Teoretične osnove računalništva

Zapiski predavanj 2010/2011

28. februar 2011

Kazalo

Poglavje 1

Turingov Stroj

1.1 Zgodovina

Leta 1900 je Nemški matematik David Hilbert objavil seznam triidvajsetih nerešenih problemov v matematiki. Eden izmed Hilbertovih problemov (deseti po vrsti), je vprašanje, ali obstaja postopek, po katerem ugotovimo rešljivost poljubne Diofantske enačbe – torej, ali lahko ugotovimo, če ima polinom s celoštevilskimi koeficienti $P(x_1, x_2, \ldots, x_n) = 0$, celoštevilsko rešitev. Kljub temu, da je Emil Post že leta 1944 slutil, da je problem nerešljiv, je to dokončno dokazal rus Jurij Matijaševič šele leta 1970 v svojem doktorskem delu. Med reševanjem problema pa so se matematiki že prej začeli ukvarjati s formalizacijo pojma postopka oz. algoritma. Intuitivna definicija tega se glasi nekako tako:

Def.: Algoritem je zaporedje ukazov, s katerimi se v končnem številu korakov opravi neka naloga.

Pri tem pa ostaja še kar nekaj odprtih vprašanj, npr.:

- Kakšni naj bodo ukazi?
  - Osnovni - algoritem ima veliko korakov
  - Kompleksni - prezapleteni ukazi so že sami algoritmi
- Koliko ukazov naj bo?
  - Končno - ali je s končno množico res mogoče rešiti vsako nalogo?
  - Neskončno - kakšen izvajalec ukazov je sposoben izvršiti neskončno različnih ukazov?
- So ukazi zvezni ali diskretni?
- V kakšnem pomnilniku so ukazi shranjeni?
  - Končnem - ali s končnim zaporedjem ukazov res lahko mogoče rešimo vsako nalogo?
  - Neskončnem -

Nekateri zgodnji poskusi formalizacije pojma algoritma so:

- GK (Kurt Gödel, Stephen Kleene)
- HG (Jacques Herbrand, Kurt Gödel)
- Produkcijski sistem (Emil Post),
- Lambda račun (Alonso Church, 1936)
- Turingov stroj (Alan Turing, 1936)

1.2 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:

- Nadzorno enoto (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: $Q \times \Gamma \to Q \times \Gamma \times \{L, D\}$, 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 simbov v celico, ki je pod oknom
- okno premakne eno celico levo ali desno

1.2.1 Trenutni opis

Def.: $TO = \Gamma^* \times Q \times \Gamma^*$ je množica vseh trenutnih opisov.
Nek trenutni opis (α_1, q, α_2), ali krajše $\alpha_1\ q\ \alpha_2$ opisuje konfiguracijo Turingovega stroja.

Iz α_1 in α_2, 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

1.2.2 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 $\delta(q, x_i) = (p, Y, D)$:
  $x_1 \dots x_{i-1}\ q\ x_i \dots x_n \vdash x_1 \dots x_{i-1}\ Y\ p\ x_{i+1} \dots x_n$
- če je $\delta(q, x_i) = (p, Y, L)$:
  - 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: $x_1 \dots x_{i-2} x_{i-1}\ q\ x_i \dots x_n \vdash x_1 \dots\ x_{i-2} p\ x_{i-1}\ Y\ x_{i+1} \dots x_n$

1.2.3 Tranzitivna ovojnica $\vdash^*$ relacije $\vdash$

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.

1.3 Jezik Turingovega stroja

Def.: Jezik Turingovega stroja je definiran kot:

$$L(M) = \{w \mid w \in \Sigma^* \wedge q_0 w \vdash^* w_1\ q\ w_2 \wedge w_1, w_2 \in \Gamma^* \wedge q \in F\}$$

Z besedami to pomeni, da je $L(M)$ množica besed $w \in \Sigma^*$, ki če jih damo na vhod stroju M, povzročijo, da se stroj M v končno mnogo korakih znajde v končnem stanju.

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

1.3.1 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.

Primeri:

Primer 1: TS, ki sprejema: $\{0^n 1^n | n \geq 1\}$

- $00\ldots011\ldots1$ Zamenja levo 0 z X
- $X0\ldots011\ldots1$gre desno in ko doseže prvo 1 zamenja 1 z Y
- $X0\ldots0Y1\ldots1$ Gre do najbolj desnega X...

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

- q_0 - začetno stanje in stanje pred zamenjavo 0 z X
- q_1 - v tem stanju se pomika desno do 1
- q_2 - v to stanje preide po zamenjavi 1 z Y in gre levo do zadnjega X
- q_3 - v to stanje preide, ko najde X in se premane nekrat v desno
- q_4 - končno stanje

Napišemo tabelo:

	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	$\langle q_2, 0, D\rangle$	–	$\langle q_0, X, D\rangle$	$\langle q_2, Y, D\rangle$	–
x_3	–	–	–	$\langle q_3, Y, D\rangle$	$\langle q_4, B, D\rangle$
x_4	–	–	–	–	–

Simuliramo s trenutnimi opisi:
$q_0 0011 \vdash X q_1 011 \vdash X0 q_1 11 \vdash X q_2 0Y1 \vdash \ldots$

Primer 2: Turingov stroj kot računalnik funkcij:
$M = \langle Q, \Sigma, \Gamma, \delta, q_0, B, F\rangle$
$00\ldots0100\ldots01\ldots1000\ldots$
Lahko, da se stroj M ustavi ter, da ima na traku tedaj m ničel (in B na levi in desni od te grupe)
Lahko, da je stroj s tem izračunal neko funkcijo $f^{(k)} : \mathbb{N}_+^k \to \mathbb{N}_+$ oz. $f(i_1, i_2, \ldots, i_n) = m$ f ni nujno definirana pri vsaki k-terici števil iz $\mathbb{N}$, torej je parcialna funkcija. TS M hkrati računa več funkcij $f^{(1)}.., f^{(2)}.., ...fn$

1.3.2 Parcialna rekurzivna funkcija

Parcialna rekurzivna funkcija je vsaka funkcija, ki jo računa nek TS (v opisanem smislu)

Def.: $f^k : \mathbb{N} \to \mathbb{N}$ je prf, če obstaja Turingov stroj, ki računa to f kot smo opisali. Če je f^k definirana za vse k-terice je totalna rf, oz. samo rf.

Trditev: Vse običajne aritmetične funkcije (na $\mathbb{N}^k$) so prf ali celo rf Primer: $+, *, n!, 2^n, |^- logn, m^n, \ldots$