алгоритам који симулира израчунавање по сваком другом алгоритму.
- (9) Алгоритама и објеката на којима се алгоритми изводе има пребројиво много, али не и више од тога.
- ~~(10)~~ Алгоритми, улазни и излазни симболи могу се ефективно кодирати у скупу природних бројева.

Примери алгоритамски решивих проблема:

- 
1. Сабирање и множење природних бројева.
 2. Одређивање највећег заједничког делиоца два природна броја (Еуклидов алгоритам).
 3. Диференцирање (налажење првог извода) елементарних функција.
 4. Решавање помоћу радикала алгебарских једначина другог, трећег и четвртог степена.
 5. Решавање система линеарних једначина (Гаусов поступак).
 6. Утврђивање таутологичности исказних формула.

Постоје и математички проблеми за које је доказано да нису алгоритамски решиви. Нити за један од следећих примера не постоји универзалан и ефективан поступак којим би се решио произвољан задатак из наведене класе.

K17

- 
1. Решавање алгебарских диофантовских једначина (Десети Хилбертов проблем).
 2. Утврђивање истинитости у структури природних бројева \mathbb{N} аритметичких исказа представљених у предикатском рачуну првог реда (Други Хилбертов проблем)
 3. Утврђивање да ли се произвољни програм, написан нпр. у програмском језику C , за произвољне улазне податке зауставља после коначно много корака (Халтинг проблем).

Формални алгоритамски системи

18

Да би се одговорило на питање "да ли је неки математички проблем алгоритамски решив" потребно је формално прецизирати појмове алгоритма и израчунљивости. С обзиром на особину (10) алгоритама, довољно је појам ефективне израчунљивости дефинисати и разматрати на скупу аритметичких функција.

Дефиниција.

- ▶ Сваку функцију $f : D \rightarrow \mathbb{N}$, $D \subseteq \mathbb{N}^k$, $k > 0$, зваћемо аритметичком функцијом (дужине k).
- ▶ Ако је домен аритметичке функције f цео скуп \mathbb{N}^k , тј. $f : \mathbb{N}^k \rightarrow \mathbb{N}$, рећи ћемо да је функција f тотална. У супротном, функција f је парцијална аритметичка функција.

Напомена. Сматраћемо да 0 припада скупу природних бројева, тј. $\mathbb{N} = \{0, 1, 2, \dots\}$.

Нпр. сабирање и множење природних бројева су тоталне аритметичке функције дужине 2, док су одузимање и дељење природних бројева парцијалне аритметичке функције дужине 2. Алгоритамски систем је формални систем S у оквиру којег се математичким средствима дефинише појам ефективне израчунљивости, односно појам алгоритма. Системом S дефинише се подскуп F_S скупа свих аритметичких функција.

- ▶ За функције из F_S кажемо да су S -израчунљиве (или израчунљиве у систему S).
- ▶ Ако аритметичка функција f не припада F_S , кажемо да f није S -израчунљива.
- ▶ Ако се неком математичком задатку M може некако придружити S -израчунљива функција која тај задатак решава, кажемо да је задатак алгоритамски решив у S . Ако то није могуће, кажемо да је M алгоритамски нерешив у систему S .

Примери формалних алгоритамских система



Прве формализације појма алгоритма јављају се тридесетих година 20.-ог века, пре појаве првих дигиталних рачунара.

Примери алгоритамских система:

- ▶ систем рекурзивних функције (Gödel, Kleene [1936,1943]);
- ▶ Турингове машине (A.Turing [1936]);
- ▶ Черчов λ -рачун (Church [1936]);
- ▶ Постови системи (Post [1936,1943]);
- ▶ нормални алгоритми Маркова (Markov [1954]);
- ▶ УРМ: систем неограничених регистарских машина (Shepherdson and Sturgis [1963]).

Сваки од ових система има особине (1)-(10) и сваки од њих дефинише једну класу аритметичких функција.

