Teorija izračunljivosti

Zoran Ognjanović

zorano@mi.sanu.ac.rs

MF Beograd, 2012/13

Teorija izračunljivosti (3I):

- predavanja: 2 časa nedeljno
- asistent:
    - Filip Samardžić
- pravila polaganja:
    - polaganje od januara 2013. godine
    - predispitne obaveze 30 poena (2 kolokvijuma)
    - ispit 70 poena (pismeni, usmeni)
    - seminarski radovi
    - za 6 je potrebno 50 poena
- Internet-stranica predmeta u pripremi
  `http://www.mi.sanu.ac.rs/~zorano/ti/2012/ti.html`

Literatura:

- Z. Ognjanović, N. Krdžavac, Uvod u teorijsko računarstvo, FON, Beograd, 2004.
  http://www.mi.sanu.ac.rs/~zorano/ti/TeorijskoRacunarstvo.pdf

Pomoćna literatura:

- N. Cutland, Computability, an introduction to recursive function theory, Cambridge university press, 1986.
- J.Hopcroft, J.Ullman, Formal languages and their relation to automata, Addison-Wesley, 1969.
- Ž. Mijajlović, Z. Marković, K. Došen, Hilbertovi problemi i logika, Zavod za udžbenike i nastavna sredstva, Beograd, 1986.
- C. Papadimitriou, Computational complexity, Addison-Wesley, 1995.
- Irena Spasić, Predrag Janičić, TAJA - Zbirka zadataka, Matematički fakultet, Beograd, 2000.

Izračunljivost

- Teorija izračunljivosti (tj. teorija algoritama)
- oblast nastala između 1930. i 1940. godine, pre razvoja digitalnih računara
- rezultat pretresanja osnova matematike zbog paradoksa koji su se pojavili krajem XIX i početkom XX veka.

Izračunljivost

- postavljalo se pitanje (*Entscheidungsproblem, problem odlučivanja*) da li postoji opšti postupak utvrđivanja istinitosti matematičkih iskaza
- vodi poreklo još od Leibnitz-a koji je u XVII veku, nakon uspešne konstrukcije mehaničke računske mašine, razmišljao o konstrukciji mašine koja bi mogla manipulisati simbolima i na taj način odrediti istinitost iskaza
- problem je aktuelizovao Hilbert, u poznatom predavanju na Kongresu matematičara održanom 1900
- odgovor - treba precizirati šta se podrazumeva pod postupkom mehaničkog izvođenja
- nezavisno jedan od drugog, Čerč i Tjuring su negativno odgovorili na ovo pitanje (jednakosti λ-izraza, halting problem)

Gottfried Wilhelm Leibnitz (Lajbnić), 1646 – 1716

Matematičar i filozof, verovatno slovenskog porekla. Jedan od kreatora diferencijalnog računa, mislilac koji je išao daleko ispred svog vremena. Nakon uspešne konstrukcije mehaničke računske mašine, razmišljao je o konstrukciji mašine koja bi mogla manipulisati simbolima jednog veštačkog univerzalnog jezika i na taj način odrediti istinitost iskaza.

D. Hilbert, 1862 – 1943, W. Ackermann, 1896 – 1962

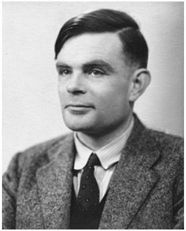
David Hilbert, nemački matematičar (levo). Smatra se da je jedan od najuticajnijih matematičara svih vremena. Wilhelm Friedrich Ackermann, nemački matematičar (desno). Sarađivao je sa Hilbertom u pisanju knjige *Grundzüge der theoretischen Logik*.

K. Gödel, 1906 – 1978, J. Herbrand, 1908 – 1931

Kurt Gödel, austrijski logičar (levo). Najpoznatiji je po rezultatima o esencijalnoj nepotpunosti aksiomatizacije Peanove aritmetike koji su imali dalekosežne posledice po razvoj ljudske misli. Uvršten je u listu 100 najznačajnijih ličnosti 20. veka koju je 1999. godine objavio magazin TIME.

Jacques Herbrand, francuski matematičar (desno). Dao je fundamentalne priloge u oblasti izračunljivih funkcija i logici. Poginuo na planinarenju.

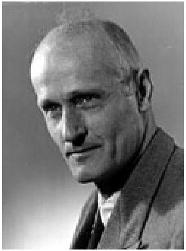
A. Church, 1903 –1995, A. Turing, 1912 – 1954

Alonzo Church, američki matematičar (levo). Najpoznatiji po doprinosima u logici i osnovama teorijskog računarstva. Uveo je λ-račun i formulisao čuvenu hipotezu nazvanu po njemu.

Alan Turing, engleski logičar (desno). Razvio ideje formalizacije koncepta izračunavanja, računara kao univeralne mašine, veštačke inteligencije. Tokom Drugog svetskog rata je rukovodio razbijanjem Enigma, razvio elektro-mehaničku mašinu Bombe 1940. godine, i dao osnovne ideje za prvi programibilni elektronski računar. Uvršten je u listu 100 najznačajnijih ličnosti 20. veka koju je 1999. godine objavio magazin TIME.

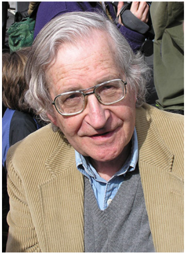
E. Post, 1897 – 1954, S. Kleene, 1909 – 1994

Emil Leon Post,matematičar poljskog porekla (levo). Najpoznatiji je po radu u oblasti teorije izračunljivost.

Stephen Kleene, američki matematičar (desno). Smatra se jednim od osnivača teorije rekurzivnih funkcija.

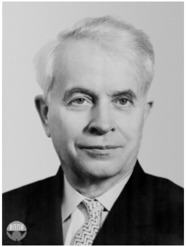
J. Nojman, 1903 – 1957, N. Čomski, 1928 –

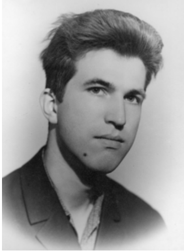


János-a Neumann, John von Neumann, matematičar mađarskog porekla (levo). Jedan od najuticijnijih matematičara 20. veka. Radio u teoriji skupova, funkcionalnoj analizi, teoriji igara i ekonomiji, kao i u fizici, posebno u kvantnoj mehanici. Učesnik projekta Manhattan, savetnik u timu u koji je realizovao projekt EDVAC i autor izveštaja *First Draft of a Report on the EDVAC* iz 1945. godine u kome su data logička načela koje su u osnovi arhitekture i današnjih računara.

Avram Noam Chomsky (desno), čuveni liberalni mislilac i aktivista. Poznat je kao osnivač moderne lingvistike.

A. Markov, 1903 – 1979, J. Matijaševič, 1947 –





Андрей Андреевич Марков (levo), ruski matematičar. Jedan je od onivača ruske škole konstruktivne matematike.

Юрий Матиясевич (desno), ruski matematičar. Rešio deseti Hilbertov problem pokazavši da rešivost diofantskih jednačina nije odlučiva.

- rešavanje problema razvojem algoritama i pisanjem programa je jedan od osnovnih zadataka u računarstvu
- matematičkim sredstvima proučavaju se i sami postupci rešavanja, algoritmi.
- formalni modeli izračunavanja, naizgled različiti, određuju jednu te istu klasu algoritama što navodi na tezu da ti modeli izračunavanja upravo određuju granice mogućnosti mehaničkog izračunavanja
- granice razdvajaju klase problema na one na za koje, u principu, postoji mogućnost programskog rešavanja i one za koje to nije slučaj

- postupak se opisuje konačnim nizom jednostavnih naredbi
- postoji idealna mašina (računar) koja izvršava te naredbe prema unapred utvrđenim pravilima,
- ta mašina započinje izračunavanje u nekom inicijalnom stanju; primenjena na ulazne podatke mašina izvršava naredbe u diskretnim koracima u kojima menja svoja stanja,
- izvršavanje svake naredbe se izvodi u konačnom vremenu pri čemu se koristi konačan memorijski prostor,
- izvršavanje naredbe je determinističko: iz jednog stanja izvršavanjem iste naredbe mašina uvek prelazi u isto stanje i
- prelaskom u završno stanje mašina prestaje sa izračunavanjem.

- determinisanost - mogućnost ponavljanja izvršavanja algoritama
- nedeterministički algoritmi
- ne zahteva se da se algoritam uvek završi, tj. da se rezultat uvek dobije u konačnom vremenu (finitnost),
- slično i sa zahtevom za ukupno memorijsko zauzeće
- algoritam predstavlja opis funkcije koja ulazne podatke preslikava u odgovor
- algoritamske (efektivne, izračunljive) funkcije

- Za probleme za koje poznajemo postupke rešavanja lako utvrđujemo da jesu algoritamski rešivi.
- Šta ako nismo u stanju da damo rešenje?
- Da li je to samo posledica naše nesposobnosti, ili je reč o principijelnoj nemogućnosti?
- Potrebno je formalno precizirati pojmove algoritma.

Diofantske jednačine

- Problem postojanja efektivnog postupka za utvrđivanje da li proizvoljna diofantskih jednačina $p(x_1, \ldots, x_m) = 0$ ima nenegativna celobrojna rešenja
- $p(x_1, \ldots, x_m)$ je polinom sa celobrojnim koeficijentima i promenljivim x_1, ..., x_m
  $x_1^4 x_3 - 3x_2^5 + 6$
- nabrajanjem svih m-torki prirodnih brojeva i proverom da li predstavljaju nule polinoma bi se, pre ili posle, stiglo do rešenja jednačine, ako ono postoji
- neke jednačine ovog tipa $(x^2 - 2 = 0)$, nemaju rešenja, prethodno opisani postupak se u takvim slučajevima ne bi nikada završio, zbog čega i ne predstavlja rešenje problema

Diofantske jednačine

- provera postojanja rešenja diofantskih jednačina je zapravo ekvivalentna formulacija desetog Hilbertovog problema
- svaki eventualni odgovor na ovo pitanje mora na neki način ponuditi i formalnu definiciju onoga što se podrazumeva pod efektivnim postupkom, bilo u smislu da ponuđeno rešenje potpada pod tu definiciju, bilo da ne postoji rešenje sa zahtevanim svojstvima
- formalna definicija efektivnog postupka pojavila se razvojem teorije izračunljivosti
- Jurij Matijaševič je 1970. godine sredstvima razvijenim u okviru te teorije negativno rešio sam problem

Formalni algoritamski sistemi

- Sistem izračunljivosti predstavljen u formalnom sistemu aritmetike je predložio Gödel između 1931. i 1934. godine, pri čemu se funkcija f smatra izračunljivom ako za svako m i n za koje je $f(m) = n$, u formalnom sistemu važi $\vdash f(m) = n$.
- Prikazivanje izračunljivih funkcija kao jedinstvenih rešenja sistema funkcionalnih jednačina je u istom periodu opisao takođe Gödel, a prema ideji Erbrana.
- λ-račun koji je razvio Čerč do 1936. godine je jednostavan formalni jezik za koji se definiše pojam redukcije koji predstavlja izračunavanje, a funkcija je izračunljiva ako se može opisati u jeziku.

Formalni algoritamski sistemi

Sistemi zasnovani na automatima formalizuju pojam algoritma opisujući idealne modele računara (neki sistemi su dati pre nastanka digitalnih računara):

- Tjuringova mašina (1936.)
- Postova mašina (1936.)
- Neograničena registarska mašina (URM, Shepherdson i Sturgis 1963.)

Formalni algoritamski sistemi

- Aritmetički opis zasnovao je Klini (generališe induktivne definicije)
- Sistemi produkcija (sistemi sa prezapisivanjem, Rewriting systems):
  - Postovi sistemi iz 1943. godine,
  - Markovljevi algoritmi uvedeni 1954. godine i
  - Gramatike Čomskog predložene 1956. godine,

  su jedna vrsta formalnih sistema u kojima se opisuju moguće transformacije jednih u druge reči na unapred fiskiranom alfabetu. Funkcije se opisuju kao skupovi parova reči (u, v) za koje postoji niz reči koje se dobijaju počev od u primenama pravila izvođenja i koji završava rečju v.
- *while*-programi su vrsta notacije proizašle iz ideja Goldstine-a i von Neumann-a o algoritamskim šemama kao formalizmu za prikazivanje izračunljivih funkcija. *while*-programi se sastoje samo od naredbi dodeljivanja, nizanja naredbi i *while*-naredbi.

Formalni algoritamski sistemi

- izvori inspiracije ovih sistema se međusobno značajno razlikuju
- pokazuje se da su sistemi međusobno ekvivalentni
- neuspeh pokušaja konstrukcije zadatka i postupka njegovog rešavanja koji ne potpadaju pod ove klasifikacije
- Čerčova teza: verovanje da je postignut nekakav apsolutni koncept i da se svi algoritmi mogu izraziti u svakom od ovih sistema

Formalni algoritamski sistemi

- izvori inspiracije ovih sistema se međusobno značajno razlikuju
- pokazuje se da su sistemi međusobno ekvivalentni
- neuspeh pokušaja konstrukcije zadatka i postupka njegovog rešavanja koji ne potpadaju pod ove klasifikacije
- Čerčova teza: verovanje da je postignut nekakav apsolutni koncept i da se svi algoritmi mogu izraziti u svakom od ovih sistema

Apstraktni model digitalnog računara

- celina sastavljena od procesora, memorije i ulazno-izlaznih uređaja
- procesor iz memorije pribavlja naredbe i podatke
- procesor vrši obradu u skladu sa značenjem naredbi
- dobijeni rezultati se vraćaju u memoriju
- preko ulazno-izlaznih uređaja podaci koji će biti obrađeni se unose u memoriju, odnosno iz memorije se preuzimaju rezultati obrade i prikazuju na odgovarajući način
- komunikacija delova računara se obavlja preko magistrala.

Tjuringova mašina je preteča ovakvog modela računara, pri čemu su neka svojstva idealizovana:

- memorija - potencijalno beskonačna
- u svakom koraku izračunavanja Tjuringove mašine zauzet je samo konačan broj memorijskih registara,
- ne postoji ograničenje koliki je taj konačan broj registara
- svakom koraku izračunavanja moguće je i zahtevati novi, do tada neiskorišteni memorijski registar i svaki takav zahtev se ispunjava

- Tjuringova mašina je restrikcija koncepta savremenog računara u smislu operacija koje je u stanju da izvršava, a koje su elementarne.
- Operacije su dovoljne za opisivanje proizvoljnih algoritama.
- Prednost u odnosu na bogatije programske jezike je upravo u jednostavnosti koja olakšava analizu.

Restrikcije Tjuringove mašine

Razmatraju se i slabije klase mašina koje su restrikcije druge vrste u odnosu na aktuelne računare:

- neki modeli nemaju memoriju (konačni automati - Finite automata)
- memorija je organizovana na poseban način (stek kod potisnih automata - Push-down automata),
- ulazno-izlazni podaci su ograničeni na reči,
- neki automati nemaju izlaz (konačni automati) itd.

- svaki problem se izražava u nekom jeziku
- alfabet je skup znaka koji su nedeljive celine
- reč na nekom alfabetu je bilo koja konačna sekvenca znaka tog alfabeta
- sekvenca od nula znaka se naziva prazna reč
- reči se razdvajaju znakom blanko koji se ne smatra delom alfabeta već pomoćnim simbolom
- jezik je neki podskup skupa svih reči odgovarajućeg alfabeta
- t je podreč od q ako postoje, možda i prazne, reči u i v tako da je $q = utv$.
- obično je alfabet konačan skup znaka
- koristićemo unarni alfabet $A = \{1\}$
- blanko-znak (0)

Tjuringova mašina se sastoji od:

- *trake* podeljene u ćelije, memorijske registre, koja se neograničeno pruža na levo i desno; broj ćelija (tj. dužina trake) je neograničen; sadržaj svake ćelije je ili znak 1 ili blanko znak (znak 0),
- *glave* koja se uvek nalazi nad tačno jednom ćelijom trake i može:
  - pročitati sadržaj ćelije nad kojom se nalazi i
  - upisati u ćeliju nad kojom se nalazi znak 1 ili 0 (blanko znak, tj. obrisati ćeliju) ili pomeriti se za jedan korak u levo ili u desno u odnosu na trenutnu poziciju,
- *indikatora stanja mašine*.

- TM se u svakom trenutku nalazi u tačno jednom od konačno mnogo stanja koje se eventualno menja nakon svakog koraka izračunavanja.
- Skup svih stanja mašine označićemo sa $S = \{q_0, q_1, \ldots\}$.
- Izvršavanje mašine se izvodi pod dejstvom programa - konačan niz naredbi.
- Svaka naredba je četvorka oblika:
  $q_i \ s \ o \ q_j$
  gde su q_i i q_j neka stanja iz skupa S, s je znak nad kojim se nalazi glava mašine, a $o \in \{1, 0, L, R\}$ je oznaka operacije.

