Programiranje 1
2. predavanje — dodatak

Saša Singer

singer@math.hr

web.math.hr/~singer

PMF – Matematički odjel, Zagreb

Sadržaj predavanja

- "Matematički" model računala — Turingov stroj:
  - Ideja i važnost.
- Glavni dijelovi Turingovog stroja:
  - Traka.
  - Glava.
  - Stanja stroja.
  - Program.

“Matematički” model računala

Turingov stroj

Sadržaj

- "Matematički" model računala — Turingov stroj:
  - Ideja i važnost.
- Glavni dijelovi Turingovog stroja:
  - Traka.
  - Glava.
  - Stanja stroja.
  - Program.

Uvod — modeli računala

Već smo ukratko opisali von Neumannov model računala. Osnovna stvar u tom modelu:

- podaci i instrukcije (algoritam) spremljeni su u istoj memoriji.

Prednost: efikasna podloga za realizaciju računala opće namjene (izvršavanje raznih algoritama).

Sad bismo trebali opisati kako izgledaju osnovni dijelovi takvog računala — memorija i procesor (to je osnova za pisanje algoritama u nekom programskom jeziku).

Prije toga, zanimljivo je pogledati kako izgleda

- matematički model računala, tzv. Turingov stroj.

(Poslije se nećemo vraćati na to.)

Turingov stroj

Turingov stroj je

- matematički (apstraktni) stroj za izvršavanje algoritma.

Važnost u matematici (tzv. Churchova teza):

- sve što se uopće "može algoritamski izračunati" (bilo u matematici, bilo u praksi), može se realizirati Turingovim strojem.

Drugim riječima:

- Turingov stroj je univerzalni model računala ("nema jačeg" stroja).

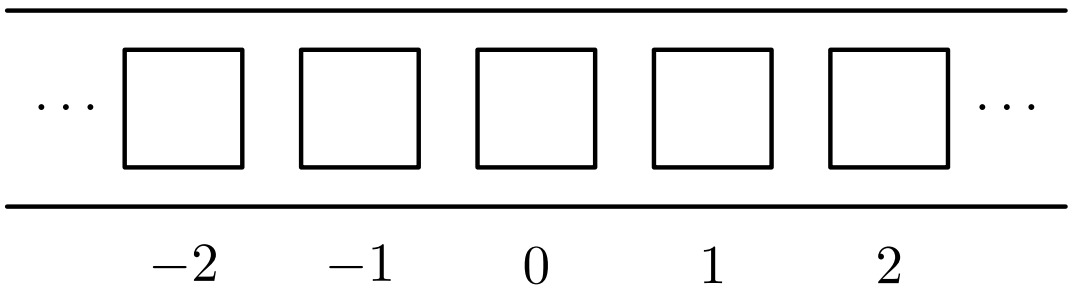
Turingov stroj — malo povijesti

- Turingov stroj je kao ideja stariji od von Neumannovoga. Nastao je krajem dvadesetih i početkom tridesetih godina prošlog stoljeća.
- Autor Alan Turing, logičar, koji nije bio samo teoretičar, već je projektirao specijalna računala koja su Britanci u Bletchley Parku koristili za razbijanje šifri njemačkih strojeva za šifriranje (Enigma).
- Jedna od najvećih nagrada u računarstvu je Turingova nagrada. Dodjeljuje ju ACM.
  - Dobitnik 2005. g.: Peter Naur, autor jezika Algol 60.

Turingov stroj — traka

- Kako izgleda Turingov stroj? On se sastoji od nekoliko bitnih dijelova.

(a) Turingov stroj ima dvostrano beskonačnu traku koja sadrži polja ("kvadratiće") numerirane cijelim brojevima:



- Svako polje može sadržavati jedan znak iz nekog skupa znakova koji stroj prepoznaje, tj. kojeg zna "pročitati" i "napisati".

Turingov stroj — traka (nastavak)

- Jednostavnosti radi, uzmimo da se taj skup znakova sastoji iz samo 3 simbola: '0', '1' i ' ' ("praznina").
- Prva 2 znaka su binarne znamenke i služe za zapis "korisnih" informacija na traci, a praznina služi kao oznaka kraja zapisa podataka.
- Uočiti: praznina osigurava konačnost zapisa na traci ("ulaznog" niza binarnih znamenki)!
- Mogli smo uzeti i veći skup znakova, ali manji ne smijemo! Razlog: kôdiranje ulaza mora biti razumno kratko.

Primjer kôdiranja

- Pogledajmo kako u ‘0’, ‘1’ alfabetu kodiramo nenegativne cijele brojeve. Ako je $n \in \mathbb{N}$, onda ćemo ga u takvom alfabetu kodirati binarnim znamenkama $0 \mapsto$ ‘0’ i $1 \mapsto$ ‘1’. Broj n prikazat ćemo u bazi $b = 2$ kao:

$$n = a_k \cdot 2^k + \cdots + a_1 \cdot 2 + a_0, \quad a_i \in \{0, 1\}, \quad a_k > 0.$$

  U stroju će to biti prikazano kao niz znamenki:

$$a_k,\ a_{k-1},\ \ldots,\ a_1,\ a_0.$$

- Uz to, moramo se dogovoriti da je zapis nule $a_0 = 0$, duljine jedne znamenke.

