Sveučilište u Zagrebu
PMF–Matematički odsjek

Mladen Vuković

Matematička logika i izračunljivost

predavanja i vježbe

Zagreb, srpanj, 2021.

Sadržaj

Predgovor

Ovaj nastavni materijal namijenjen je prije svega studentima diplomskih studija Fakulteta elektrotehnike i računarstva u Zagrebu. Kolegij *Matematička logika i izračunljivost* trenutno je izborni kolegij na prvoj godini raznih diplomskih studija s tjednom satnicom 3+0. Navedeni kolegij predajem od akademske godine 2009./10.

Za greške u ovom nastavnom materijalu kriv sam samo ja. Biti ću zahvalan svakome tko me upozori na grešku bilo kakve vrste.

Zagreb, srpanj 2021. Mladen Vuković

Poglavlje 1

Prvo predavanje – Uvod i logika sudova

1.1 Uvod

Predstavljanje:

MLADEN VUKOVIĆ

vukovic@math.hr

www.math.pmf.unizg.hr/hr/vukovic

Konzultacije: po dogovoru (e–mailom)

SADRŽAJ KOLEGIJA

1. Logika sudova
    1. ponavljanje o skupovima
    2. sintaksa i semantika
    3. normalne forme
    4. glavni test
    5. formalni sistem
    6. modalna propozicionalana logika
2. Logika prvog reda
    1. sintaksa i semantika
    2. preneksna normalna forma

3. glavni test
4. formalni sistem

3. Izračunljivost

1. RAM–stroj
2. parcijalno rekurzivne funkcije
3. kodiranje; indeksi; Kleenijev teorem
4. Churchova teza.

LITERATURA

1. M. Vuković, *Matematička logika*, Element, Zagreb, 2009.
2. M. Vuković, *Izračunljivost*, nastavni materijal, PMF–MO, Zagreb, 2009. https://www.math.pmf.unizg.hr/sites/default/files/pictures/izn-skripta-2009.pdf
3. http://www.fer.hr/predmet/mli
4. R. Cori, D. Lascar, *Mathematical Logic* I, II, Oxford University Press, 2000.
5. E. Mendelson, *Introduction to mathematical logic*, Chapman&Hall, 1997.
6. M. Sipser, *Introduction to the Theory of Computation*, PWS Publishing Company, 1996.

IZVEDBENI PLAN

- međuispit: 45 bodova
- završni ispit: 45 bodova
- Domaće zadaće: 10 bodova
- Prag za prolaz je 45 bodova

OSNOVNO O SKUPOVIMA – podsjećanje i ponavljanje

1. označavanje i zadavanje skupova
2. oznake: relacija "biti element", prazan skup, podskup, partitivni skup

3. jednakost skupova
4. operacije sa skupovima: unija, presjek, razlika, komplement
5. uređeni par, Kartezijev produkt, relacija, relacija ekvivalencije
6. funkcija; domena, slika, graf; injekcija, surjekcija, bijekcija
7. skupovi brojeva:
   a) skup prirodnih brojeva $\mathbb{N} = \{0,\ 1,\ 2,\ \ldots\}$
   b) skup cijelih brojeva $\mathbb{Z} = \{\ldots, -3,\ -2,\ -1,\ 0,\ 1,\ 2,\ 3,\ \ldots\}$
   c) skup racionalnih brojeva $\mathbb{Q} = \{p/q\ :\ p \in \mathbb{Z},\ q \in \mathbb{N} \setminus \{0\}\}$
   d) skup realnih brojeva $\mathbb{R}$
   e) skup kompleksnih brojeva $\mathbb{C} = \{a + ib : a, b \in \mathbb{R}\}$
8. konačni i beskonačni, te prebrojivi i neprebrojivi skupovi.