- U svakom koraku rada mašina analizira stanje u kojem se nalazi i sadržaj ćelije nad kojom je glava, a zatim izvršava naredbu koja ima odgovarajuće vrednosti parametara q_i i s.
- Efekat izvršenja naredbe $q_i\ s\ o\ q_j$:
    - ako je $o = 1$, u ćeliju nad kojom se nalazi glava upisuje se znak 1,
    - ako je $o = 0$, u ćeliju nad kojom se nalazi glava upisuje se 0, tj. blanko znak,
    - ako je $o = L$, glava se pomera ulevo za jednu ćeliju i
    - ako je $o = R$, glava se pomera udesno za jednu ćeliju.
    - potom mašina menja stanje i prelazi u stanje q_j.

Primeri determinističkih naredbi:

q_5 0 1 q_{17},

q_1 0 0 q_2 i

q_0 1 L q_0.

Primeri nedeterminističkih naredbi (q_i i s se poklapaju, a vrednosti parametara o i q_j se razlikuju):

q_4 1 1 q_5 i

q_4 1 L q_2

Konvencije

- Stanje $q_0 \in S$ - početno stanje
- Inicijalno, mašina se uvek nalazi u početnom stanju.
- Pri tome traka sadrži samo konačno mnogo ćelija u koje je upisan znak 1, dok sve ostale ćelije sadrže znak 0.
- Reč se na traci prikazuje kao neprekidan niz ćelija koje sadrže znak 1, a sa leve i desne strane tog niza se nalazi najmanje po jedan blanko znak (0).
- Po pravilu, na početku i na kraju izvršavanja glava mašine se nalazi iznad najlevlje ćelije koja sadrži znak 1.
- Skup stanja S proširićemo jednim novi stanjem q_z koje ne pripada do sada razmatranom skupu stanja.
- q_z - završno stanje; Kada se mašina nađe u stanju q_z ona prekida sa izvršavanjem.

Konfiguracije Tjuringove mašine

Definition

Pod *konfiguracijom* Tjuringove mašine podrazumevamo opis koji sadrži: opis sadržaja trake, položaj glave i stanje mašine. *Standardna konfiguracija* je konfiguracija u kojoj je:

- ili traka prazna (tj. sve ćelije sadrže blanko znak) ili sadrži najviše konačno mnogo nepraznih reči razdvojenih po jednim blanko znakom,
- glava mašine je iznad prve (gledano sa leva) ćelije trake koja sadrži znak 1 i
- ako počinje sa izvršavanjem, mašina se nalazi u početnom stanju q_0, a ako završava sa radom u završnom stanju q_z.

Programi - funkcije koje preslikavaju skup konfiguracija mašine u samog sebe.

Na traci je jedna reč sastavljena od 1 (ostatak su 0) nad čijim krajnjim levim znakom se nalazi glava. Program dopisuje dva znaka 1 sa desne strane reči, glavu vraća na levo, na početak reči, pa staje:

$q_0\ 1\ R\ q_0$	glava se pomera udesno, na kraj reči
$q_0\ 0\ 1\ q_1$	na mestu prve 0 upisuje se 1
$q_1\ 1\ R\ q_2$	glava se pomera udesno
$q_2\ 0\ 1\ q_3$	na mestu druge 0 upisuje se 1
$q_3\ 1\ L\ q_3$	glava se pomera ulevo
$q_3\ 0\ R\ q_z$	do prve 0, ide udesno i zaustavlja se

$\ldots 01q_0 1000\ldots$	$\ldots 0111q_3 10\ldots$
$\ldots 011q_0 000\ldots$	$\ldots 011q_3 110\ldots$
$\ldots 0110q_0 00\ldots$	$\ldots 01q_3 1110\ldots$
$\ldots 0111q_1 00\ldots$	$\ldots 0q_3 11110\ldots$
$\ldots 01110q_2 0\ldots$	$\ldots 01q_z 1110\ldots$
$\ldots 01111q_3 0\ldots$	

U opisu Tjuringove mašine ne kaže se šta se događa ako za sadržaj ćelije nad kojim se nalazi glava i tekuće stanje mašine u programu ne postoji odgovarajuća naredba.
Ovo odgovara 'zaglavljivanju' programa pisanih na standardnim programskim jezicima i može se formalizovati kompletiranjem programa naredbama koje u takvim situacijama ne menjaju ni stanje, ni poziciju glave, ni sadržaj ćelije nad kojom se glava nalazi. Ako u programu ne postoji naredba koja odgovara situaciji kada je mašina u stanju q_i, a sadržaj ćelije nad kojom se nalazi glava 0, možemo program proširiti naredbom koja predstavlja jednu beskonačnu petlju:

$q_i \ 0 \ 0 \ q_i$

Tjuringove mašine se mogu iskoristiti kao algoritmi, tj. za izračunavanje funkcija koje preslikavaju prirodne brojeve u prirodne brojeve.

Aritmetička funkcija

Definition

Aritmetička funkcija je preslikavanje f za koje važi:

- domen preslikavanja, $Dom(f)$, je podskup skupa $\mathbb{N}^k$ $(k > 0)$ i
- kodomen preslikavanja, $Im(f)$, je podskup skupa $\mathbb{N}$.

Ako je za neki $k > 0$, $Dom(f) = \mathbb{N}^k$, f je *totalna funkcija*. Ako je $Dom(f) \subset \mathbb{N}^k$, za neki $k > 0$ i $Dom(f) \neq \mathbb{N}^k$, f je *parcijalna funkcija*.

Definition

Unarna reprezentacija prirodnog broja n u unarnom alfabetu $A = \{1\}$ je reč koja sadrži $n + 1$ znak 1.

Tjuring-izračunljiva funkcija

Definition

Neka je f aritmetička funkcija oblika $f : X \to \mathbb{N}$, gde je $X \subset \mathbb{N}$. Funkcija f je *Tjuring-izračunljiva* ako postoji program P za Tjuringovu mašinu tako da je za svaki $m \in X$:

- pre početka izvršavanja programa P Tjuringova mašina u standardnoj konfiguraciji, pri čemu je jedina reč zapisana na traci unarna reprezentacija broja m i
- po završetku rada programa P Tjuringova mašina u standardnoj konfiguraciji, pri čemu je jedina reč zapisana na traci unarna reprezentacija broja $f(m)$.

Program P tada *izračunava* funkciju f.

Analogno je moguće definisati k-arne aritmetičke Tjuring-izračunljive funkcije.

Tjuring-izračunljive funkcije su parcijalne, odnosno ako se neki m ne nalazi u domenu Tjuring-izračunljive funkcije f, odgovarajući program P ne staje.

q_0 0 0 q_0
q_0 1 1 q_0

Program nikada ne staje, pa izračunava jedino funkciju čiji je domen prazan skup.

- $P(x_1, x_2, \ldots, x_k) \downarrow y$ označava da program P polazeći od standardne konfiguracije u kojoj traka sadrži unarne reprezentacije prirodnih brojeva x_1, x_2, ..., x_k završava rad pri čemu se mašina nalazi u standardnoj konfiguraciji u kojoj traka sadrži unarnu reprezentaciju prirodnog broja y.
- $P(x_1, x_2, \ldots, x_k) \downarrow$ znači da je za neko y ispunjeno $P(x_1, x_2, \ldots, x_k) \downarrow y$.
- $P(x_1, x_2, \ldots, x_k) \uparrow$ znači da nije $P(x_1, x_2, \ldots, x_k) \downarrow$.

Definition

Program P *konvergira* za ulaz x_1, x_2, ..., x_k ako je ispunjeno $P(x_1, x_2, \ldots, x_k) \downarrow$. Program P *divergira* za ulaz x_1, x_2, ..., x_k ako je ispunjeno $P(x_1, x_2, \ldots, x_k) \uparrow$.

Sledeći program izračunava funkciju $f(x) = 0$.

q_0 1 0 q_1
q_0 0 1 q_z
q_1 0 R q_0

Sadržaj trake se briše, pri čemu se glava pomera na desno. Kada se naiđe na prvi znak 0, upisuje se znak 1 i završava rad. Dakle, $P(x) \downarrow 0$.

Sledeći program izračunava funkciju naslednika prirodnog broja u nizu prirodnih brojeva, $f(x) = x'$.

$q_0\ 1\ L\ q_0$
$q_0\ 0\ 1\ q_z$

U programu se glava najpre pomera na levo, nakon čega se nalazi iznad ćelije koja sadrži znak 0. U tu ćeliju se upisuje znak 1 i prelazi u završno stanje. Dakle, $P(x) \downarrow x'$.

Funkcije projekcije, $f(x_1, \ldots, x_k) = x_i$ (1)

Sledeći program za fiksirane k i i ($k \geq i \geq 1$) izračunava funkciju koja se naziva i-ta projekcija, $f(x_1, \ldots, x_k) = x_i$.

$q_0\ 1\ 0\ q_1$	briše zapisa broja x_1
$q_0\ 0\ R\ q_2$	
$q_1\ 0\ R\ q_0$	
$\ldots$	
$q_j\ 1\ 0\ q_{j+1}$	briše zapisa broja x_{i-1}
$q_j\ 0\ R\ q_{j+2}$	
$q_{j+1}\ 0\ R\ q_j$	
$q_{j+2}\ 1\ R\ q_{j+2}$	prelazi zapis broja x_i
$q_{j+2}\ 0\ R\ q_{j+3}$	
$q_{j+3}\ 1\ 0\ q_{j+4}$	briše zapisa broja x_{i+1}
$q_{j+3}\ 0\ R\ q_{j+5}$	
$q_{j+4}\ 0\ R\ q_{j+3}$	
$\ldots$	

Funkcije projekcije, $f(x_1, \ldots, x_k) = x_i$ (2)

$\ldots$

q_l 1 0 q_{l+1}	briše zapisa broja x_k
q_l 0 L q_s	
q_{l+1} 0 R q_l	
q_s 0 L q_s	vraća se na početak zapisa broja x_i
q_s 1 L q_{s+1}	
q_{s+1} 1 L q_{s+1}	
q_{s+1} 0 R q_z	

Kako je svaki program konačan niz naredbi, a svaka naredba konačan niz simbola iz nekog prebrojivog skupa, to postoji samo prebrojivo mnogo programa. Kako svih aritmetičkih funkcija ima neprebrojivo mnogo, to znači da postoje funkcije koje nisu Tjuring-izračunljive.

Theorem

Tjuring-izračunljivih funkcija ima prebrojivo mnogo. Postoje funkcije koje nisu Tjuring-izračunljive.

- alfabet kojim se zapisuje sadržaj ćelija trake ne mora biti unarni,
- pored završnog stanja q_z uvode se i neka specijalna završna stanja, recimo q_{da} i q_{ne} koja, intuitivno, znače pozitivan, odnosno, negativan odgovor na postavljeni problem,
- dozvoljena je traka koja je beskonačna samo na jednu stranu, tj. postoji krajnja leva ćelija, dok se na desno traka pruža neograničeno,
- umesto samo jedne postoji više traka, a za svaku traku postoji posebna glava,
- nad jednom trakom postoji više glava umesto samo jedne,
- traka je dvodimenzionalna, a ne jednodimenzionalna, tj. traka podseća na beskonačnu šahovsku ploču,
- u jednoj naredbi mašine moguće je i upisati znak u ćeliju i pomerati glavu,
- ne važi zahtev za determinisanošću itd.

- Uglavnom, u smislu izračunljivosti gotovo sve od ovih varijanti Tjuringove mašine odgovaraju istoj klasi funkcija, tj. klasi Tjuring-izračunljivih funkcija, kao i osnovna verzija mašine.
- Izuzetak predstavljaju neki slabiji, restriktivni slučajevi: recimo, mašina čija traka je ograničena sa jedne strane i koristi unarni alfabet ili mašina koja ima samo dva stanja i koristi alfabet od dva znaka.
- Ekvivalencija varijanti Tjuringove mašine se dokazuje tako što se pokaže da za svaki program P za neku od varijanti Tjuringove mašine postoje programi za preostale varijante koji simuliraju izvršenje programa P i izračunavaju istu funkciju.

Tjuringova mašina sa bogatijim alfabetom

- reči bilo kog prebrojivog alfabeta se mogu prikazati pomoću unarnog alfabeta
- u većini slučajeva (mašine sa trakom koja nije ograničena ni sa jedne strane) alfabet može, zavisno od potrebe, slobodno određivati
- binarna, a ne kao do sada u unarnoj, reprezentacija prirodnih brojeva - ušteda (unarna reprezentacija prirodnih brojeva je eksponencijalno duža od binarne).

Tjuringova mašina sa trakom koja ima početak sa leve strane

- Alfabet sadrži još jedan specijalni znak, $\triangleright$
- služi da se prepozna početna ćelija sa leve strane
- preko $\triangleright$ se nikada ne prepisuje ni jedan drugi simbol, niti se glava sme pomeriti levo
- kada je $\triangleright$ u ćeliji ispod glave dozvoljene su jedino narede oblika $q_i \triangleright R\ q_j$

Tjuringova mašina sa više traka

- Za neki $k > 0$ Tjuringova mašina sa k traka sastoji se od traka označenih sa 1, 2, 3, ..., k
- Sve trake imaju kraj sa leve strane, što nije ograničenje
- Nad svakom trakom nalazi se posebna glava
- U svakom koraku čita se tekuća ćelija na svakoj od traka i preduzima odgovarajuća akcija

  $q_i \; s_1 \ldots s_k \; o_1 \ldots o_k \; q_j$
- Na početku rada ulazni podaci su na prvoj traci, ostale trake su prazne
- Ako izvršavanje mašine shvatimo kao izračunavanje neke funkcije, rezultat se smešta u k-tu traku.

Tjuringova mašina sa više traka

Example

Da li je neka reč palindrom (čitanjem sa leva na desno i sa desna na levo se dobija isti niz znaka)?

- reč iz prve trake se iskopira na drugu traku
- glava prve trake se vrati na levo
- glava druge ostaje u krajnjoj desnoj poziciji
- dve glave se kreću u suprotnim smerovima i ispituju da li se nalaze nad jednakim znacima.

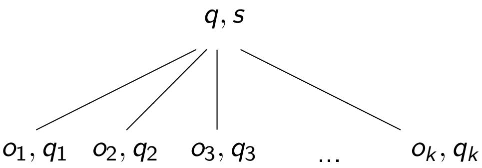
Tjuringova mašina sa više traka

Theorem

Za datu determinističku Tjuringovu mašinu M_1 *sa k-traka može se konstruisati deterministička Tjuringova mašina* M_2 *sa jednom trakom koja simulira rad mašine* M_1. *Dužina rada mašine* M_2 *je ograničena polinomijalnom funkcijom dužine rada mašine* M_1.

Nedeterministička Tjuringova mašina

- za tekuću konfiguraciju mašine može postojati više različitih konfiguracija do kojih se dolazi izvršavanjem neke od naredbi programa - mogućnost izbora
- grananje u drvetu je konačno - konačno mnogo opcija za izbor
- svaka grana - jedan mogući redosled rada



Nedeterministička Tjuringova mašina

- nedeterminističke mašine su pogodne za davanje odgovora 'da' ili 'ne' na pitanja oblika 'da li za ulazne podatke važi ...?'
- nova završna stanja: q_{da} i q_{ne}
- brzina nedeterminističkih je posledica asimetrične konvencije:
    - odgovor 'da' - bar jedno od mogućih izračunavanja završava u stanju q_{da},
    - odgovor 'ne' - sva izračunavanja završavaju u stanju q_{ne}
- ako ni jedno izračunavanje ne dovodi do stanja q_{da} i bar jedno izračunavanje ne dovodi ni do kog završnog stanja, nedeterministička mašina divergira.

Nedeterministička Tjuringova mašina

Model nedeterminističke mašine:

- višeprocesorski sistem
- u svakom koraku svaki od procesora kreira onoliko novih procesora koliko ima različitih konfiguracija u koje taj procesor može preći izvršavanjem tekuće naredbe
- ako mu bilo koji od potomaka vrati potvrdni odgovor, procesor tu informaciju šalje neposrednom pretku
- negativan odgovor se prosleđuje samo ako je dobijen od svih neposrednih potomaka
- svaki procesor izračunava disjunkciju odgovora svojih potomaka

Nedeterministička Tjuringova mašina

Example

Pretpostavimo da želimo ispitati da li je neki prirodan broj n složen ili prost. Običnom Tjuringovom mašinom problem bi se mogao rešiti na sledeći način: delili bismo broj svim prirodnim brojevima između 2 i $\frac{n}{2}$ i na osnovu toga dali odgovor. U slučaju nedeterminističke Tjuringove mašine na jednom mestu bismo imali mogućnost izbora broja kojim delimo broj n, pa ako je n složen, a izabrani broj delilac, mogli bismo dati odgovor u jednom koraku, što bi bio značajan dobitak u odnosu na deterministički postupak.

Nedeterministička Tjuringova mašina

Lako je uočiti da ovaj postupak nije realan, u smislu da izbor delioca podrazumeva da mi već znamo da je n složen, tj. da nam je poznat bar jedan njegov činilac. Međutim, i pored toga, nedeterministička Tjuringova mašina se može simulirati determinističkom mašinom, tako da se izražajnost u smislu onoga šta mašina može odgovoriti ne menja.