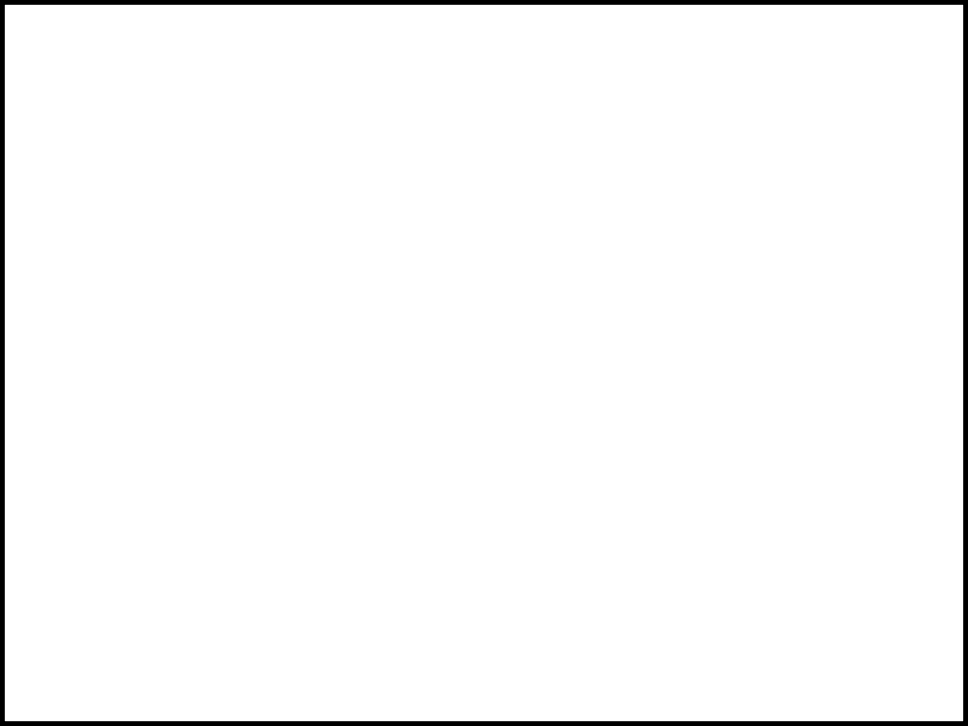


Каква је веза између класа израчунљивих функција F_S и $F_{S'}$ два различита система S и S' ?

Да ли алгоритамска решивост задатка M зависи од избора алгоритамског система, тј. да ли је могуће да је задатак M решив у алгоритамском систему S , али не и у систему S' ?

Иако наизглед различити, сви ови системи одређују исти скуп израчунљивих функција, тј. једну те исту класу алгоритама ($F_S = F_{S'}$), што наводи на тезу да ти системи израчунавања управо одређују границе могућности механичког израчунавања.









Идеални рачунар

19

Сада ћемо дати описну дефиницију алгоритамског система ИР - идеални рачунар (или УРМ "*unlimited register machine*" -неограничена регистарска машина).

Сагласно неформалној дефиницији алгоритма, идеални рачунар ИР има следеће особине:

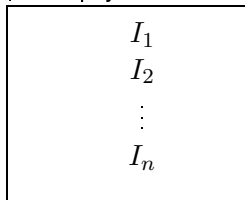
- ▶ нема никаквих ограничења на величину меморијског простора,
- ▶ нема никаквих ограничења на величину улазних података
- ▶ програми су коначни, односно сва израчунавања се изводе у коначно много корака,
- ▶ број улазних података је коначан, а излаз је само један,
- ▶ улазни и излазни подаци су искључиво природни бројеви.

Дефиниција. Идеални рачунар (краће ИР) има следеће делове:

- ▶ неограничен низ регистара R_1, R_2, R_3, \dots који у сваком тренутку садрже природне бројеве r_1, r_2, r_3, \dots . Ови бројеви се могу мењати током процеса израчунавања.

R_1	R_2	R_3	\dots
r_1	r_2	r_3	\dots

- ▶ простор за програм - простор који чува коначан низ инструкција $P = (I_1, I_2, \dots, I_n)$ који зовемо програм. Идеални рачунар може да интерпретира и изводи инструкције. Инструкције су нумерисане природним бројевима.



- ▶ бројач који у сваком тренутку садржи природни број k - редни број инструкције коју ИР треба да изврши у том тренутку.

к

ИР ради на следећи начин:

- ▶ уноси у бројач 1 и почиње да извршава инструкције унетог програма
- ▶ ИР извршава ону инструкцију чији је редни број уписан у бројачу у том тренутку
- ▶ са завршетком извршавања неке инструкције, у бројач се уписује редни број следеће инструкције
- ▶ ИР престаје са радом када је у бројачу уписан број већи од броја инструкција које чине програм; ако се то не догоди ИР ради заувек.