Primjer kôdiranja (nastavak)

- Pogledajmo koliko nam je znakova z potrebno za kodiranje broja n, za $n > 0$. Očito je

$$2^k \leq n < 2^{k+1}.$$

Logaritmiranjem dobivamo $k \leq \log_2 n < k + 1$, pa je $\lfloor \log_2 n \rfloor = k$, što znači da je

$$\lfloor \log_2 n \rfloor \leq \log_2 n < \lfloor \log_2 n \rfloor + 1.$$

Dakle, za kodiranje nam je potrebno:

$$z = \begin{cases} \lfloor \log_2 n \rfloor + 1, & n > 0, \\ 1, & n = 0 \end{cases}$$

znamenki.

Primjer kôdiranja (nastavak)

- Što ako uzmemo manji alfabet, koji se sastoji samo od '0' i ' ', uz dogovor da praznina opet služi za oznaku kraja ulaza?
- Tada se svaki $n \in \mathbb{N}$ može zapisati korištenjem n znakova 0, pa je duljine zapisa linearna u n!
- Javlja se još jedan problem, kako zapisati broj 0. Očiti zapis '0' nije moguće koristiti, jer smo ga "potrošili" na zapis broja 1.
- Zaključak: nije dobro koristiti alfabet sa samo 2 znaka, jer je duljina zapisa linearna, a ne više logaritamska u n, što može drastično utjecati na složenost algoritama.

Turingov stroj — glava

- Iz svega što smo dosad rekli, očito je da traka služi kao memorija Turingovog stroja.

(b) Traka Turingovog stroja ima glavu koja može napraviti sljedeće operacije s trakom:

- pročitati jedan znak s trake (s polja koje se nalazi "ispod" glave),
- napisati jedan znak na traku (u polje "ispod" glave),
- pomaknuti se, relativno obzirom na trenutnu poziciju, za jedno mjesto (polje) nadesno (pomak $+1$) ili za jedno mjesto nalijevo (pomak -1).

Turingov stroj — programski dio

- Pisanje, čitanje i pomaci glave Turingovog stroja pokazuju da ona služi kao kontrolni mehanizam Turingovog stroja.

(c) "Programski dio" Turingovog stroja sastoji se od konačnog niza stanja u kojima se stroj može nalaziti. U svakom trenutku stroj se nalazi u točno jednom od mogućih stanja.

Moguća stanja su podijeljena u 3 "vrste":

- "regularna stanja" ("međustanja" ili radna stanja): zovemo ih $q_1, \ldots, q_s$,
- "početno stanje": zovemo ga q_0,
- "završno stanje": zovemo ga q_f.

Turingov stroj — završna stanja

- Katkad se uzima da stroj ima više završnih stanja — koja odgovaraju raznim mogućim rezultatima algoritma.
  - Na primjer, ako algoritam daje odgovor da/ne na neko pitanje, onda stroj ima 2 završna stanja: q_y ako je odgovor da i q_n ako je odgovor ne.

Međutim, to nije jako bitno. Zapisivanjem odgovora na traku uvijek možemo postići da stroj ima jedno završno stanje.

Turingov stroj — programski modul

(d) Program ili programski modul koji stvarno upravlja strojem i "vodi ga" kroz korake pojedinog algoritma.

Kako radi programski modul? Ovisi o:

- trenutnom stanju stroja i
- znaku koji glava učita s trake.

Tada stroj:

- prijeđe u neko drugo stanje,
- napiše neki znak na traku i
- pomiče glavu jedno mjesto udesno ili ulijevo.

Dakle, različita stanja stroja su neka vrsta "programske" memorije stroja.

Turingov stroj — programski modul (precizno)

- Precizniji opis kako radi programski modul. Uzmimo da je stroj u nekom stanju q, koje nije završno, $q \neq q_f$ i da je glava u tom trenutku učitala "simbol" s trake. Na osnovu para

$$(q, \text{simbol})$$

programski modul odlučuje sljedeće 3 stvari:

(i) koje je sljedeće stanje q' u koje prelazi stroj,

(ii) koji će znak napisati u polje ispod glave (taj znak zovemo "novi simbol"),

(iii) pomiče li se traka jedno polje nadesno ili jedno polje nalijevo (pomak za $+1$ ili -1).

Turingov stroj — programski modul (precizno)

- Dakle, jedan programski korak je veza

$$(q, \text{simbol}) \quad \longrightarrow \quad (q', \text{novi simbol}, \text{pomak}).$$

  Ako je ova veza funkcija, tj. svakom paru (q, simbol) pridruži točno jednu trojku $(q', \text{novi simbol}, \text{pomak})$, onda je stroj deterministički. Ali to ne mora biti tako!

- Kad stroj dođe u završno stanje q_f, onda se računanje prekida (završava).

- Na početku stroj je u stanju q_0, a glava se nalazi nad poljem s brojem 1.

Ovim smo, zapravo, napravili sve elemente za formalnu definiciju Turingovog stroja.