Nedeterministička Tjuringova mašina

M_1 nedeterministička TM, M_2 deterministička TM:

- M_2 sistematski prelazi sve moguće redoslede izvršavanja M_1
- najpre dužine 1, pa dužine 2 itd
- ni jedno moguće konačno izvršavanje neće biti preskočeno
- ako bi se M_1 u nekom trenutku našla u stanju q_{da}, to isto će pre ili posle biti slučaj i sa M_2
- ako sva izvršavanja M_1 dovode do stanja q_{ne}, i M_2 će se naći u tom stanju kada iscrpi sve mogućnosti
- ako M_1 divergira za date ulazne podatke ni M_2 se neće zaustaviti.

Nedeterministička Tjuringova mašina

- M_2 u najgorem slučaju bar jednom posećuje svaki čvor drveta koje prikazuje izvršavanje M_1
- čvorova drveta za M_1 može biti eksponencijalno više nego što je dužina najkraćeg mogućeg izračunavanja mašine M_1 koje dovodi do stanja q_{da}
- izvršavanje determinističke mašine M_2 u najgorem slučaju eksponencijalno duže nego izvršavanje nedeterminističke mašine M_1
- za sada nije poznato da li je simulaciju moguće izvesti uz samo polinomijalni gubitak vremena
- problem da li je $P = NP$

Nedeterministička Tjuringova mašina

Theorem

Za datu nedeterminističku Tjuringovu mašinu M_1 sa k-traka može se konstruisati deterministička Tjuringova mašina M_2 sa jednom trakom koja simulira rad mašine M_1. Dužina rada mašine M_2 je ograničena eksponencijalnom funkcijom dužine rada mašine M_1.

Za sada nije poznato da li je determinističku simulaciju rada nedeterminističkih mašina moguće izvesti u polinomijalnom vremenu.

Nedeterministička Tjuringova mašina

Theorem

Za datu nedeterminističku Tjuringovu mašinu M_1 *sa k-traka može se konstruisati deterministička Tjuringova mašina* M_2 *sa jednom trakom koja simulira rad mašine* M_1*. Dužina rada mašine* M_2 *je ograničena eksponencijalnom funkcijom dužine rada mašine* M_1*.*

Za sada nije poznato da li je determinističku simulaciju rada nedeterminističkih mašina moguće izvesti u polinomijalnom vremenu.

- *UTM* primer programibilnog digitalnog računara opšte namene sa programom i podacima smeštenim u memoriju
- simulira izvršavanje ostalih Tjuringovih mašina
- ulazni podaci za *UTM* su opis - program neke posebne mašine i ulazni podaci te mašine,
- rezultat izvršavanja *UTM* je rezultat rada simulirane posebne mašine
- zamisao o postojanju ovakve mašine Tjuring je konkretizovao: napisao je njen program.

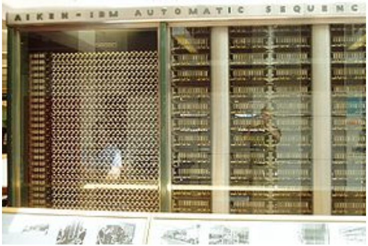
Univerzalna logička povezanost pojmova programa, podataka i automata koji izvršava dati program nad odgovarajućim podacima, potpuno revolucionarna u to vreme, predstavlja temelj savremenog računarstva:

- *"...Smatrao bih kao najčudniju koincidenciju na koju sam ikada naišao ako bi se ispostavilo da se osnovna logika mašine dizajnirane za numeričko rešavanje diferencijalnih jednačina polkapa sa logikom mašine koja proizvodi račune u nekoj robnoj kući ..."*,
  izjavio je 1956. godine Howard Aiken.

Univerzalna logička povezanost pojmova programa, podataka i automata koji izvršava dati program nad odgovarajućim podacima, potpuno revolucionarna u to vreme, predstavlja temelj savremenog računarstva:

- "…*Smatrao bih kao najčudniju koincidenciju na koju sam ikada naišao ako bi se ispostavilo da se osnovna logika mašine dizajnirane za numeričko rešavanje diferencijalnih jednačina polkapa sa logikom mašine koja proizvodi račune u nekoj robnoj kući* …",
izjavio je 1956. godine Howard Aiken.

Howard Aiken, 1900 – 1973



Howard Aiken, američki fizičar, jedan od pionira u konstrukciji računara, osnivač računarske laboratorije Univerziteta Harvard, konceptualni dizajner elektro-mehaničkog računara Automatic Sequence Controlled Calculator (kasnije nazvanog Harvard Mark I) 1944. godine.

- *"... Vratimo se sada na analogiju sa teorijskim mašinama za izračunavanje ... Može se pokazati da jedna specijalna mašina tog tipa može obavljati posao svih njih. U stvari, ona može raditi kao model bilo koje druge mašine. Ta specijalna mašina se može nazvati univerzalnom mašinom ..."*,
  izjavio je 1947. godine Alan Turing na predavanju u Londonskom Matematičkom društvu.

- induktivna definicija funkcije faktorijel (pomoću funkcije množenja):

  $0! =_{def} 1$
  $(n+1)! =_{def} (n!) \cdot (n+1)$

- opšti slučaj definisanja funkcije f na osnovu funkcije g:

  $f(0) = m$
  $f(n+1) = g(f(n), n)$

- Pod pretpostavkom da je funkcija g definisana, lako se vidi da je i funkcija f dobro definisana

  $f(0) = m$
  $f(1) = g(f(0), 0) = g(m, 0)$
  - $f(2) = g(f(1), 1) = g(g(m, 0), 1)$

  $f(n+1) = g(f(n), n) = \ldots = g(g(\ldots(g(m, 0), 1), \ldots, n-1), n)$

Definicija primitivno rekurzivnih funkcija

Klasa *primitivno rekurzivnih funkcija* sadrži osnovne funkcije:

- nula funkciju $Z(n) = 0$, za svako $n \in \mathbb{N}$,
- funkciju naslednika $S(n)$ prirodnog broja n u nizu prirodnih brojeva i
- funkcije projekcije $P^i_k(x_1, \ldots, x_k) = x_i$ za sve i i k takve da je $1 \leq i \leq k$

i sve funkcije koje se od njih dobijaju konačnim brojem primena osnovnih operacija:

- *kompozicije*, tj. ako su već definisane funkcije $g : \mathbb{N}^m \to \mathbb{N}$ i $h_1, \dots, h_m : \mathbb{N}^k \to \mathbb{N}$, definisana je i funkcija $f : \mathbb{N}^k \to \mathbb{N}$ tako da je za sve $(x_1, \dots, x_k) \in \mathbb{N}^k$:

$$f(x_1, \dots, x_k) = g(h_1(x_1, \dots, x_k), \dots, h_m(x_1, \dots, x_k))$$

  i
- *primitivne rekurzije*, tj. ako su već definisane funkcije $g : \mathbb{N}^m \to \mathbb{N}$ i $h : \mathbb{N}^{m+2} \to \mathbb{N}$, definisana je i funkcija $f : \mathbb{N}^{m+1} \to \mathbb{N}$ tako da je za sve $(x_1, \dots, x_m) \in \mathbb{N}^m$:
  - $f(0, x_1, \dots, x_m) = g(x_1, \dots, x_m)$ i
  - $f(n+1, x_1, \dots, x_m) = h(f(n, x_1, \dots, x_m), n, x_1, \dots, x_m)$ za $n \in \mathbb{N}$,.

  Pri tome je funkcija f definisana primitivnom rekurzijom nad h sa bazom g.

Primitivna rekurzija:

- $f(0, x_1, \ldots, x_m) = g(x_1, \ldots, x_m)$ i
- $f(n+1, x_1, \ldots, x_m) = h(f(n, x_1, \ldots, x_m), n, x_1, \ldots, x_m)$ za $n \in \mathbb{N}$,.

u slučaju kada je funkcija f unarna svodi na induktivnu definiciju:

- g je konstanta d,
- $f(0) = d$
- $f(n+1) = h(f(n), n)$.

- sve osnovne funkcije su totalne
- osnovne operacije primenjene na totalne funkcije takođe daju totalne funkcije
- primitivno rekurzivne funkcije su totalne
- primitivno rekurzivne funkcije Tjuring-izračunvljive (biće dokazano)

Primeri primitivno rekurzivnih funkcija

Da bi se za neku funkciju pokazalo da je primitivno rekurzivna potrebno je naći njen opis u smislu definicije - pokazati da je ona inicijalna ili da se iz inicijalnih funkcija dobija konačnim nizom primena operacija kompozicije i/ili primitivne rekurzije.

Example

Posmatramo konstantne funkcije $K_k^j : \mathbb{N}^k \to \mathbb{N}$ za koje je $K_k^j(x_1, \dots, x_k) = j$ za sve $(x_1, \dots, x_k) \in \mathbb{N}^k$

- $k = 1$ - funkcije su oblika $K_1^j : \mathbb{N} \to \{j\}$, $K_1^j(x) = S^j(Z(x))$
- $k \geq 1$ - indukcijom, K_k^j je definisano
  $K_{k+1}^j(0, x_1, \dots, x_k) = K_k^j(x_1, \dots, x_k)$
  $K_{k+1}^j(n+1, x_1, \dots, x_k) = P_{k+2}^1(K_{k+1}^j(n, x_1, \dots, x_k), n, x_1, \dots, x_k)$
- mogu'ce i: $K_k^j(x_1, \dots, x_k) = S^j(Z(P_k^1(x_1, \dots, x_k)))$

Primeri primitivno rekurzivnih funkcija

Example

Funkcija sabiranja $f(n,x) = n + x$ je primitivno rekurzivna.

- $S(P_3^1(x,y,z))$ je primitivno rekurzivna jer se dobija kompozicijom osnovnih funkcija naslednika i projekcije
- funkciju sabiranja možemo definisati sa:

  $f(0,x) = P_1^1(x)$
  $f(n+1,x) = S(P_3^1(f(n,x),n,x))$.
- f je definisana primitivnom rekurzijom nad $S(P_3^1())$ sa bazom $P_1^1()$

Primeri primitivno rekurzivnih funkcija

Example

Funkcija prethodnik definisana sa

$$\mathrm{preth}(x) = \begin{cases} 0 & \text{za } x = 0 \\ \text{prethodnik od } x \text{ u nizu prirodnih brojeva} & \text{za } x > 0 \end{cases}$$

je primitivno rekurzivna:

$$\mathrm{preth}(0) = 0$$
$$\mathrm{preth}(n+1) = P^2_2(\mathrm{preth}(n), n).$$

Eksplicitna transformacija

Definition

Neka je $g : \mathbb{N}^k \to \mathbb{N}$ k-arna funkcija i neka su $h_1, \ldots, h_k : \mathbb{N}^l \to \mathbb{N}$ l-arne funkcije od kojih je svaka ili projekcija ili konstantantna funkcija. Funkcija f definisana sa:

$$f(x_1, \ldots, x_l) = g(h_1(x_1, \ldots, x_l), \ldots, h_k(x_1, \ldots, x_l))$$

je dobijena iz funkcije g *eksplicitnom transformacijom*.

Theorem

Ako je funkcija g primitivno rekurzivna, primitivno rekurzivne su i sve funkcije dobijene iz nje eksplicitnom transformacijom.

Eksplicitna transformacija - dupliranje i/ili permutovanje argumenata i zamena argumenata konstantama, pa je to jedna vrsta kompozicije.

Eksplicitna transformacija

Example

Neka je funkcija $g : \mathbb{N}^2 \to \mathbb{N}$ primitivno rekurzivna. Tada je i funkcija

$$f(x, y, z) = g(z, x)$$

primitivno rekurzivna jer je dobijena eksplicitnom transformacijom iz primitivno rekurzivne funkcije g.

Eksplicitna transformacija

Example

Funkcija *monus* (ograničeno oduzimanje):

$$\dot{-}(n, x) = \begin{cases} 0 & \text{za } x \leq n \\ x - n, & \text{za } x > n \end{cases}$$

je primitivno rekurzivna:

$$\dot{-}(0, x) = P_1^1(x)$$
$$\dot{-}(n + 1, x) = h(\dot{-}(n, x), n, x)$$
$$h(x, y, z) = \mathrm{preth}(P_3^1(x, y, z)), \text{ prethodnik prve projekcije}$$

h je primitivno rekurzivna jer je dobijena eksplicitnom transformacijom iz primitivno rekurzivne funkcije preth.

Napomenimo da funkcija $x - y$ nije totalna, pa nije ni primitivno rekurzivna, zbog čega se i razmatra funkcija monus.

Ograničena suma

Definition

Ograničena suma je funkcija oblika $f_0(x) + \ldots + f_n(x)$ za koju ćemo koristiti oznaku $\sum_{i=0}^{n} f_i(x)$.

Theorem

Neka su funkcije $f_0, \ldots, f_n$ primitivno rekurzivne. Tada je i ograničena suma $\sum_{i=0}^{n} f_i(x)$ primitivno rekurzivna funkcija.

- $n = 0$, $\sum_{i=0}^{0} f_i(x) = f_0(x)$ je trivijalno primitivno rekurzivna funkcija
- za sve $k < m$ tvrđenje dokazano
- $\sum_{i=0}^{m} f_i(x) = +(\sum_{i=0}^{m-1} f_i(x), f_m(x))$ primitivno rekurzivna funkcija jer je dobijena kompozicijom primitivno rekurzivnih funkcija

Ograničeni proizvod

Definition

Ograničeni proizvod je funkcija oblika $f_0(x) \cdot \ldots \cdot f_n(x)$ za koju ćemo koristiti oznaku $\Pi_{i=0}^{n} f_i(x)$.

Theorem

Neka su funkcije $f_0, \ldots, f_n$ *primitivno rekurzivne. Tada je i ograničeni proizvod* $\Pi_{i=0}^{n} f_i(x)$ *primitivno rekurzivna funkcija.*

Primitivno rekurzivni predikati

Predikat je relacija, odnosno neki podskup skupa $\mathbb{N}^k$ za neki prirodan broj $k > 0$. Unarni predikat R je primitivno rekurzivan ako je primitivno rekurzivna njegova karakteristična funkcija

$$C_R(n) = \begin{cases} 1 & \text{ako važi } R(n) \\ 0 & \text{ako ne važi } R(n) \end{cases}$$

i slično za predikate proizvoljne arnosti.

Primitivno rekurzivni predikati

Example

Predikat $=$, tj. relacija jednakosti, je primitivno rekurzivan jer je primitivno rekurzivna njegova karakteristična funkcija
$C_{=}(x, y) = \dot{-}(\mathrm{sgn}(\dot{-}(x, y) + \dot{-}(y, x)), 1)$.
Domaći: Pokazati da je funkcija sgn primitivno rekurzivna.

Example

Predikat $<$, tj. relacija biti manji, je primitivno rekurzivan jer je primitivno rekurzivna njegova karakteristična funkcija $C_{<}(x, y) = \mathrm{sgn}(\dot{-}(x, y))$.

Primitivno rekurzivni predikati

Definition

Neka su P i Q dva unarna predikata.
Konjunkcija predikata P i Q, u oznaci $\mathrm{P} \wedge \mathrm{Q}$, je unarni predikat:

$$\mathrm{C}_{\mathrm{P}\wedge\mathrm{Q}}(\mathrm{n}) = \begin{cases} 1 & \text{ako važi } \mathrm{P}(\mathrm{n}) \text{ i } \mathrm{Q}(\mathrm{n}) \\ 0 & \text{ako ne važi } \mathrm{P}(\mathrm{n}) \text{ i } \mathrm{Q}(\mathrm{n}) \end{cases}$$

Disjunkcija predikata P i Q, u oznaci $\mathrm{P} \vee \mathrm{Q}$, je unarni predikat:

$$\mathrm{C}_{\mathrm{P}\vee\mathrm{Q}}(\mathrm{n}) = \begin{cases} 1 & \text{ako važi } \mathrm{P}(\mathrm{n}) \text{ ili } \mathrm{Q}(\mathrm{n}) \\ 0 & \text{ako ne važi } \mathrm{P}(\mathrm{n}) \text{ ili } \mathrm{Q}(\mathrm{n}) \end{cases}$$

Negacija predikata P, u oznaci $\neg\mathrm{P}$, je unarni predikat:

$$\mathrm{C}_{\neg\mathrm{P}}(\mathrm{n}) = \begin{cases} 1 & \text{ako ne važi } \mathrm{P}(\mathrm{n}) \\ 0 & \text{ako važi } \mathrm{P}(\mathrm{n}) \end{cases}$$

Primitivno rekurzivni predikati

Theorem

Ako su P *i* Q *dva primitivno rekurzivna predikata, tada su primitivno rekurzivni i predikati* $\mathrm{P} \wedge \mathrm{Q}$, $\mathrm{P} \vee \mathrm{Q}$, $\neg\mathrm{P}$.

Neka su predikati P i Q unarni.

- $\mathrm{C}_{\mathrm{P}\wedge\mathrm{Q}}(\mathrm{n}) = \mathrm{C}_{\mathrm{P}}(\mathrm{n}) \cdot \mathrm{C}_{\mathrm{Q}}(\mathrm{n})$
- $\mathrm{C}_{\mathrm{P}\vee\mathrm{Q}}(\mathrm{n}) = \mathrm{sgn}(\mathrm{C}_{\mathrm{P}}(\mathrm{n}) + \mathrm{C}_{\mathrm{Q}}(\mathrm{n}))$
- $\mathrm{C}_{\neg\mathrm{P}}(\mathrm{n}) = \dot{-}(\mathrm{C}_{\mathrm{P}}(\mathrm{n}), 1)$

Tvrđenje se slično dokazuje i za k-arne predikate i za n-arne konjunkcije i disjunkcije.