Инструкције које препознаје идеални рачунар

ИР препознаје и извршава следећа четири типа инструкција:

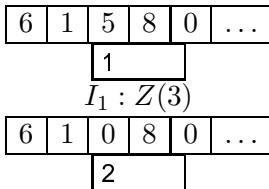
НУЛА ИНСТРУКЦИЈА Ако је

$$\underline{I_k : Z(n)}, \quad (n \geq 1)$$

ИР је извршава тако што у регистар R_n уноси 0, остали регистри су непромењени, а у бројач уписује број $k + 1$.

Пишемо $0 \rightarrow R_n$ или $r_n = 0$.

Пример.



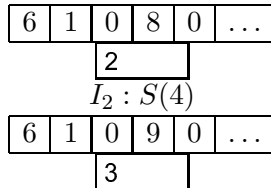
ИНСТРУКЦИЈА СЛЕДБЕНИКА Ако је

$$\underline{I_k : S(n)}, \quad (n \geq 1)$$

ИР је извршава тако што садржај регистра R_n увећава за 1, остали регистри су непромењени, а у бројач уписује број $k + 1$.

Пишемо $r_n + 1 \rightarrow R_n$ или $r_n := r_n + 1$

Пример.



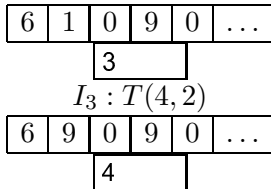
ИНСТРУКЦИЈА ПРЕНОСА Ако је

$$\underline{I_k : T(m, n)}, \quad (m \geq 1, n \geq 1)$$

ИР је извршава тако што у регистар R_n уноси садржај регистра R_m , остали регистри су непромењени, а у бројач уписује $k + 1$.

Пишемо: $r_m \rightarrow R_n$ или $r_n := r_m$

Пример.



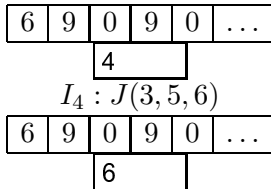
ИНСТРУКЦИЈА ПРЕЛАЗА Ако је

K 19

$$\underline{I_k : J(m, n, p)}, \quad (m \geq 1, n \geq 1, p \geq 1)$$

ИР је извршава тако што садржаји регистара остају непромењени, а у бројач се уписује p ако су садржаји регистара R_m и R_n једнаки, а уписује $k + 1$, ако су садржаји R_n и R_m различити.

Пример.



20

Конвергенција и дивергенција програма

Нека је $P = (I_1, \dots, I_s)$ неки програм и $n \geq 1$ природни број.

Дефиниција. Бесконачни низ природних бројева a_1, a_2, \dots који се налази у регистрима R_1, R_2, \dots непосредно пре стартовања програма се назива почетна конфигурација.

R_1	R_2	R_3	\dots	R_n	\dots	почетна конфигурација
a_1	a_2	a_3	\dots	a_n	\dots	

Како се свако израчунавање извршава над коначним низом бројева, без губитка општости можемо претпоставити да је у свакој почетној конфигурацији само коначно много a_i различито од нуле, тј. да је свака почетна конфигурација облика $a_1, a_2, \dots, a_n, 0, 0, \dots$.

Са $P(a_1, a_2, \dots, a_n)$ означимо израчунавање по програму P за почетну конфигурацију $a_1, a_2, \dots, a_n, 0, \dots$.

Низ природних бројева у регистрима након завршетка програма чини завршну конфигурацију.

Дефиниција.

- ▶ Ако ИП заврши израчунавање $P(a_1, \dots, a_n)$ и у завршној конфигурацији у регистру R_1 је уписан број b , тада кажемо да програм P конвергира за почетну конфигурацију (a_1, \dots, a_n) ка излазу b и пишемо $P(a_1, \dots, a_n) \downarrow b$.
- ▶ Уколико се ИП никада не заустави при израчунавању $P(a_1, \dots, a_n)$, кажемо да програм P дивергира за почетну конфигурацију (a_1, \dots, a_n) и пишемо $P(a_1, \dots, a_n) \uparrow$.

