1
Narava računanja in stroji za računanje

Razlogi za strojno računanje

Čemu strojno računanje?

Ročno računanje, 2 problema:

1. počasnost
2. nezanesljivost

Povezava med ročnim in strojnim računanjem

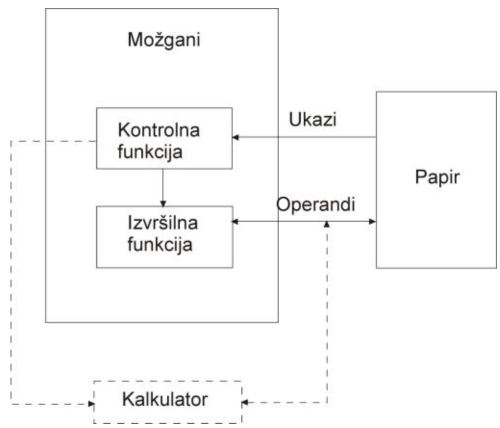
Ročno računanje
- papir (→ pomnilnik)
- možgani (→ procesor)

Papir
- ukazi (navodila)
- operandi

Možgani pri računanju opravljajo 2 funkciji:
- kontrolna funkcija
  - prevzema ukaze in skrbi za pravilen vrstni red izvrševanja ukazov
- izvršilna funkcija
  - npr. seštevanje, množenje, itd.

Papir lahko delimo v 2 vrsti:
- knjiga z navodili (→ ukazi)
- papir za vmesne in končne rezultate (→ operandi)

Ročno računanje



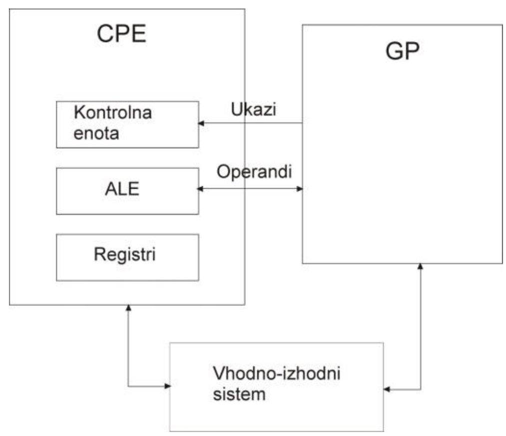
Strojno računanje

Današnji računalniki računajo na podoben način kot človek

Tudi računalnik ima lahko pomnilnik ločen na 2 dela:

- del za ukaze
- del za operande

Strojno računanje



Računanje in izračunljivost

Kakšni naj bodo stroji, ki znajo računati?

- Potrebno je najprej natančno definirati, kaj sploh je računanje

Tudi teoretično zanimiv problem:

- Kakšen naj bo stroj, da bo znal izračunati vse, kar se da izračunati?
- Kaj sploh pomeni, da se nekaj da izračunati?

Kako definirati računanje?

Računanje lahko definiramo kot določanje vrednosti funkcije $z = f(x)$

- funkcija f je mišljena zelo široko
- x so vhodni podatki, z pa izhodni

Beseda *računanje* (v slovenskem jeziku) ima 2 pomena:
- numerično računanje (calculation)
- računanje v širšem pomenu (computing)

Definicija izračunljivosti:

Funkcija *f(x)* je **izračunljiva**, če obstaja postopek, s katerim lahko določimo njeno vrednost (*z*) za vse možne vhodne podatke (*x*), nad katerimi je definirana.

Ta postopek je lahko zaporedje več korakov

Rečemo mu tudi algoritem

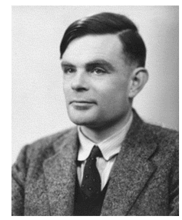
Algoritem je navodilo, ki v končnem številu korakov pripelje do želenega rezultata
- npr. Evklidov algoritem za izračun NSD 2 števil
- algoritem ni nujno povezan z računalniki
  - Npr.: recept iz kuharske knjige

Definicija izračunljivosti je torej tudi:

Funkcija je izračunljiva, če zanjo obstaja algoritem

Ali za vsak problem obstaja algoritem?

oz. Ali je vsak problem izračunljiv?



Teoretični modeli računanja:
- Turingov stroj (Alan Turing), 1936

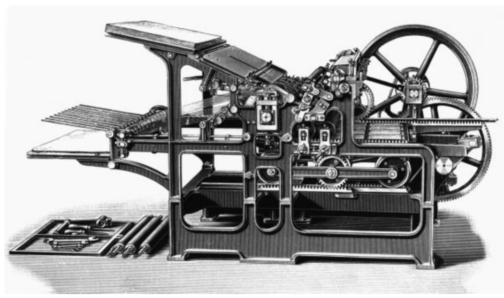
Church-Turingova hipoteza:

Problem je izračunljiv, če ga je možno v končnem številu korakov izračunati na Turingovem stroju

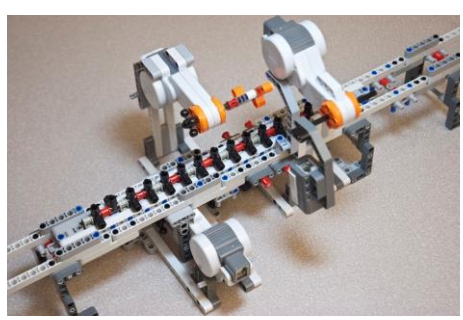
Turingovi stroji

Turingov stroj (Turing machine, TM) sestavljajo:
- procesor
- bralno-pisalna glava
- neskončno dolg trak
- mehanizem za pomik traku

- “Stroj” je mišljen kot abstrakten model računanja
  - ne kot neka mehanska naprava, npr.:



Kar pa ne pomeni, da ga ni možno fizično realizirati (v približku)



Pisanje programov za TM ni enostavno
- primitivni ukazi

Za vsako kombinacijo stanja avtomata in vhodne črke (na traku) definiramo, kaj glava zapiše na trak in smer pomika

Program za TM lahko ponazorimo s tabelo ali diagramom prehajanja stanj (DPS)

Računalniki in Turingovi stroji

Današnji rač. delujejo po von Neumannovem modelu
- ta je ekv. TM (če bi bil pomnilnik neskončen)
- manj primitiven, hitrejši
- TM je abstrakten (matematičen) model
  - enostavnost je v funkciji lažjega teoretičnega dokazovanja

Če je trak TM končen, a dolg, se da rešiti večino praktičnih problemov

Omejitve računalnikov

2 vrsti "težavnih" problemov:
- Neizračunljivi problemi
- Neobvladljivi problemi

Neizračunljivi problemi

Ustavitveni problem (Halting problem)
- Turing je dokazal, da ni mogoče napisati algoritma, ki bo ugotovil, ali se bo poljuben TM s poljubnim podatkom kdaj ustavil

Teoretične raziskave izračunljivosti
- Prevedba problema ustavljanja na problem, ki ga raziskujemo

Neobvladljivi problemi

To so izračunljivi problemi, ki pa jih ne moremo rešiti zaradi
- omejenega pomnilnika, in/ali
- omejenega časa

Teorija kompleksnosti
- prostorska kompleksnost
- časovna kompleksnost (običajno hujša)
  - polinomska: $O(n)$, $O(n*\log n)$, $O(n^2)$, $O(n^3)$, ...
  - eksponentna: $O(2^n)$, $O(n!)$, $O(n^n)$, ...