Definicija po slučaju

Funkcija f je definisana po slučaju:

$$f(n) = \begin{cases} h_0(n) & \text{ako važi } A_0(n) \\ \dots & \\ h_m(n) & \text{ako važi } A_m(n) \end{cases}$$

$A_0, \dots, A_m$ su unarni primitivno rekurzivni predikati takvi da je za svaki prirodan broj n ispunjen tačno jedan od njih. Funkcije $h_0, \dots h_m$ su primitivno rekurzivne. Analogno se po slučaju definišu k-arne funkcije.

Theorem

Funkcija f definisana po slučaju u odnosu na predikate $A_1, \dots, A_m$ i funkcije $h_1, \dots h_m$ je primitivno rekurzivna.

$f(x_1, \dots, x_k) = \sum_{i=0}^{m} h_i(x_1, \dots, x_k) \cdot C_{A_i}(x_1, \dots, x_k)$

Primer definicije po slučaju: funkcija se definiše eksplicitnim zadavanjem vrednosti za prvih l prirodnih brojeva, a ostatak koristeći neku drugu primitivno rekurzivnu funkciju.

Ograničena kvantifikacija

Definition

Neka je k prirodan broj i R jedan $k+1$-arni predikat.
Ograničena egzistencijalna kvantifikacija predikata R je $k+1$-arni predikat Q za koji važi $Q(x_1, \ldots, x_k, m)$ ako i samo ako postoji prirodan broj i, takav da je $0 \leq i \leq m$ i važi $R(x_1, \ldots, x_k, i)$. Oznaka za $Q(x_1, \ldots, x_k, m)$ je $(\exists i \leq m)R(x_1, \ldots, x_k, i)$.
Ograničena univerzalna kvantifikacija predikata R je $k+1$-arni predikat Q za koji važi $Q(x_1, \ldots, x_k, m)$ ako i samo ako za svaki prirodan broj i, takav da je $0 \leq i \leq m$ i važi $R(x_1, \ldots, x_k, i)$. Oznaka za $Q(x_1, \ldots, x_k, m)$ je $(\forall i \leq m)R(x_1, \ldots, x_k, i)$.

Ograničena kvantifikacija

Theorem

Ako je R primitivno rekurzivni $\mathrm{k}+1$-arni predikat, primitivno rekurzivni su i predikati ograničene egzistencijalne i univerzalne kvantifikicije za R.

Primitivno rekurzivna je karakteristična funkcija predikata ograničene univerzalne kvantifikicije:

$$\mathrm{C}_{\forall \mathrm{i} \leq}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k}, \mathrm{m}) = \Pi_{\mathrm{i}=0}^{\mathrm{m}} \mathrm{C}_\mathrm{R}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k}, \mathrm{i}).$$

$(\exists \mathrm{i} \leq \mathrm{m})\mathrm{R}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k}, \mathrm{i})$ važi ako i samo ako važi $\neg(\forall \mathrm{i} \leq \mathrm{m})\neg\mathrm{R}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k}, \mathrm{i})$, to se karakteristična funkcija predikata ograničene egzistencijalne kvantifikicije zapisuje kao

$$\mathrm{C}_{\exists \mathrm{i} \leq}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k}, \mathrm{m}) = \dot{-}(\Pi_{\mathrm{i}=0}^{\mathrm{m}}(\dot{-}(\mathrm{C}_\mathrm{R}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k}, \mathrm{i}), 1), 1),$$

pa je i on primitivno rekurzivan.

Ograničena kvantifikacija

Example

$P(x, y, z, t)$ važi ako i samo ako $x^t + y^t = z^t$. On je primitivno rekurzivan jer se karakteristična funkcija C_P dobija kompozicijom primitivno rekurzivnih funkcija stepenovanja i karakteristične funkcije $C_=$.
Neka je $m \in \mathbb{N}$ fiksiran prirodni broj. Binarni predikat Q definisan tako da važi $Q(k, m)$ ako i samo ako

$$x^k + y^k = z^k$$

za bar neke $x, y, z \leq m$ je primitivno rekurzivan jer se definiše sa

$$(\exists x \leq m)(\exists y \leq m)(\exists z \leq m)P(x, y, z, k)$$

Ograničena kvantifikacija

Example

Neka predikat $\mathrm{neparno}(\mathrm{n})$ važi ako i samo ako je n neparan broj. $\mathrm{neparno}(\mathrm{n})$ je primitivno rekurzivan jer važi ako i samo ako važi

$$(\exists \mathrm{m} \leq \mathrm{n})(\mathrm{n} = 2 \cdot \mathrm{m} + 1).$$

Predikat $\mathrm{parno}(\mathrm{n})$, tj. biti paran, je primitivno rekurzivan jer važi ako i samo ako važi

$$(\exists \mathrm{m} \leq \mathrm{n})(\mathrm{n} = 2 \cdot \mathrm{m})$$

Ograničena kvantifikacija

Example

Neka predikat $\div(\mathrm{m}, \mathrm{n})$ važi ako i samo m deli n bez ostatka.
$\div(\mathrm{m}, \mathrm{n})$ je primitivno rekurzivan jer važi ako i samo ako važi

$$(\exists \mathrm{k} \leq \mathrm{n})(\mathrm{n} = \mathrm{k} \cdot \mathrm{m}).$$

Predikat $\mathrm{prost}(\mathrm{n})$, tj. biti prost broj, je primitivno rekurzivan jer važi ako i samo ako važi

$$(1 < \mathrm{n}) \wedge (\forall \mathrm{m} \leq \mathrm{n})((\mathrm{m} = 1) \vee (\mathrm{m} = \mathrm{n}) \vee (\neg \div (\mathrm{m}, \mathrm{n}))$$

Ograničena minimizacija

Definition

Neka je k prirodan broj i R jedan $\mathrm{k}+1$-arni predikat. *Ograničena minimizacija* za predikat R je $\mathrm{k}+1$-arna funkcija $\mathrm{f}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k}, \mathrm{n})$ definisana sa

$$(\mu \mathrm{p} \leq \mathrm{n})\mathrm{R}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k}, \mathrm{p}) = \begin{cases} \begin{array}{l} \text{najmanji prirodni broj } \mathrm{p} \leq \mathrm{n} \\ \text{za koji je } \mathrm{R}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{k}, \mathrm{p}) \\ \text{ako takav postoji} \end{array} & \\ 0 & \text{inače} \end{cases}$$

Theorem

Neka je k *prirodan broj i* R *jedan* $\mathrm{k}+1$*-arni primitivno rekurzivni predikat. Ograničena minimizacija za predikat* R *je primitivno rekurzivna funkcija.*

Ograničena minimizacija

$$h(p,n,x_1,\ldots,x_k)=\begin{cases} p & \text{ako } (\exists i\leq n)R(x_1,\ldots,x_k,i) \\ n+1 & \text{ako} \\ & R(x_1,\ldots,x_k,n+1)\wedge\neg(\exists i\leq n)R(x_1,\ldots,x_k,i) \\ 0 & \text{ako nije } (\exists i\leq n+1)R(x_1,\ldots,x_k,i) \end{cases}$$

je primitivno rekurzivna - definisana je po slučaju preko primitivno rekurzivnih predikata i funkcija.

Funkcija $(\mu p\leq n)R(x_1,\ldots,x_k,p)$ je definisana primitivnom rekurzijom:

$$(\mu p\leq 0)R(x_1,\ldots,x_k,p)=0$$

$$(\mu p\leq n+1)R(x_1,\ldots,x_k,p)=h((\mu p\leq n)R(x_1,\ldots,x_k,p),n,x_1,\ldots,x_k)$$

Ograničena minimizacija

U drugom koraku ove definicije, vrednost funkcije je minimalno $p \leq n$ za koje važi $R(x_1, \dots, x_k, p)$, ako takvo postoji ili $n + 1$, ako takvo p ne postoji, a važi $R(x_1, \dots, x_k, n + 1)$ ili 0, ako ni za jedno $p \leq n + 1$ nije $R(x_1, \dots, x_k, p)$. Pošto je h primitivno rekurzivna funkcija, takva je i funkcija ograničene minimizacije $(\mu p \leq n)R(x_1, \dots, x_k, p)$.

Ograničena minimizacija

Example

Neka je vrednost funkcije $\mathrm{p}(\mathrm{n})$ jednaka n-tom prostom broju.
Pokazano je da je predikat $\mathrm{prost}(\mathrm{n})$, tj. biti prost broj, primitivno rekurzivan.
Funkciju $\mathrm{h}(\mathrm{n}, \mathrm{y}) = (\mu \mathrm{x} \leq (!(\mathrm{n}) + 1))((\mathrm{n} < \mathrm{x}) \wedge \mathrm{prost}(\mathrm{x}))$ je primitivno rekurzivna (Koristi se poznato tvrđenje: ako je p prost broj, onda između p i $\mathrm{p}! + 1$ postoji bar još jedan prost broj).
Funkciju $\mathrm{p}(\mathrm{n})$ definišemo primitivnom rekurzijom

$$\mathrm{p}(0) = 2,$$

$$\mathrm{p}(\mathrm{n} + 1) = \mathrm{h}(\mathrm{p}(\mathrm{n}), \mathrm{n}).$$

Ograničena minimizacija

Example

Neka funkcija $[x]_n$ izračunava stepen n-tog prostog broja u dekompoziciji broja x. Ona je primitivno rekurzivna jer je

$$[x]_n = (\mu y \leq x)(\neg \div (p(n)^{y+1}, x)),$$

tj. $[x]_n$ je najmanje y manje od x sa osobinom da $y + 1$-ti stepen n-tog prostog broja ne deli x.

Ograničena minimizacija

Example

Funkcija dužina dekompozicije prirodnog broja $\mathrm{duzina}(x)$ daje redni broj najmanjeg prostog broja čiji je stepen u dekompoziciji broja x jednak 0. I ova funkcija je primitivno rekurzivna jer se definiše sa

$$\mathrm{duzina}(x) = (\mu n \leq x)([x]_n = 0).$$

Na primer, kako je u nizu prostih brojeva $p(0) = 2$, $p(1) = 3$, ..., za $x = 280 = 2^3 \cdot 5^1 \cdot 7^1$ je stepen uz 3, $[x]_1$, jednak nula, pa je $\mathrm{duzina}(x) = 1$.

Ograničena minimizacija

Primitivno rekurzivne su funkcije:

- $\mathrm{p(n)}$
- $[\mathrm{x}]_{\mathrm{n}}$
- $\mathrm{duzina(x)}$

U proučavanju modela izračunavanja postupak kodiranja igra značajnu ulogu.
Primeri kodiranja:

- ASCII-znaci se kodiraju brojevima između 0 i 127 (255 kod takozvanog proširenog skupa ASCII-znaka), a zatim se reči zapisuju kao nizovi brojeva, tj. brojevi u nekoj pozicionoj notaciji. Na primer, znak 'a' se kodira sa 97, a znak '*' sa 76. Reč 'a*' se kodira sa $256^1 \cdot 97 + 76$
- prvođenje izvornih u izvršne programe.

- kodiranje objektima (funkcije ili programi) pridružuje prirodne brojeve.
- umesto polaznih objekata sredstvima aritmetike se razmatraju njihove brojevne slike
- neka funkcija, tj. njena slika - kod, može da se javi kao argument neke funkcije, pa i same sebe.

Gedelizacija je kodiranje koje ispunjava:

- ni koja dva objekta nemaju isti pridruženi broj,
- za svaki objekat možemo u konačnom broju koraka izračunati njemu pridruženi broj i
- za dati prirodan broj može se u konačnom broju koraka proveriti da li je to broj nekog objekta i naći taj objekat, ako postoji.

Kod pridružen objektu - *Gedelov broj*.
Ako kodiranje zadovoljava prethodne uslove, onda sama njegova forma nije bitna, jer tvrđenja i dokazi ne zavise od nje.

Nešto kasnije biće detaljno opisan postupak kodiranja jedne klase funkcija koja sadrži sve primitivno rekurzivne funkcije.
Odatle, primitivno rekurzivnih funkcija, a time i unarnih primitivno rekurzivnih funkcija, ima prebrojivo mnogo, te ih je moguće poređati u niz.
Pozicija u nizu je *indeks* funkcije.

U kodiranju koje ćemo koristiti značajni su prosti brojevi i jedinstvenost prikazivanja svakog broja u obliku proizvoda prostih brojeva dignutih na neki stepen.

$$270 = 2^1 \cdot 3^3 \cdot 5^1.$$

Od pomoćnih funkcija upotrebićemo primitivno rekurzivne funkcije:

- $\mathrm{p}(\mathrm{n})$
- $[\mathrm{x}]_\mathrm{n}$
- $\mathrm{duzina}(\mathrm{x})$

- niz brojeva $(\mathrm{a}_0, \mathrm{a}_1, \ldots, \mathrm{a}_\mathrm{n})$
- kod niza je:
  $\mathrm{gb}(\mathrm{a}_0, \mathrm{a}_1, \ldots, \mathrm{a}_\mathrm{n}) = \mathrm{p}(0)^{\mathrm{a}_0+1} \cdot \mathrm{p}(1)^{\mathrm{a}_1+1} \cdot \ldots \cdot \mathrm{p}(\mathrm{n})^{\mathrm{a}_\mathrm{n}+1}$
- gb je primitivno rekurzivna funkcija
- a_i koji se nalazi na i-tom mestu u nizu:
  - kodira se sa: $p(i)^{a_i+1}$
  - dekodira se sa: $(x)_i = \dot{-}(1, [x]_i)$
    za 1 umanjen stepen n-tog prostog broja u dekompoziciji
- Razlog: množenje sa $p(n)^0 = 1$ ne menja vrednost proizvoda, a komplikuje dekompoziciju.

- dogovor: x kodira sekvencu dužine do $duzina(x)$
- x kodira sekvencu $((x)_0, \ldots, (x)_{duzina(x)-1})$
- x za koji je $duzina(x) = 0$ kodira praznu reč
- $x = 756 = 2^2 \cdot 3^3 \cdot 7 = p(0)^2 \cdot p(1)^3 \cdot p(3)$
- $duzina(x) = 2$, pa x kodira $(1, 2)$
- kako je $gb(1, 2) = 108$, razni brojevi mogu predstavljati isti niz brojeva
- različiti nizovi se uvek kodiraju raznim brojevima

- posledica mogućnosti nabrajanja svih primitivno rekurzivnih funkcija - možemo zamisliti intuitivno izračunljivu totalnu funkciju koja nije primitivno rekurzivna
- postupak se naziva *dijagonalizacija*
- posmatrajmo samo unarne primitivno rekurzivne funkcije i bilo koje njihovo nabrajanje: $f_0, f_1, f_2, \dots$
- neka je $g(x) = f_x(x) + 1$
- totalna i intuitivno izračunljiva (za x konstruišemo f_x, izračunamo $f_x(x)$ i dodamo 1)
- ako je g primitivno rekurzivna, ona ima neki indeks k u nabrajanju
- $f_k(k) = g(k) = f_k(k) + 1$

Akermanova funkcija

$$A(0, n, k) = n + k$$
$$A(1, n, 0) = 0$$
$$A(2, n, 0) = 1$$
$$A(m + 1, n, 0) = n, \text{ za } m > 1,$$
$$A(m + 1, n, k + 1) = A(m, n, A(m + 1, n, k)), \text{ za } k \geq 0$$

je intuitivno izračunljiva, ali se pokazuje da nije primitivno rekurzivna.

Za funkciju A važi:

- $A(0, n, k) = n + k$
- $A(1, n, k) = n \cdot k$
- $A(2, n, k) = n^k$
- dijagonala ove funkcije $A(n, n, n)$ raste brže od svake unarne primitivno rekurzivne funkcije
- A nije primitivno rekurzivna funkcija.

- rekurzija u Akermanovoj funkciji sprovodi istovremeno po dva argumenta - prvom i trećem
- istovremenom rekurzijom po tri, četiri itd. argumenta dobijaju redom sve šire i šire klase funkcija koje su intuitivno izračunljive
- klasa funkcija koje se definišu rekurzijom po n argumenata je pravi podskup klase funkcija definisanih rekurzijom nad $n+1$ argumentom, za svaki $n \geq 0$
- klasa intutivno izračunljivih funkcija nije podskup ni jedne klase u toj hijerarhiji.

Iako se u klasi primitivno rekurzivnih funkcija nalazi veliki broj interesantnih funkcija, postoje funkcije koje jesu intuitivno izračunljive a koje toj klasi ne pripadaju.

Parcijalno rekurzivne funkcije

Jedno od ograničenja koje u sebi ima klasa primitivno rekurzivnih funkcija je da je zatvorena za operaciju minimizacije koja sme biti samo ograničena. Klasa parcijalno rekurzivnih funkcija se dobija kada se eliminiše to ograničenje. Pokazaćemo da je to dovoljno da se klasa funkcija proširi toliko da se poklopi sa Tjuring-izračunljivim funkcijama.