Пример. Нека је дат програм:

$I_1 : J(3, 2, 5)$

$I_2 : S(1)$

$I_3 : S(3)$

$I_4 : J(1, 1, 1)$



1 < 20

Тада:

- (a)
 - ▶ $P(3, 2) \downarrow 5,$
 - ▶ $P(5, 2) \downarrow 7,$
 - ▶ уопште $P(x, y) \downarrow x + y.$
- (б)
 - ▶ $P(3, 2, 1) \downarrow 4,$
 - ▶ $P(5, 2, 1) \downarrow 6,$ али
 - ▶ $P(3, 2, 3) \uparrow.$
 - ▶ Уопште $P(x, y, z) \downarrow x + y - z,$ за $z \leq y,$
 - ▶ $P(x, y, z) \uparrow,$ за $z > y.$

Израчунљиве функције

21

Нека је P програм и $n \geq 1$ природни број.

- ▶ Програм P и број n одређују функцију f_P^n дужине n дефинисану са

$$f_P^n(x_1, \dots, x_n) \stackrel{\text{def}}{=} \begin{cases} y, & \text{ако } P(x_1, \dots, x_n) \downarrow y \\ \text{недефинисано,} & \text{ако } P(x_1, \dots, x_n) \uparrow \end{cases}$$

за произвољне природне бројеве x_1, \dots, x_n .

- ▶ Домен D функције $f_P^n : D \rightarrow \mathbb{N}$ чине све n -торке природних бројева за које програм P конвергира, тј.

$$D = \{(x_1, \dots, x_n) \mid P(x_1, \dots, x_n) \downarrow\}$$

- ▶ За функцију f_P^n кажемо да је израчунљива.
- ▶ Само функције које се могу дефинисати на овај начин сматраћемо израчунљивим.

Дефиниција. Функција $f : D \rightarrow \mathbb{N}$, $D \subseteq \mathbb{N}^n$ је израчунљива ако постоји програм P који израчунава њене вредности, тј.

- ▶ ако $(x_1, \dots, x_n) \in D$ онда $P(x_1, \dots, x_n) \downarrow f(x_1, \dots, x_n)$,
- ▶ ако $(x_1, \dots, x_n) \notin D$ онда $P(x_1, \dots, x_n) \uparrow$.

Пример. Сабирање природних бројева је израчунљива функција, јер за програм $P = (I_1, I_2, I_3, I_4)$,

$$I_1 : J(3, 2, 5)$$

$$I_2 : S(1)$$

$$I_3 : S(3)$$

$$I_4 : J(1, 1, 1)$$

важи


$$P(x, y) \downarrow x + y \quad \text{за свако } x, y \in \mathbb{N},$$

што доказује да је функција

$$f_P^2(x, y) = x + y \quad (x, y \in \mathbb{N})$$

израчунљива. Њен домен је скуп \mathbb{N}^2 .

Исти програм доказује да је функција


$$g_P^3(x, y, z) = \begin{cases} x + y - z, & \text{ако } y \geq z \\ \text{недефинисано,} & \text{ако } y < z \end{cases}$$

израчунљива.

Њен домен је $D = \{(x, y, z) \in \mathbb{N}^3 \mid y \geq z\}$

Теорема.

- (а) Нула функција $z : \mathbb{N} \rightarrow \mathbb{N}$, $z(x) = 0$ је израчунљива функција.
- (б) Функција следбеник $s : \mathbb{N} \rightarrow \mathbb{N}$, $s(x) = x + 1$ је израчунљива.
- (в) Пројекције $\pi_i^n : \mathbb{N}^n \rightarrow \mathbb{N}$ $\pi_i^n(x_1, \dots, x_n) = x_i$, $1 \leq i \leq n$, су израчунљиве функције.

Доказ. Одговарајуће програме који доказују израчунљивост ових функција чини само по једна инструкција и то:

- (а) $I_1 : Z(1)$
- (б) $I_1 : S(1)$
- (в) $I_1 : T(i, 1).$

Пример. Показати да је функција $x \dot{-} 1 = \begin{cases} x - 1, & x > 0 \\ 0, & x = 0 \end{cases}$ **к 21**
израчунљива.

Ова функција је израчунљива, јер следећи програм рачуна њене вредности:

$I_1 \quad J(1, 2, 10)$

$I_2 \quad S(3)$

$I_3 \quad J(1, 3, 7)$

$I_4 \quad S(2)$

$I_5 \quad S(3)$

$I_6 \quad J(1, 1, 3)$

$I_7 \quad T(2, 1)$

Пример. Показати да је функција

$f(x) = \begin{cases} \frac{x}{2}, & \text{ако је } x \text{ паран} \\ \text{недефинисано,} & \text{ако је } x \text{ непаран} \end{cases}$ израчунљива.