Definicija parcijalno rekurzivnih funkcija

Klasa *parcijalno rekurzivnih funkcija* sadrži osnovne funkcije:

- nula funkciju $Z(n) = 0$, za svako $n \in \mathbb{N}$,
- funkciju naslednika $S(n)$ prirodnog broja n u nizu prirodnih brojeva i
- funkcije projekcije $P^i_k(x_1, \ldots, x_k) = x_i$ za sve i i k takve da je $1 \leq i \leq k$

i sve funkcije koje se od njih dobijaju konačnim brojem primena osnovnih operacija:

- kompozicije
- primitivne rekurzije i
- *neograničene minimizacije*, tj. ako je već definisana $k+1$-arna funkcija f, definisana je i funkcija

$$(\mu y)(f(x_1,\dots,x_k,y)=0)=\begin{cases}\text{najmanje } z, \text{ za koje je} \\ f(x_1,\dots,x_k,z)=0 \text{ i} \\ \text{za svako } y<z \text{ je} \\ f(x_1,\dots,x_k,y) \text{ definisano} \\ \\ \text{nedefinisano, inače.}\end{cases}$$

Neograničene minimizacija

- $(\mu y)(f(x_1, \dots, x_k, y) = 0)$ je funkcija arnosti k čiji argumenti su x_1, $\dots$, x_k,
- intuitivno izračunljiva funkcija
- računa se $f(x_1, \dots, x_k, 0)$
- ako je f definisana za te argumente i ima vrednost 0, rezultat je 0
- ako je f definisana za te argumente, a vrednost joj nije 0, prelazi se na $f(x_1, \dots, x_k, 1)$
- ako f nije definisana za te argumente, ni $(\mu y)(f(x_1, \dots, x_k, y) = 0)$ neće biti definisana.

Neograničene minimizacija

Example

Funkcija $(\mu y)(\frac{x}{y} - 1 = 0)$ nije definisana ni za jedno x pošto nije definisano ni deljenje nulom.

- prema definiciji klasa parcijalno rekurzivnih funkcija sadrži sve primitivno rekurzivne funkcije,
- u klasi parcijalno rekurzivnih funkcija se nalaze i funkcije koje nisu totalne, pa samim tim nisu ni primitivno rekurzivne
- postoje i totalne funkcije koje nisu primitivno rekurzivne, a koje jesu parcijalno rekurzivne (Akermanova funkcija)

Definition

Rekurzivne funkcije su totalne parcijalno rekurzivne funkcije.

Definition

Predikat R je *rekurzivan (odlučiv)* ako je njegova karakteristična funkcija C_R rekurzivna.

Oznake:

- $f(x_1, \ldots, x_k) \downarrow$ (f konvergira)
- $f(x_1, \ldots, x_k) \downarrow y$ (f konvergira ka y)
- $f(x_1, \ldots, x_k) \uparrow$ (f divergira)

Definition

Dve parcijalno rekurzivne funkcije f i g su *jednake* ($f \simeq g$) ako su iste arnosti i za iste arumente ili obe nisu definisane, ili su obe definisane i imaju istu vrednost.

Parcijalno rekurzivne operacije

Theorem

Ako je predikat $R \subset \mathbb{N}^{k+1}$ rekurzivan, a h $k+1$-arna parcijalno rekurzivna funkcija, parcijalno rekurzivne su i sledeće funkcije:

- $(\mu y)(h(x_1, \dots x_k, y) = a)$, *za* $a \in \mathbb{N}$ *i*
- $(\mu y)(R(x_1, \dots x_k, y))$.

U prvom slučaju umesto $h(x_1, \dots x_k, y)$ dovoljno je posmatrati funkciju $sgn(\dot{-}(h(x_1, \dots x_k, y), a) + \dot{-}(a, h(x_1, \dots x_k, y))$ i tvrđenje sledi neposredno. U slučaju funkcije $(\mu y)(R(x_1, \dots x_k, y))$, umesto predikata $R(x_1, \dots x_k, y)$ posmatra se jednakost $C_R(x_1, \dots x_k, y) = 1$.

Izračunljive i parcijalno rekurzivne funkcije

- Gedelizacija je iskorištena da bi se primenio postupak dijagonalizacije i pokazalo da postoje intuitivno izračunljive funkcije koje nisu primitivno rekurzivne
- i parcijalno rekurzivne funkcije su nabrojive
- dijagonalizacija ne prolazi
- može se definisati $g(x) = f_x(x) + 1$, ali $f_x(x)$ ne mora biti definisano
- problem da li je funkcija $f_x(x) \downarrow$ se ne može rešiti u opštem slučaju

Izračunljive i parcijalno rekurzivne funkcije

Theorem

Ne postoji rekurzivna funkcija definisana sa

$$g(x) = \begin{cases} 1 & \text{ako je } f_x(x) \text{ definisano} \\ 0 & \text{ako } f_x(x) \text{ nije definisano.} \end{cases}$$

Izračunljive i parcijalno rekurzivne funkcije

- $f_0, f_1, \ldots$- niz svih unarnih parcijalno rekurzivnih funkcija
- g je moguće definisati
- $$h(x) = \begin{cases} \text{nedefinisana} & \text{ako je } f_x(x) \text{ definisano} \\ 0 & \text{ako } f_x(x) \text{ nije definisano} \end{cases}$$
- h je parcijalno rekurzivna funkcija oblika $(\mu y)(y + g(x) = 0)$
- h je f_l za neko l
- kontradikcija

$$h(l) = \begin{cases} \text{nedefinisano} & \text{ako je } h(l) = f_l(l) \text{ definisano} \\ 0 & \text{ako } h(l) = f_l(l) \text{ nije definisano} \end{cases}$$

Nabrajanje parcijalno rekurzivnih funkcija

Koristi se:

- funkcije: $p(n)$, $[x]_n$, $duzina(x)$
- kod niza brojeva $(a_0, a_1, \ldots, a_n)$ je:
  $gb(a_0, a_1, \ldots, a_n) = p(0)^{a_0+1} \cdot p(1)^{a_1+1} \cdot \ldots \cdot p(n)^{a_n+1}$
- a_i koji se nalazi na i-tom mestu u nizu:
  - kodira se sa: $p(i)^{a_i+1}$
  - dekodira se sa: $(x)_i = \dot{-}(1, [x]_i)$
    za 1 umanjen stepen n-tog prostog broja u dekompoziciji

Nabrajanje parcijalno rekurzivnih funkcija

- kod osnovnih funkcija: $gb(Z) = 0$, $gb(S) = gb(1)$, $gb(P^i_k) = gb(1, i)$,
- ako su $gb(g) = a$, $gb(h_1) = b_1$, ..., $gb(h_m) = b_m$ već određeni, kod kompozicije je $gb(g(h_1, \ldots, h_m)) = gb(a, gb(b_1, \ldots, b_m), 0)$,
- ako su $gb(g) = a$ i $gb(h) = b$ već određeni, onda je $gb(a, b, 0, 0)$ kod fje dobijene primitivnom rekurzijom nad h sa osnovom g
- ako je $gb(g) = a$ već određen, onda je $gb(a, 0, 0, 0, 0)$ kod funkcije dobijene neograničenom minimizacijom $(\mu y)(g(x_1, \ldots, x_k, y) = 0)$

Nabrajanje parcijalno rekurzivnih funkcija

- broj promenljivih ne figuriše u samom kodiranju, određuje prilikom dekodiranja
- različitim funkcijama i operacijama pridruženi su kodovi nizova različite dužine
- dogovor pri dekodiranju - broj x kodira sekvencu dužine do $duzina(x)$, pa beskonačno mnogo različitih brojeva označava istu funkciju
- treba obezbediti da je svaki prirodan broj kod neke funkcije,
- na primer: ako je $duzina(x) \geq 5$, onda x kodira funkciju dobijenu neograničenom minimizacijom funkcije $f_{(x)_0}$.

Nabrajanje parcijalno rekurzivnih funkcija

Ideja za dekodiranje:

- koristi se jedinstvenost dekompozicije na proste faktore
- $x = gb(3276, 817, 0, 0)$
- ako su $gb(g) = a$ i $gb(h) = b$ već određeni, onda je $gb(a, b, 0, 0)$ kod fje dobijene primitivnom rekurzijom nad h sa osnovom g
- funkcija f_x je dobijena primenom primitivne rekurzije na f_{817} sa bazom f_{3276}.
- dekodiraje kodova 817 i 3276 se nastavlja dok se ne stigne do osnovnih funkcija

Nabrajanje parcijalno rekurzivnih funkcija

Funkcija arnosti k čiji je indeks n označava se sa f_n^k.

- kodiranje: $gb(Z) = 0$,
- dekodiranje: $duzina(n) = 0$, tj. n kodira prazni niz brojeva, reč je o nula-funkciji Z,
- kodiranje: $gb(S) = gb(1)$,
- dekodiranje: $duzina(n) = 1$, tj. n kodira jednočlani niz brojeva $(n)_0$, reč je o funkciji naslednika S

Nabrajanje parcijalno rekurzivnih funkcija

Funkcija arnosti k čiji je indeks n označava se sa f_n^k.

- kodiranje: $gb(P_k^i) = gb(1, i)$,
- dekodiranje: $duzina(n) = 2$, tj. n kodira niz brojeva $((n)_0, (n)_1)$ i ako je:
  - $(n)_0 = 0$, ili se projektuje po koordinati $(n)_1 = 0$, reč je o nula-funkciji Z,
  - $(n)_0 \geq 1$ i projektuje se po koordinati $1 \leq (n)_1 \leq k$, reč je o funkciji $P_k^{(n)_1}$ i
  - $(n)_0 \geq 1$ i projektuje se po koordinati $(n)_1 > k$, reč je o funkciji P_k^k,

Nabrajanje parcijalno rekurzivnih funkcija

Funkcija arnosti k čiji je indeks n označava se sa f_n^k.

- kodiranje: ako su $gb(g) = a$, $gb(h_1) = b_1$, …, $gb(h_m) = b_m$ već određeni, kod kompozicije je

$$gb(g(h_1, \dots, h_m)) = gb(a, gb(b_1, \dots, b_m), 0)$$

- dekodiranje: $duzina(n) = 3$, tj. n kodira niz brojeva $((n)_0, (n)_1, (n)_2)$ i ako je:
  - kod funkcija argumenata $duzina((n)_1) = 0$, reč je o funkciji $f_{(n)_0}^k$ i
  - kod funkcija argumenata $duzina((n)_1) > 0$, reč je o funkciji dobijenoj kompozicijom $f_{(n)_0}^{duzina((n)_1)}(f_{((n)_1)_0}^k, \dots, f_{((n)_1)_{\dot{-}(1, duzina((n)_1))}}^k)$,

Nabrajanje parcijalno rekurzivnih funkcija

Funkcija arnosti k čiji je indeks n označava se sa f_n^k.

- kodiranje: ako su $gb(g) = a$ i $gb(h) = b$ već određeni, onda je $gb(a, b, 0, 0)$ kod fje dobijene primitivnom rekurzijom nad h sa osnovom g
- dekodiranje: $duzina(n) = 4$, tj. n kodira niz brojeva $((n)_0, (n)_1, (n)_2, (n)_3)$ i ako je
  - broj argumenata funkcije $k = 1$, tada je reč o funkciji $f_n^1(0) = (n)_0$, $f_n^1(x_1 + 1) = f^2_{(n)_1}(f_n^1(x_1), x_1)$ dobijenoj primitivnom rekurzijom i
  - broj argumenata funkcije $k > 1$, tada je reč o funkciji $f_n^k(0, x_2, \ldots, x_k)$ $= f^{k-1}_{(n)_0}(x_2, \ldots, x_k)$, $f_n^k(x_1 + 1, x_2, \ldots, x_k) = f^{k+1}_{(n)_1}(f_n^k(x_1, x_2, \ldots, x_k), x_1,$ $x_2, \ldots, x_k)$ dobijenoj primitivnom rekurzijom

Nabrajanje parcijalno rekurzivnih funkcija

Funkcija arnosti k čiji je indeks n označava se sa f_n^k.

- ako je $gb(g) = a$ već određen, onda je $gb(a, 0, 0, 0, 0)$ kod funkcije dobijene neograničenom minimizacijom $(\mu y)(g(x_1, \dots, x_k, y) = 0)$
- dekodiranje: $duzina(n) \geq 5$, tj. n kodira niz brojeva $((n)_0, (n)_1, (n)_2, (n)_3, (n)_4, \dots)$ onda je $f_n^k(x_1, x_2, \dots, x_k) = (\mu x_{k+1})(f_{(n)_0}^{k+1}(x_1, x_2, \dots, x_{k+1}) = 0)$

Nabrajanje parcijalno rekurzivnih funkcija

Zaključak: parcijalno rekurzivnih funkcija ima prebrojivo mnogo i postoji postupak njihovog nabrajanja koji je primitivno rekurzivan

Univerzalni predikat i univerzalna funkcija

- $UP(n, gb(a_1, \dots, a_k), b, y)$, univerzalni predikat
- f_n^k je definisana za argumente $a_1, \dots, a_k$
- $f_n^k(a_1, \dots, a_k) = b$
- najveći broj koji se koristi u tom izračunavanju manji do jednak od y
- UP proverava za proizvoljnu funkciju f_n i fiksirane argumente $a_1, \dots,$ a_k da li je $f_n(a_1, \dots, a_k) \downarrow b$

Theorem

Postoji primitivno rekurzivni predikat UP arnosti 4 tako da je za sve n i k, $f_n^k(a_1, \dots, a_k) = b$ ako i samo ako $(\exists y) UP(n, gb(a_1, \dots, a_k), b, y)$.

Univerzalni predikat i univerzalna funkcija

- Predikat UP ćemo definisati po slučaju koristeći pri tome opisani postupak gedelizacije
- Pošto to kodiranje garantuje da je svaki prirodan broj kod neke funkcije, predikat koga razmatramo će biti uvek definisan
- uslovi će biti u skladu sa zahtevima za definiciju po slučaju, pa će predikat koga razmatramo će biti primitivno rekurzivan

Definicija važenja $UP(n, a, b, y)$

Slučaj nula funkcije

- kodiranje: $gb(Z) = 0$
- dekodiranje: $duzina(n) = 0$, tj. n kodira prazni niz brojeva, reč je o nula-funkciji Z,
- $C_{UP}(n, a, b, y) = 1$ ako $duzina(n) = 0$ i $b = 0$

Ako je $duzina(n) = 0$ radi se o funkciji Z pa predikat UP važi za $b = 0$, bez obzira na vrednosti ostalih argumenata.

Definicija važenja $UP(n, a, b, y)$

Slučaj funkcije naslednika

- kodiranje: $gb(S) = gb(1)$,
- dekodiranje: $duzina(n) = 1$, tj. n kodira jednočlani niz brojeva $(n)_0$, reč je o funkciji naslednika S
- $C_{UP}(n, a, b, y) = 1$ ako $duzina(n) = 1$ i $b = (a)_0 + 1$

Ako je $duzina(n) = 1$ radi se o funkciji S pa UP važi za $b = (a)_0 + 1$ bez obzira na vrednosti ostalih argumenata, pri čemu je $(a)_0$ vrednost prve koordinate u dekompoziciji broja a.

Definicija važenja $UP(n, a, b, y)$

Slučaj funkcije projekcije

- kodiranje: $gb(P^i_k) = gb(1, i)$
- dekodiranje: $duzina(n) = 2$, tj. n kodira niz brojeva $((n)_0, (n)_1)$ i ako je:
  - $(n)_0 = 0$, ili se projektuje po koordinati $(n)_1 = 0$, reč je o funkciji Z,
  - $(n)_0 \geq 1$ i projektuje se po koordinati $1 \leq (n)_1 \leq k$, reč je o $P_k^{(n)_1}$ i
  - $(n)_0 \geq 1$ i projektuje se po koordinati $(n)_1 > k$, reč je o funkciji P^k_k,
- $C_{UP}(n, a, b, y) = 1$ ako $duzina(n) = 2$ i
  - ako je $(n)_0 = 0$ ili $(n)_1 = 0$ i $b = 0$ (Z),
  - ako je $(n)_0 \geq 1$ i $1 \leq (n)_1 \leq duzina(a)$ i $b = (a)_{\dot{-}(1,(n)_1)}$, $(P_k^{(n)_1})$
  - ako je $(n)_0 \geq 1$ i $duzina(a) < (n)_1$ i $b = (a)_{\dot{-}(1,duzina(a))}$, (P^k_k)

Definicija važenja $UP(n, a, b, y)$

Slučaj funkcije projekcije

- $C_{UP}(n, a, b, y) = 1$ ako $duzina(n) = 2$ i
  - ako je $(n)_0 = 0$ ili $(n)_1 = 0$ i $b = 0$ (Z),
  - ako je $(n)_0 \geq 1$ i $1 \leq (n)_1 \leq duzina(a)$ i $b = (a)_{\dot{-}(1,(n)_1)}$, $(P_k^{(n)_1})$
  - ako je $(n)_0 \geq 1$ i $k < (n)_1$ i $b = (a)_{\dot{-}(1,duzina(a))}$, (P_k^k)

Ako je $duzina(n) = 2$ radi se o P_k^i koja se kodira sa $n = gb(1, i) = gb((n)_0, (n)_1)$ i mogući su sledeći slučajevi. Ako je $(n)_0 = 0$ ili se projektuje po koordinati $(n)_1 = 0$ reč je o degenerisanom slučaju koji predstavlja funkciju Z, pa je rezultat $b = 0$. Ako je $(n)_0 \geq 1$ i projektuje se po koordinati $1 \geq (n)_1 \geq duzina(a)$, rezultat je argument a_i za $i = \dot{-}(1, (n)_1)$. Konačno, ako je $(n)_0 \geq 1$ i za koordinatu projekcije važi $k < (n)_1$, rezultat je $b = (a)_{\dot{-}(1,duzina(a))}$, projektovanje po poslednjoj kooordinati.

Definicija važenja $UP(n, a, b, y)$

Slučaj kompozicije

- kodiranje: ako su $gb(g) = a$, $gb(h_1) = b_1, \ldots, gb(h_m) = b_m$ već određeni, kod kompozicije je

$$gb(g(h_1, \ldots, h_m)) = gb(a, gb(b_1, \ldots, b_m), 0)$$

- dekodiranje: $duzina(n) = 3$, tj. n kodira niz brojeva $((n)_0, (n)_1, (n)_2)$ i ako je:
  - kod funkcija argumenata $duzina((n)_1) = 0$, reč je o funkciji $f^k_{(n)_0}$ i
  - kod funkcija argumenata $duzina((n)_1) > 0$, reč je o funkciji dobijenoj kompozicijom $f^{duzina((n)_1)}_{(n)_0}(f^k_{((n)_1)_0}, \ldots, f^k_{((n)_1)_{\dot{-}(1, duzina((n)_1))}})$,
- $C_{UP}(n, a, b, y) = 1$ ako $duzina(n) = 3$ i
  - ako je $duzina((n)_1) = 0$ i važi $UP((n)_0, a, b, y)$,
  - ako je $duzina((n)_1) \geq 1$ i postoji $d \leq y$ koje kodira rezultate funkcija argumenata kompozicije za koje je $duzina(d) = duzina((n)_1)$ i važe predikati $UP(((n)_1)_i, a, (d)_i, y)$, $0 \leq i \leq \dot{-}(1, duzina((n)_1))$, i $UP((n)_0, d, b, y)$,

Definicija važenja $UP(n, a, b, y)$

Slučaj primitivne rekurzije

- kodiranje: ako su $gb(g) = a$ i $gb(h) = b$ već određeni, onda je $gb(a, b, 0, 0)$ kod fje dobijene primitivnom rekurzijom nad h sa osnovom g
- dekodiranje: $duzina(n) = 4$, tj. n kodira niz brojeva $((n)_0, (n)_1, (n)_2, (n)_3)$ i ako je
  - broj argumenata funkcije $k = 1$, tada je reč o funkciji $f_n^1(0) = (n)_0$, $f_n^1(x_1 + 1) = f_{(n)_1}^2(f_n^1(x_1), x_1)$ dobijenoj primitivnom rekurzijom i
  - broj argumenata funkcije $k > 1$, tada je reč o funkciji $f_n^k(0, x_2, \dots, x_k)$ $= f_{(n)_0}^{k-1}(x_2, \dots, x_k)$, $f_n^k(x_1 + 1, x_2, \dots, x_k) = f_{(n)_1}^{k+1}(f_n^k(x_1, x_2, \dots, x_k), x_1,$ $x_2, \dots, x_k)$ dobijenoj primitivnom rekurzijom

Definicija važenja $UP(n, a, b, y)$

Slučaj primitivne rekurzije

- $C_{UP}(n, a, b, y) = 1$
- ako $duzina(n) = 4$ i
  - ako je $duzina(a) = 1$ i $(a)_0 = 0$ i $b = (n)_0$ (slučaj funkcije oblika $f(0) = c$.)
  - ako je $duzina(a) = 1$ i $(a)_0 \geq 1$ i postoji e, $0 < e \leq y$ i važi $UP(n, \frac{a}{2}, e, y)$ i $UP((n)_1, gb(e, \dot{-}(1, (a)_0)), b, y)$(Slučaj funkcije oblika $f(x+1) = h(f(x), x)$. Izraz $\frac{a}{2}$ znači da se za 1 smanjuje argument.),
  - ako je $duzina(a) > 1$ i $(a)_0 = 0$ i važi predikat $UP((n)_0, gb((a)_1, \ldots, (a)_{\dot{-}(1, duzina(a))}), b, y)$(Slučaj funkcije oblika $f(0, \ldots) = g(\ldots))$
  - ako je $duzina(a) > 1$ i $(a)_0 \geq 1$ i postoji e, $0 < e \leq y$ tako da je $UP(n, \frac{a}{2}, e, y)$ i $UP((n)_1, gb(e, \dot{-}(1, (a)_0), (a)_1, \ldots, (a)_{\dot{-}(1, duzina(a))}), b, y)$(Slučaj funkcije oblika $f(x+1, \ldots) = h(f(x, \ldots), x, \ldots)$. Izraz $\frac{a}{2}$ znači da se za 1 smanjuje argument po kome se sprovodi indukcija),

Definicija važenja $UP(n, a, b, y)$

Slučaj neograničene minimizacije

- kodiranje: ako su $gb(g) = a$, onda je $gb(a, 0, 0, 0, 0)$ kod funkcije $(\mu y)(g(x_1, \ldots, x_k, y) = 0)$
- dekodiranje: $duzina(n) \geq 5$, tj. n kodira niz brojeva $((n)_0, (n)_1, (n)_2, (n)_3, (n)_4, \ldots)$ onda je $f^k_n(x_1, x_2, \ldots, x_k) = (\mu x_{k+1})(f^{k+1}_{(n)_0}(x_1, x_2, \ldots, x_{k+1}) = 0)$
- $C_{UP}(n, a, b, y) = 1$ ako $duzina(n) \geq 5$ i
    - važi $UP((n)_0, a \cdot p(duzina(a))^{1+b}, 0, y)$ i za svaki $i < b$ postoji e, $0 < e < y$ tako da važi $UP((n)_0, a \cdot p(duzina(a))^{1+i}, e, y)$

Ako je $duzina(n) \geq 5$, radi se o neograničenoj minimizaciji. Vrednost $(n)_0$ prve koordinate dekompozicije broja n je indeks funkcije po kojoj se radi minimizacija, $a \cdot p(duzina(a))^{1+b}$ je kod argumenata funkcije $f_{(n)_0}$ koja za te argumente ima vrednost 0, a y je ograničenje brojeva korištenih u izračunavanju. Slično je i u preostalim slučajevima.

Univerzalni predikat i univerzalna funkcija

- Na osnovu definicije lako se uočava da je predikat UP primitivno rekurzivan
- Preostaje da se pokaže da je $f_n^k(a_1, \ldots, a_k) = b$ ako i samo ako je $(\exists y) UP(n, gb(a_1, \ldots, a_k), b, y)$.
- Dokaz sa desna na levo je direktna posledica definicije predikata UP.

Univerzalni predikat i univerzalna funkcija

Dokaz sa leva na desno se sprovodi indukcijom po n i po $a = gb(a_1, \ldots, a_k)$.

- Za slučajeve $duzina(n) \leq 2$ radi se o osnovnim funkcijama i tvrđenje je istinito
- Neka za svako $l < n$ tvrđenje važi, kao i za n i sve $x < a$
- Neka je $f_n^k(a_1, \ldots, a_k) = b$ i $duzina(n) = 5$, odnosno radi se o funkciji dobijenoj neograničenom minimizacijom.
- Tada je $f_{(n)_0}^{k+1}(a_1, \ldots, a_k, b) = 0$ i pošto je $(n)_0 < n$ po indukcijskoj hipotezi postoji neko v tako da važi $UP((n)_0, gb(a_1, \ldots, a_k, b), 0, v)$.

Univerzalni predikat i univerzalna funkcija

- za svako $i < b$ je $f^{k+1}_{(n)_0}(a_1, \ldots, a_k, i)$ definisano i veće od nule
- Prema indukcijskoj hipotezi postoje $u_0, \ldots, u_{b-1}$ i $v_0, \ldots, v_{b-1}$ takvi da za svako $i \leq b-1$ važi $UP((n)_0, gb(a_1, \ldots, a_k, i), u_i, v_i)$, gde je $u_i > 0$
- Definišimo $y = \max\{u_0, \ldots, u_{b-1}, v_0, \ldots, v_{b-1}, b\} + 1$
- Zaključujemo da važi $UP(n, gb(a_1, \ldots, a_k), b, y)$
- Slično se analiziraju i ostali slučajevi.

Univerzalni predikat i univerzalna funkcija

Theorem (Klinijeva teorema o normalnoj formi)

Svaka parcijalno rekurzivna funkcija se može definisati korištenjem najviše jednog operatora neograničene minimizacije.

- $f_U(n, a) = ((\mu z)(UP(n, a, (z)_0, (z)_1)))_0$
- argumenti su redom kod funkcije n i kod niza argumenata $a = gb(a_1, \ldots, a_k)$
- definisana ako postoji $z = gb(b, y)$, kod dvočlanog niza sastavljenog od rezultata b i ograničenja y za maksimalno dozvoljeni broj koji se koristi u izračunavanju funkcije f_n

Univerzalni predikat i univerzalna funkcija

- funkcija je parcijalno rekurzivna pošto je UP primitivno rekurzivni predikat
- naziva se *univerzalna funkcija*
- na osnovu prethodne teoreme, za svaku parcijalno rekurzivnu funkciju f_n^k važi:
- $f_n(a_1, \ldots, a_k) \simeq ((\mu z)(UP(n, a, (z)_0, (z)_1)))_0$
- univerzalna funkcija je parcijalno rekurzivna, ali nije totalna.
- analogija između ove funkcije i univerzalne Tjuringove mašine UTM: funkcija izračunava sve ostale funkcije kao što UTM simulira izvršavanje svih ostalih Tjuringovih mašina.

s-*m*-*n*-teorema

- $f(x, y)$
- fiskiramo vrednost prvog argumenta funkcije $f(x, y)$ $(x = a)$.
- dobija se unarnu funkciju $f(a, y)$ čiji jedini argument je y
- nova funkcija je parcijalno rekurzivna, čak se može izračunati i njen indeks
- ovo odgovara prenosu argumenata u potprogram.

Theorem (*s*-*m*-*n*-teorema)

Neka su prirodni brojevi $m, n \geq 1$. Tada postoji $m+1$-arna rekurzivna funkcija S^m_n takva da za proizvoljnu $m+n$-arnu parcijalno rekurzivnu funkciju f_e važi

$$f_e(x_1, \ldots, x_m, y_1, \ldots, y_n) \simeq f_{S^m_n(e, x_1, \ldots, x_m)}(y_1, \ldots, y_n).$$

- f je nastala iz funkcije f_e zamenom prvih m argumenata konstantama $a_1, \ldots, a_m$.
- f je nastala kompozicijom funkcije f_e, konstantnih funkcija i funkcija projekcije tako da je

$$\begin{aligned} f(y_1, \ldots, y_n) \simeq f_e \quad (\quad & K_n^{a_1}(y_1, \ldots, y_n), \ldots, K_n^{a_m}(y_1, \ldots, y_n), \\ & P_n^1(y_1, \ldots, y_n), \ldots, P_n^n(y_1, \ldots, y_n)). \end{aligned}$$

s-m-n-teorema

- kodiranje kompozicije: ako su $gb(g) = a$, $gb(h_1) = b_1$, ..., $gb(h_m) = b_m$ već određeni, kod kompozicije je

$$gb(g(h_1, \ldots, h_m)) = gb(a, gb(b_1, \ldots, b_m), 0)$$

- za fiksirane e, i a_1, ..., a_m, Gedelov broj funkcije f je $S^m_n(e, a_1, \ldots, a_m) =$ $gb(e, gb(gb(K^{a_1}_n), \ldots, gb(K^{a_m}_n), gb(P^1_n), \ldots, gb(P^n_n)), 0)$
- S^m_n koja izračunava taj broj je primitivno rekurzivna, pa i rekurzivna, jer je definisana kao kompozicija primitivno rekurzivnih funkcija

Fiksna tačka funkcije

Theorem (O fiksnoj tački; Teorema o rekurziji)

Ako je h unarna rekurzivna funkcija, postoji prirodan broj n tako da je $f_n \simeq f_{h(n)}$, gde su f_n i $f_{h(n)}$ parcijalno rekurzivne funkcije sa indeksima n odnosno $h(n)$.

- parcijalno rekurzivna funkcija $g(x, y) \simeq f_{h(f_x(x))}(y)$
- po *s*-*m*-*n*-teoremi postoji rekurzivna funkcija $S(x)$ koja daje indeks tako da je
- $g(x, y) \simeq f_{S(x)}\ (y)$
- $f_{h(f_x(x))}(y) \simeq f_{S(x)}(y)$
- funkcija S ima svoj indeks m
- za $x = m$ i $S(m) = n$ dobijamo
- $f_{h(f_m(m))}(y) \simeq f_{S(m)}(y)$, odnosno
- $f_{h(n)}(y) \simeq f_n(y)$.

Fiksna tačka funkcije

Theorem

Svaka unarna rekurzivna funkcija ima beskonačno mnogo fiksnih tačaka.

- za svaku unarnu rekurzivnu funkciju h i za svaki prirodan broj m postoji prirodan broj $n > m$ tako da je $f_n \simeq f_{h(n)}$.
- $M = \{f_1, \ldots, f_m\}$ prvih m parcijalno rekurzivnih funkcija
- pošto postoji beskonačno mnogo međusobno različitih parcijalno rekurzivnih funkcija, postoji i parcijalno rekurzivna funkcija $f_c \notin M$
- definišimo funkciju

$$g(x) = \begin{cases} c & \text{za } x \leq m \\ h(x) & \text{za } x > m \end{cases}$$

- g je rekurzivna i ima bar jednu fiksnu tačku n_0 koja je veća od m jer za $n \leq m$ je $f_{g(n)} = f_c \not\simeq f_n$.
- prema definiciji funkcije g, n_0 je istovremeno i fiksna tačka funkcije h

Fiksna tačka funkcije

Theorem

Ako je $f(x,y)$ parcijalno rekurzivna funkcija, onda postoji prirodan broj n takav da je $f(n,y) \simeq f_n(y)$.

- n fiksna tačka funkcije $S_1^1(x)$ iz s-m-n-teoreme
- $f_{S_1^1(n)} \simeq f_n$
- $f(x,y) \simeq f_{S_1^1(x)}(y)$
- $f(n,y) \simeq f_{S_1^1(n)}(y) \simeq f_n(y)$.

Fiksna tačka funkcije

Example

- posmatramo $f(y,x) = x^y$
- primenimo teoremu: Ako je $f(x,y)$ parcijalno rekurzivna funkcija, onda postoji prirodan broj n takav da je $f(n,y) \simeq f_n(y)$.
- postoji broj $n \in \mathbb{N}$ takav da je $f(n,x) \simeq f_n(x)$
- postoji broj $n \in \mathbb{N}$ takav da je $f_n(x) \simeq x^n$

Fiksna tačka funkcije

Example

- postoji broj $n \in \mathbb{N}$ takav da je $Dom(f_n) = \{n\}$
- posatramo funkciju

$$g(x,y) = \begin{cases} 0 & \text{za } x = y \\ \uparrow & \text{za } x \neq y \end{cases}$$

- postoji $n \in \mathbb{N}$ tako da je $f(n,y) \simeq f_n(y)$
- pri tome je $Dom(f_n) = \{n\}$.

Fiksna tačka funkcije

Definition

Funkcija f je *samoreprodukujuća* ako važi $f(x) \downarrow gb(f)$

Theorem

Postoji samoreprodukujuća funkcija.

- posmatramo $f(y, x) = y$
- postoji prirodan broj n takav da je $f(n, x) = f_n(x) = n$

samoreprodukujući program = programski virus koji se razmnožava šireći svoje kopije tokom izvršavanja

Tjuring-izračunljive i parcijalno rekurzivne funkcije:

- međusobno različiti formalni modeli izračunavanja
- međusobno su ekvivalentni
- određuju jednu te istu klasu izračunljivih funkcija

Theorem

Svaka parcijalno rekurzivna funkcija je Tjuring-izračunljiva.

- za sve osnovne funkcije (Z, S, P_k^j) postoje odgovarajuće Tjuringove mašine
- Tjuringovim mašinama je moguće simulirati operacije kompozicije, primitivne rekurzije i neograničene minimizacije

TM-simulacija kompozicije

- $h_1, \ldots h_m$ parc. rek. funkcije arnosti k
- g parc. rek. funkcija arnosti m
- za njih postoje odgovarajuće TM koje ih izračunavaju: $M_{h_1}, \ldots,$ M_{h_m}, M_g,
- treba opisati TM M_{komp} koja odgovara $g(h_1(x_1, \ldots, x_k), \ldots, h_m(x_1, \ldots, x_k))$
- na početku rada M_{komp} na traci se nalaze $x_1, \ldots x_k$

TM-simulacija kompozicije

- u prvoj fazi rada M_{komp} pravi kopiju ulaznih podataka i na njihovo mesto smešta rezultat rada M_{h_1}
- postupak se ponavlja $m-1$ puta sa $M_{h_2}, \ldots, M_{h_m}$
- traka sadrži m unarnih zapisa rezultata $M_{h_1}, \ldots, M_{h_m}$
- primenjuje se M_g, nakon čega M_{komp} završava rad
- ako za neko i, $h_i(x_1, \ldots, x_k) \uparrow$, biće i $M_{h_i} \uparrow$, pa i $M_{komp} \uparrow$

TM-simulacija primitivne rekurzije

- g parc. rek. funkcija arnosti m, $f(0, \ldots) = g(\ldots)$
- h parc. rek. funkcije arnosti $m+2$, $f(n+1, \ldots) = h(f(n, \ldots), n, \ldots)$
- za njih postoje odgovarajuće TM koje ih izračunavaju: M_g, M_h
- treba opisati TM M_{pr} koja odgovara funkciji f dobijenoj primitivnom rekurzijom
- na početku rada M_{pr} na traci se nalaze n i $x_1, \ldots x_k$

TM-simulacija primitivne rekurzije

- u petlji M_{pr} proverava da li je n po kome se radi primitivna rekurzija jednak 0
- ako $n \neq 0$, pomera udesno sadržaj trake i na oslobođeno mesto smešta argumente rekurzivnog poziva
- nakon završetka petlje, kada je argument po kome se radi rekurzija 0, izgled trake je:

$$101^{x_1+1}0\ldots01^{x_k+1}01101^{x_1+1}0\ldots01^{x_k+1}0\ldots01^{n+1}01^{x_1+1}0\ldots01^{x_k+1}$$

- stek na kome se nalaze argumenti za pozivanje funkcija
- redom rade M_g i odgovarajući broj puta M_h, nakon M_{pr} staje

TM-simulacija neograničene minimizacije

- f parc. rek. funkcija arnosti $k+1$,
- za nju postoji odgovarajuće TM koja je izračunava: M_f
- treba opisati TM M_μ koja odgovara funkciji $(\mu y)(f(x_1, \ldots, x_k, y) = 0)$
- na početku rada M_μ na traci se nalaze x_1, ..., x_k, 0

TM-simulacija neograničene minimizacije

- sadržaj se kopira na desno od krajnje desne ćelije koja sadrži znak 1
- nad tom kopijom se izvrši M_f
- ako je dobijeni rezultat 0, brišu se svi zapisi brojeva $x_1, \ldots, x_k$, rezultat je zapis argumenta po kome se vrši minimizacija
- ako dobijeni rezultat nije 0, briše se, uvećava se vrednost argumenta po kome se vrši minimizacija i postupak ponavlja

Theorem

Svaka Tjuring-izračunljiva funkcija je parcijalno rekurzivna.

- f funkcija koju izračunava Tjuringova mašina M_f
- pretpostavka: f unarna funkcija, a ako nije ulaz je Gedelov broj niza ulaznih podataka koji najpre raspakuje
- kodiranje programa i konfiguracija M_f
- opisivanje parc.rek. funkcija koje manipulišući tim kodovima simuliraju rad M_f

Parc. rek.-simulacija TM

- svim stanjima i operacijama pridružujemo kod
- kod stanja q_z je 0,
- kod stanja q_i je $i + 1$
- kodovi operacija 0, 1, L i R su 0, 1, 2 i 3
- program za M_f opisujemo sa $h(x) = a$, gde je x kod para (q_i, s), a a kod para (o, q_j)
- program je konačni niz naredbi, funkcija h se definiše po slučajevima, pa je primitivno rekurzivna

Parc. rek.-simulacija TM

Posmatramo funkcije:

- $$s(x,t) = \begin{cases} 1 & \text{nakon } t \text{ koraka rada glava } M_f(x) \\ & \text{je nad ćelije u kojoj je 1} \\ 0 & \text{inače} \end{cases}$$
- $l(x,t)$ - Gedelov broj niza znaka koji se nakon t koraka rada $M_f(x)$ nalaze levo od aktuelne ćelije zaključno sa najlevljom ćelijom u kojoj se nalazi znak 1
- $d(x,t)$ - Gedelov broj niza znaka koji se nakon t koraka rada $M_f(x)$ nalaze desno od aktuelne ćelije zaključno sa krajnjom desnom ćelijom u kojoj se nalazi znak 1
- $q(x,t)$ - broj tekućeg stanja nakon t koraka rada $M_f(x)$

Parc. rek.-simulacija TM

- $l(x,t)$, $s(x,t)$, $d(x,t)$ i $q(x,t)$ jednoznačno opisuju konfiguracija mašine $M_f(x)$ nakon t koraka rada

$$\begin{array}{ccc} l(x,t) & s(x,t) & d(x,t) \\ & q(x,t) & \end{array}$$

Parc. rek.-simulacija TM

- $l(x,t)$, $s(x,t)$, $d(x,t)$ i $q(x,t)$ se definišu induktivno
- Za $t = 0$, što odgovara početku izvršavanja mašine, ako su na traku smeštene $x+1$ jedinica koje čine unarni zapis ulaznog podatka x, biće:
    - $q(x,0) = 1$ - početno stanje
    - $l(x,0) = 0$ - levo od glave su sve nule
    - $s(x,0) = 1$ - glava je iznad najlevlje jedinice
    - $d(x,0) = gb(1^x)$ - desno od glave je x jedinica

Parc. rek.-simulacija TM

- za $t > 0$, $h(gb(q(x,t-1), s(x,t-1))) = a$ i $a = gb(o, q(x,t))$ kod para koji čine operacija i naredno stanje
    - $q(x,t) = (a)_1$
    - zavisno od $o = (a)_0$, određuju se $l(x,t)$, $s(x,t)$ i $d(x,t)$
    - ako je o kod upisa jedinice, $l(x,t) = l(x,t-1)$, $s(x,t) = 1$ i $d(x,t) = d(x,t-1)$.
    - Slično se postupa i u ostalim slučajevima
- pošto je h primitivno rekurzivna, a funkcije l, s, d i q se na osnovu nje definišu po slučajevima i one su primitivno rekurzivne

Parc. rek.-simulacija TM

M_f se zaustavlja nakon t koraka u standardnoj konfiguraciji ako i samo ako je ispunjeno:

- $(\forall i < t)(q(x, i) \neq 0)$ - prethodno nije bila u završnom stanju
- $q(x, t) = 0$ sada je u završnom stanju
- $l(x, t) = 0$ - levo nema jedinica
- $s(x, t) = 1$ - glava je nad jedinicom i
- $(\forall i < d(x, 0) + t)(\neg \div (p(i)^2, d(x, t)) \to \neg \div (p(i+1)^2, d(x, t)))$

Parc. rek.-simulacija TM

Poslednji deo uslova:

- $(\forall i < d(x,0)+t)(\neg \div (p(i)^2, d(x,t)) \to \neg \div (p(i+1)^2, d(x,t)))$
- nakon t koraka rada broj korištenih ćelija je manji do jednak od $d(x,0)+t$,na traci postoji tačno jedan neprekidni blok 1, kada se pojavi prva 0, desno su samo 0
- prisustvo 1 u ćeliji i - prisustvo $p(i)^2$ u proizvodu, tj., delivošću koda sa $p(i)^2$
- ako se u standardnoj konfiguraciji pojavi ćelija i koja sadrži 0, tj. $p(i)^2$ ne deli $d(x,t)$, isto će važiti i za sve ostale ćelije $i+1$, $i+2$, ...

Parc. rek.-simulacija TM

Ako se mašina M_f zaustavi u standardnoj konfiguraciji, tj. važi SK, rezultat y je predstavljen sa:

- 1^{y+1}
- odnosno blokom neprekidnih jedinica dužine $duzina(d(x,t))$
- $f(x) = duzina(d(x,(\mu t)SK))$

Univerzalna Tjuringova mašina *UTM*

UTM primer programibilnog digitalnog računara opšte namene sa programom i podacima smeštenim u memoriju, simulira izvršavanje ostalih TM, ulazni podaci za *UTM* su opis - program neke posebne TM i ulazni podaci te TM, rezultat rada *UTM* je rezultat rada simulirane posebne TM.

Theorem

Postoji UTM.

- univerzalna parcijalno rekurzivna funkcija: $f_U(n,a) = ((\mu z)(UP(n,a,(z)_0,(z)_1)))_0$
- *UTM* - Tjuringova mašina koja interpretira f_U - je univerzalna TM jer:
  - *UTM* postoji prema teoremi o simulaciji parc. rek. f-ja
  - neku TM M simuliramo nekom parc. rek. f-jom, prema teoremi o simulaciji TM
  - tu f-ju simuliramo sa f_U, a nju sa *UTM*

Čerč (1936):

svaki algoritam definiše funkciju koja pripada jednoj dobro definisanoj klasi funkcija

(klasa parcijalno rekurzivnih funkcija, klasa Tjuring-izračunljivih funkcija, klasa λ-definabilnih funkcija ili neka druga ekvivalentna klasa), tj.klasa intuitivno izračunljivih funkcija se poklapa sa svakom od tih klasa

Čerčova teza

- intuitivni pojam algoritma je zasnovan na iskustvenom znanju o ljudskim umnim sposobnostima
- klase izračunljivih funkcija precizno definisane odgovarajućim formalnim modelima izračunavanja
- Čerčova teza izjednačava neformalni i formalni pristup pojmu efektivne izračunljivosti
- u strogom smislu nije matematičko tvrđenje (sličnija je formulacijama raznih fizičkih zakona)
- ne može se dokazati u okviru neke formalne teorije
- može biti opovrgnuta ako bi bila pronađena funkcija koja jeste intuitivno izračunljiva, a nije Tjuring-izračunljiva

Čerčova teza

- Argumenti u prilog tezi:
  - nema kontraprimera
  - međusobna ekvivalentnost raznorodnih formalnih modela izračunavanja do koje teško da bi došlo da neka od intuitivnih karakteristika algoritama nije njima obuhvaćena
- Čerčova teza se može prihvatiti i kao definicija izračunljivosti

Primene Čerčove teza

- da bi se u raznim dokazima istakle suštinske ideje i izbegli tehnički detalji često se pribegava formulaciji oblika: 'funkcija je intuitivno izračunljiva, pa je prema Čerčovoj tezi parcijalno izračunljiva'
- argument pri objašnjavanju zašto neki problem nije rešiv: postupak za rešavanje problema ne nalazi u nekoj od formalizovanih klasa izračunljivih funkcija, na osnovu Čerčove teze ne postoji efektivni postupak za rešavanje tog problema

Čerčova teza i praktična izračunljivost

- izračunljivost u principu: parc. rek. funkcije
- praktična izračunljivost: ono što se stvarno može izračunati
- postoje parc. rek. funkcije za čije izračunavanje je potrebno vreme duže od vremena proteklog od nastanka kosmosa, i/ili se zahteva veći broj memorijskih registara nego što je broj atoma na Zemlji
- da li su takve parc. rek. funkcije zaista izračunljive
- Čerčova teza predstavlja korisnu granicu klase funkcija izvan koje sigurno nema praktično izračunljivih funkcija

Relativna izračunljivost

- klasa funkcija koje nisu parcijalno rekurzivne (neprebrojiva) je daleko veća od klase parcijalno rekurzivnih funkcije (prebrojiva)
- relativna izračunljivost predstavlja odnos između dvaju funkcija f i g
- f možemo izračunati pomoću osnovnih funkcija i g primenom kompozicije, primitivne rekurzije i minimizacije
- ako g jeste parc. rek. to isto važi i za f
- to prestaje da važi ako g nije parcijalno rekurzivna

Relativna izračunljivost

- g proizvoljna totalna funkcija
- funkcije izračunljive pomoću *orakla* za funkciju g su funkcije koje se relativno izračunavaju u odnosu na g, odnosno formiraju klasu koja kao osnovnu funkciju ima i funkciju g
- ako je f izračunljiva pomoću g, onda je f lakša za izračunavanje od g, jer znajući da izračunamo g to umemo da uradimo i za f
- opisivanje klase funkcija koju dobijamo uključujući i g u osnovne funkcije se sprovodi analogno postupku za opis parcijalno rekurzivnih funkcija
- $f \leq g$ - f je izračunljiva pomoću orakla za g

Relativna izračunljivost

Example

- $g(x) = 1$ ukoliko $f_x(x) \downarrow$, a inače $g(x) = 0$
- nije parc. rek. f-ja (tj. nije rekurzivna, pošto je totalna)
- C_g - klasa funkcija koja pored parcijalno rekurzivnih sadrži i funkcije koje su relativno izračunljive u odnosu na g
- C_g sadrži i funkcije koje nisu parcijalno rekurzivne, jer je $g \leq g$ (C_g je prebrojivo beskonačna)
- $f_1, f_2, \ldots$ jedno nabrajanje funkcija klase C_g
- funkcija h definisana sa

$$h(x) = \begin{cases} 1 & \text{ako } f_x(x) \downarrow \\ 0 & \text{inače} \end{cases}$$

ne pripada klasi C_g.

Tjuringovi stepeni

- klasa funkcija C sa osobinom da za svake dve funkcije $f, g \in C$ važi da je $f \leq g$ i $g \leq f$ naziva se *Tjuringov stepen*.
- Tjuringovi stepeni čine jednu hijerarhiju klasa funkcija koja se intenzivno proučava.

Odlučivost

- razlog uvođenja formalnih modela izračunavanja: utvrđivanje da li za neki problem postoji algoritam koji ga rešava
- Čerčova teza i definicije Tjuring-izračunljivih i parcijalno rekurzivnih funkcija određuju jasnu granicu dosega algoritama
- problemi se obično formulišu sa 'da li je ...' ili 'da li važi ...'
- ako smo u stanju da na takvo pitanje uvek odgovorimo problem je rešiv (odlučiv)
- probleme možemo shvatiti kao predikate (skupove)

Odlučivost

- predikat je relacija nad $\mathbb{N}^k$ za $k = 1, 2, \ldots$
- karakteristična funkcija predikata za k-torku argumenata ima vrednost 1 ako za tu k-torku predikat važi, inače je vrednost funkcije 0

Definition

Predikat (skup) R je *rekurzivan (odlučiv)* ako je njegova karakteristična funkcija C_R rekurzivna (postoji TM koja je izračunava).

Definition

Rekurzivne funkcije su totalne parcijalno rekurzivne funkcije.

Primeri odlučivih skupova:

- skup $\mathbb{N}$
- svaki konačan podskup od $\mathbb{N}$
  - za konačan skup $\{a_1, \ldots, a_n\}$ prirodnih brojeva karakteristična funkcija $x = a_1 \vee \ldots \vee x = a_n$ je rekurzivna
- skup parnih brojeva
- skup neparnih brojeva
- klasa odlučivih skupova (predikata) je zatvorena za osnovne operacije komplementiranja (u odnosu na skup $\mathbb{N}^k$), $\cup$, $\cap$, $\setminus$ (iz razmatranja karakterističnih funkcija za skupove dobijene tim operacijama)

Neodlučivi predikati

Example

Problem utvrđivanja da li je $f_x(x)\downarrow$, tj. da li je unarna parcijalno rekurzivna funkcija sa indeksom x definisana za argument x nije odlučiv. Znači da predikat P definisan tako da $P(x)$ važi ako i samo ako je $f_x(x)\downarrow$ nije odlučiv.

Neodlučivi predikati: $Tot(x)$

Example

- C_{Tot} karakteristična funkcija unarnog predikata $Tot(x)$ koji važi ako i samo ako je f_x totalna funkcija
- Tot nije odlučiv - metoda dijagonalizacije
- h (rekurzivna, ako i C_{Tot}) definisana sa:
  - $h(0) = (\mu y)(C_{tot}(y) = 1)$
    $h(n+1) = (\mu y)(y > h(n) \wedge C_{Tot}(y) = 1)$
- h lista kodove svih totalnih funkcija
- f (rekurzivna, ako i h) definisana sa:
  - $f(x) = f_{h(x)}(x) + 1$
- e - indeks za f tako da za neko $n_0 \in \mathbb{N}$, $h(n_0) = e$
- $f(n_0) = f_{h(n_0)}(n_0) + 1 = f(n_0) + 1$, kontradikcija
- problem 'da li je funkcija totalna' nije odlučiv

Neodlučivi predikat: $Halt(x, y)$

Example

- C_{Halt} karakteristična funkcija predikata $Halt(x, y)$ koji važi ako i samo ako je $f_x(y) \downarrow$, tj. ako je funkcija f_x definisana za ulaz y
- *halting*-problem, *problem zaustavljanja*
- uopštenje problema da li $f_x(x) \downarrow$ koji nije odlučiv
- ako je $C_{Halt}(x, y)$ rekurzivna, to bi bila i $C_{Halt}(x, x)$, pa bi problem da li $f_x(x) \downarrow$ bio odlučiv
- pošto to nije, $Halt(x, y)$ nije odlučiv
- na osnovu veze parc. rek. funkcija i TM: u opštem slučaju ne možemo odgovoriti da li neki program za dati ulaz završava rad u konačnom broju koraka (negativan odgovor na Hilbertov Entscheidungsproblem)

Neodlučivi predikat: $\mathbb{O}(x)$

Example

- $C_{\mathbb{O}}$ karakteristična funkcija predikata $\mathbb{O}(x)$ koji važi ako i samo ako je f_x nula funkcija $f\mathbb{O}$ ($f\mathbb{O}(x) = 0$)
- pretpostavimo da $C_{\mathbb{O}}$ jeste rekurzivna funkcija
- posmatramo parc. rek. funkciju f definisanu sa:
  - $f(x, y) = f\mathbb{O}(f_x(y))$
- e - indeks za f
- s-m-n-teorema: $f(x, y) \simeq f_{S_1^1(e,x)}(y)$.
- $f_{S_1^1(e,x)}$ je nula funkcija, tj. $C_{\mathbb{O}}(S_1^1(e, x)) = 1$ ako i samo ako je $C_{Halt}(x, y) = 1$
- iz odlučivosti predikata $\mathbb{O}$ sledi odlučivost predikata *Halt*, pa problem 'da li je funkcija nula-funkcija', nije odlučiv

Neodlučivi predikat: jednakost dve funkcije

Example

- problem jednakosti dve parcijalno rekurzivne funkcije, odnosno da li je $f_x \simeq f_y$, nije odlučiv
- direktna posledica prethodnog primera
- pošto je nula-funkcija $f\mathbb{O}$ rekurzivna, a time i parc. rek., ona ima svoj indeks e
- ako bi problem $f_x \simeq f_y$ bio odlučiv, to bi bio i problem $f_x \simeq f_e$ ($f_x \simeq f\mathbb{O}$), što nije slučaj
- na osnovu veze parc. rek. funkcija i TM: u opštem slučaju ne možemo odgovoriti da li su neka dva programa jednaka

Neodlučivi predikati- Rajsova teorema

Theorem (Rajsova teorema)

Neka je $\mathbb{B}$ neprazna prava potklasa klase svih parcijalno rekurzivnih funkcija. Problem da li je $f_x \in \mathbb{B}$ nije odlučiv.

- h: $h(x) \uparrow$, za svako $x \in \mathbb{N}$
- neka: $h \notin \mathbb{B}$
- ako jeste, posmatramo $\mathbb{C}\mathbb{B}$, odlučiv akko odlučiv $\mathbb{B}$)
- $g \in \mathbb{B}$ funkcija različita od h
- uvek postoji: $\mathbb{B} \neq \emptyset$, $h \notin \mathbb{B}$
- ako je klasa $\mathbb{B}$ odlučiva, g pronalazimo proveravajući redom da li se neka od funkcija iz bilo kog nabrajanja svih parcijalno rekurzivnih funkcija nalazi u $\mathbb{B}$

Neodlučivi predikati - Rajsova teorema

- $K_1^1(x)$, za $x \in \mathbb{N}$, $K_1^1(x) = 1$
- totalna, konstanta funkcija
- parc. rek. f-ja $f(x, y)$ definisanu sa $g(y) \cdot K_1^1(f_x(x))$
- ima indeks e
- s-m-n-teorema: $f(x, y) \simeq f_{S_1^1(x)}(y)$
- ako $x \in Dom(f_x)$, biće $f_{S_1^1(x)} \simeq g \in \mathbb{B}$
- ako $x \notin Dom(f_x)$ biće $f_{S_1^1(x)} \simeq h \notin \mathbb{B}$
- $f_{S_1^1(x)} \in \mathbb{B}$ akko $x \in Dom(f_x)$
- pripadanja funkcije klasi $\mathbb{B}$ nije odlučivo

Neodlučivi predikati- posledice Rajsove teoreme

Rajsova teorema: svi netrivijalni skupovi parcijalno rekurzivnih funkcija su neodlučivi.

Theorem

Sledeći problemi nisu odlučivi:

- *domen funkcije je konačan,*
- *domen funkcije je beskonačan,*
- *kodomen funkcije je konačan i*
- *kodomen funkcije je beskonačan.*

Neodlučivi problemi

- problem reči za grupe: grupa G sa jediničnim elementom e generisana je skupom elemenata $Gen_G = \{g_1, g_2, \ldots\}$, da li za proizvoljan izraz t_1 sastavljan od elemenata iz Gen_G (recimo $t_1 = g_2^3 g_1^{-1} g_5$) važi $t_1 = e$,
- rešivost diofantskih jednačina, da li polinom $p(x_1, \ldots, x_m) = 0$ ima rešenja u $\mathbb{N}$
- problemi zadovoljivosti i valjanosti formula predikatskog računa prvog reda,
- problem pokrivanja ravni: dat je konačan broj proizvoljnih oblika poligona, pitanje je da li je moguće u potpunosti, bez preklapanja, pokriti ravan poligonima samo tih oblika

(Ne)odlučivi problemi

- odlučivo: zadovoljivost i valjanost iskaznih formula
- teorija T je odlučiva ako postoji rekurzivna karakteristična funkcija problema 'formula α je teorema teorije T'
- odlučive teorije: teorija Bulovih algebri, teorija množenja prirodnih brojeva, teorija Abelovih grupa, teorija realno zatvorenih polja, elementarna euklidska geometrija
- neodlučive teorije: Peanova aritmetika, teorija grupa, teorija prstena, teorija polja, ZF-teorija skupova itd.

Parcijalno odlučivi predikati

Definition

Predikat R je *parcijalno odlučiv (rekurzivno nabrojiv)* ako je njegova karakteristična funkcija parcijalno rekurzivna:

$$C_R(x_1, \ldots, x_n) = \begin{cases} 1 & \text{ako važi } R(x_1, \ldots, x_n) \\ \text{nedefinisano} & \text{inače.} \end{cases}$$

Ova definicija se može oslabiti tako što se dozvoli da za neke, ali ne nužno sve, $x_1, \ldots, x_n$ za koje $R(x_1, \ldots, x_n)$ ne važi, bude $C_R(x_1, \ldots, x_n) = 0$

Parcijalno odlučivi predikati

Example

- $p(x_1, \ldots, x_m, y_1, \ldots, y_n)$ je polinom sa celobrojnim koeficijentima, $x_1^3 - 2$, $x_1^4 \cdot x_2^5 + 6$,
- predikat P: $P(x_1, \ldots, x_m)$ akko $(\exists y_1) \ldots (\exists y_n) p(x_1, \ldots, x_m, y_1, \ldots, y_n) = 0$, je *diofantovski*
- Jurij Matijaševič: svaki p.o. predikat je ekvivalentan nekom diofantovskom
- problem rešivosti diofantovskih jednačina nije odlučiv

Parcijalno odlučivi predikati

Theorem

Skup P je parcijalno odlučiv ako i samo ako postoji parcijalno rekurzivna funkcija f čiji je domen skup P.

- ako je P p.o., funkcija f je njegova karakteristična funkcija C_P
- obrnuto, neka je f tražena funkcija
- $C_P(x_1, \dots, x_n) = K_1^1(f(x_1, \dots, x_n))$ je karakteristična funkcija za P
- C_P je parcijalno rekurzivna, pa je P p.o.

Parcijalno odlučivi predikati

Example

Na osnovu definicije - svaki odlučiv predikat je i parcijalno odlučiv pošto je svaka rekurzivna funkcija istovremeno i parcijalno rekurzivna, dok obrnuto ne mora važiti.

Example

Predikat *Halt* je neodlučiv. Kako je

$$C_{Halt}(x, y) = K_1^1(f_x(y))$$

parc. rek. funkcija, predikat *Halt* je p.o.

Parcijalno odlučivi predikati

Theorem

Predikat $P(x_1, \ldots, x_n)$ *je p.o. akko postoji odlučiv predikat* $R(x_1, \ldots, x_n, y)$ *tako da:*

$$P(x_1, \ldots, x_n) \text{ akko } (\exists y) R(x_1, \ldots, x_n, y).$$

- R traženi predikat, definišemo $C_P(x_1, \ldots, x_n) \simeq K_1^1((\mu y)(R(x_1, \ldots, x_n, y)))$
- C_P je parc.rek., pa je P p.o.
- obrnuto: P je p.o., indeks od C_P je e
- posmatramo: $R(x_1, \ldots, x_n, y) = UP(e, gb(x_1, \ldots, x_n), (y)_0, (y)_1)$ UP je univerzalni predikat, prim. rek., R je odlučiv
- $P(x_1, \ldots, x_n)$ važi akko $C_P(x_1, \ldots, x_n) \downarrow$ akko $(\exists y) R(x_1, \ldots, x_n, y)$

Parcijalno odlučivi predikati

Theorem

Predikat P je odlučiv ako i samo ako su predikati P i $\mathbb{C}P$ parcijalno odlučivi.

- P je p.o. ako postoji program koji odgovara potvrdno ako predikat važi za argumente, inače ne mora da se zaustavi
- neka su programi $Prog_P$ za P i $Prog_{\mathbb{C}P}$ za njegov komplement $\mathbb{C}P$
- $Prog_P$ i $Prog_{\mathbb{C}P}$ rade paralelno
- za svaki x je ili $P(x)$ ili $\mathbb{C}P(x)$ pa jedan od programa staje
- ako stane: $Prog_P$ - važi $P(x)$, $Prog_{\mathbb{C}P}$ - ne važi $P(x)$
- P je odlučiv

Parcijalno odlučivi predikati

Theorem

Predikat P je odlučiv ako i samo ako su predikati P i $\mathbb{C}P$ parcijalno odlučivi.

- P odlučiv - odlučiv je i $\mathbb{C}P$, pa i p.o.
- neka su P i $\mathbb{C}P$ p.o.
- postoje odlučivi R i S tako da:
  - $P(x_1, \dots, x_n)$ akko $(\exists y)R(x_1, \dots, x_n, y)$
  - $\mathbb{C}P(x_1, \dots, x_n)$ akko $(\exists y)S(x_1, \dots, x_n, y)$
- za svaku fiksiranu n-torku $x_1, \dots, x_n$ važi tačno jedan od $(\exists y)R(x_1, \dots, x_n, y)$, odnosno $(\exists y)S(x_1, \dots, x_n, y)$

Parcijalno odlučivi predikati

- $g(x_1, \ldots, x_n, y) = C_R(x_1, \ldots, x_n, y) + C_S(x_1, \ldots, x_n, y)$
  C_R, C_S rekurzivne karakteristične funkcije R i S
- g rekurzivna
- za svaku fiksiranu n-torku $x_1, \ldots, x_n$, jedna od funkcija C_R i C_S je nula-funkcija za svako y, dok ona druga za neke y ima takođe vrednost 0, ali i za neke y ima vrednost 1
- $Im(g) = \{0, 1\}$
- posmatramo: $f(x_1, \ldots, x_n) = (\mu y)(g(x_1, \ldots, x_n, y) = 1)$
- f je rekurzivna

Parcijalno odlučivi predikati

- za $f(x_1, \ldots, x_n) = (\mu y)(g(x_1, \ldots, x_n, y) = 1)$
- za fiksirane $x_1, \ldots, x_n$ vrednost f-a je minimalno y za koje je ispunjeno da važi bilo $R(x_1, \ldots, x_n, y)$, bilo $S(x_1, \ldots, x_n, y)$
- definišimo funkciju:

$$C_P(x_1, \ldots, x_n) = \begin{cases} 1 & \text{ako } C_R(x_1, \ldots, x_n, f(x_1, \ldots, x_n)) = 1 \\ 0 & \text{inače.} \end{cases}$$

- C_P je rekurzivna i ima vrednost 1 ako i samo ako $C_R(x_1, \ldots, x_n, f(x_1, \ldots, x_n)) = 1$ ako i samo ako važi $(\exists y)R(x_1, \ldots, x_n, y)$, odnosno ako i samo ako važi $P(x_1, \ldots, x_n)$, pa je to karakteristična funkcija predikata P
- P je odlučiv

Parcijalno odlučivi predikati

Example

- $P(x)$ akko $x \in Dom(f_x)$ nije odlučiv
- $C_P(x) = K_1^1(f_x(x))$ je parcijalno rekurzivna
- $P(x)$ je parcijalno odlučiv
- ako bi komplement $\complement P(x)$ akko $x \notin Dom(f_x)$ bio p.o., $x \in Dom(f_x)$ bi bio odlučiv
- $\complement P(x)$ nije parcijalno odlučiv

Parcijalno odlučivi predikati

Theorem

Neka je $A \subset \mathbb{N}$. Tada je ekvivalentno:

1. *A je parcijalno odlučiv,*
2. *$A = \emptyset$ ili A je kodomen neke rekurzivne funkcije i*
3. *postoji indeks e i broj n ($e, n \in \mathbb{N}$) takvi da je A kodomen n-arne parcijalno rekurzivne funkcije f_e.*

- $(2 \Rightarrow 3)$: trivijalno, svaka rek. f-ja je parc. rek.
- $(3 \Rightarrow 1)$:
  - $A = Im(f_e)$
  - $C_A(x) = K_1^1((\mu y) f_e((y)_0, \dots, (y)_{n-1}) = x))$ je parc. rek.
  - A parcijalno odlučiv

Parcijalno odlučivi predikati

($1 \Rightarrow 2$, A je p.o. $\Rightarrow A = Im(g)$):

- A je p.o. $\Rightarrow$ za neku parc. rek. f_e, $A = Dom(f_e)$
- ako $f_e(x) \uparrow$ za svaki x, $A = \emptyset$
- neka $A \neq \emptyset$, postoji $a \in A$
- $T(e, x, t)$ definisan kao UP, bez rezultata funkcije
- UP prim. rek. $\Rightarrow$ T je primitivno rekurzivan
- definišemo: $g(x, t) = x \cdot C_T(e, x, t) + a \cdot C_{\neg T}(e, x, t)$
- trivijalno: g je rekurzivna
- $x \in Im(g)$ akko $x \in A$, ako $x \notin A$ onda $a \in Im(g)$
- $A = Im(g)$

Parcijalno odlučivi predikati

Example

- $Tot = \{x : f_x \text{ je totalna funkcija}\}$ nije p.o.
- ako jeste:
- $A = Im(h)$, h rekurzivna f-ja
- niz funkcija $f_{h(n)}$ sadržao bi sve rekurzivne funkcije
- kontradikcije kao i u primeru o nerekur. predikata Tot $(f(x) = f_{h(x)}(x) + 1)$

- *Tot*: postoje problemi (skupovi) koji ne samo da nisu odlučivi već nisu ni parcijalno odlučivi
- postoji jedna hijerarhija složenosti skupova prirodnih brojeva - *aritmetička hijerarhija*
- klase odlučivih i p.o. skupova su na samom dnu ove hijerarhije - najjednostavnije klase
- skupovi koji pripadaju klasama iz aritmetičke hijerarhije - *aritmetički skupovi*
- postoje skupovi prirodnih brojeva koji nisu u ovoj hijerarhiji

Aritmetička hijerarhija

Definition

- $A \subset \mathbb{N}$ je u klasi Σ^0_1 akko postoji odlučivi binarni predikat B takav da $x \in A$ akko $(\exists y)B(x, y)$.
- $A \subset \mathbb{N}$ je u klasi Π^0_1 akko postoji Σ^0_1 skup A' takav da je $A = \complement A'$.
- $A \subset \mathbb{N}$ je u klasi Δ^0_1 akko pripada klasi $\Sigma^0_1 \cap \Pi^0_1$.
- $A \subset \mathbb{N}$ je u klasi Σ^0_{n+1} akko postoji Π^0_n binarni predikat B takav da $x \in A$ akko $(\exists y)B(x, y)$.
- $A \subset \mathbb{N}$ je u klasi Π^0_{n+1} akko postoji Σ^0_{n+1} skup A' takav da je $A = \complement A'$.
- $A \subset \mathbb{N}$ je u klasi Δ^0_{n+1} akko pripada klasi $\Sigma^0_{n+1} \cap \Pi^0_{n+1}$.

Aritmetička hijerarhija

- generalizacija za skupove veće arnosti: kodiranje k-torki
- klase Δ^0_1 i Σ^0_1 - klase odlučivih i p.o. skupova
- Skupovi Π^0_1 - *koparcijalno odlučivi*, *korekurzivno nabrojivi*
- pokazuje se:
  - hijerarhija je prava
  - $\Delta^0_n \subset \Sigma^0_n$, $\Delta^0_n \neq \Sigma^0_n$, $\Delta^0_n \subset \Pi^0_n$, $\Delta^0_n \neq \Pi^0_n$, $\Pi^0_n \neq \Sigma^0_n$, $\Sigma^0_n, \Pi^0_n \subset \Delta^0_{n+1}$
  - klase su zatvorene za konačne unije i preseke

Teorija izračunljivosti i programski jezici

- zapisivanje algoritamskih šema u vidu nizova naredbi je dovelo do uvođenja labela, tj. adresa naredbi i naredbi uslovnog i bezuslovnog skoka:

  $$\text{if } S \text{ then go to } A \quad \text{i} \quad \text{go to } A$$

  koje su tipične za nestruktuirane programske jezike
- naredbe Tjuringove mašine u sebi sadrže ideju programskog skoka u formi promene stanja
- jezgro PASCAL-a: formalizam *while*-programa obogaćen proizvoljnim logičkim izrazima
- formulacija izračunljivih funkcija na bazi λ-računa, proširena numeričkim i funkcijama za rad sa rečima predstavlja osnovu za programski jezik LISP
- Osnova PROLOG-a se može shvatiti kao deduktivno zasnovan mehanizam rešavanja sistema jednačina kojima se definišu funkcije.