1.2 Logika sudova

1.2.1 Intuitivno o sudovima

Intuitivno, sud je svaka suvisla izjavna rečenica koja je istinita ili lažna, ali ne oboje. No, to svakako ne može biti definicija suda, jer tada se postavlja pitanje npr. što je rečenica, ili pak što je istinita rečenica. Pokušat ćemo objasniti pojam suda pomoću nekoliko primjera.

a) Rečenica *"Dva plus dva je jednako četiri."* jeste sud i to istinit.

b) Rečenica *"Dva plus dva je jednako pet."* jeste sud i to lažan.

c) Rečnica *"x plus dva je jednako osam."* nije sud, jer za ovu rečenicu ne možemo reći je li istinita ili lažna, dok nismo rekli koliko je x.

d) Rečenica *"Ja sada lažem."* nije sud, jer pretpostavimo li da je istinita, onda sam zaista lagao, pa je ono što sam rekao lažno. Obrnuto, pretpostavimo li da je ta rečenica lažna onda nisam lagao, pa je ono što sam rekao istina. Dakle, za ovu rečenicu ne možemo reći ni da je istinita, a ni da je lažna.

e) Rečenica *"Koliko je sati?"* nije sud, jer nije izjavna rečenica.

Sudovi a) i b) su jednostavnog oblika. Pomoću veznika *i, ili, ako ... onda* i *nije* možemo iz jednostavnijih sudova graditi složene. Primjerice, rečenica *"Ako Vanja uči, onda Ivona gleda crtane filmove."* je primjer složenog suda, jer je nastala pomoću veznika *ako ... onda* iz jednostavnih sudova.

U logici sudova proučavamo i **logička zaključivanja**, te određujemo koja su korektna, a koja nisu. Promotrimo dva primjera. Zaključivanje:

$$\frac{\begin{array}{l}\text{Ako si nabavio ulaznice tada idemo na utakmicu.}\\ \text{Nabavio sam ulaznice.}\end{array}}{\text{Idemo na utakmicu.}}$$

je naravno primjer korektnog zaključivanje. Formalno zapisano ono je oblika

$$\frac{\begin{array}{l}A \to B\\ A\end{array}}{B}$$

Nadamo se da se slažete da zaključivanje:

$$\frac{\begin{array}{l}\text{U subotu ću dugo spavati.}\\ \text{Danas nije subota.}\end{array}}{\text{Danas sam rano ustao.}}$$

nije korektno. Formalno ga možemo zapisati u obliku:

$$\frac{\begin{array}{l}A \to B\\ \neg A\end{array}}{\neg B}$$

U ovom poglavlju ćemo definirati što je logička posljedica, tj. koje zaključivanje smatramo korektnim.

1.2.2 Osnovni pojmovi

Alfabet je proizvoljan neprazan skup. Svaki element alfabeta nazivamo **simbol** ili **znak. Riječ** alfabeta je svaki konačan niz danog alfabeta. **Duljina riječi** je broj simbola koji dolaze u riječi. Ako je sa A označen neki alfabet tada se skup svih riječi obično označava sa A^*. Po dogovoru smatramo da skup svih riječi proizvoljnog alfabeta sadrži **praznu riječ**, tj. prazan niz simbola. Praznu riječ obično označavamo s ϵ. Konkatenacija je binarna operacija na A^*, koja je definirana na sljedeći način:

> ako su a i b riječi (bolje reći oznake za riječi!) tada kažemo da je riječ ab nastala konkatenacijom riječi a i b.

Kažemo da je b **podriječ** riječi a ako postoje riječi c i d tako da je riječ a nastala konkatenacijom riječi c, b i d, tj. a je jednaka cbd.

Navodimo neke **primjere alfabeta**. Neka je $A_1 = \{\alpha, \beta\}$. Neke riječi tog alfabeta su npr. $\alpha\alpha\alpha$, $\alpha\beta\alpha\beta\beta\beta$, $\alpha\alpha\beta\beta\alpha\alpha\beta$. Iz riječi $\alpha\alpha\beta\beta$ i $\beta\beta\alpha\beta$ konkatenacijom dobivamo riječ $\alpha\alpha\beta\beta\beta\beta\alpha\beta$.

Neka je, zatim, $A_2 = \{+,\ \cdot,\ s,\ 0,\ =\} \cup \{x_n : n \in \mathbb{N}\}$. Tada su riječi alfabeta A_2 npr. $x_1 + x_2 = x_2$, $x_1 \cdot x_4 + 0 = x_5$, ali i $++\cdot x_4 === .$

U sljedećoj propoziciji ističemo činjenicu koju ćemo kasnije često koristiti.

Propozicija 1.1. *Skup svih riječi konačnog ili prebrojivog alfabeta je prebrojiv.*

1.2.3 Sintaksa logike sudova

Definicija 1.2. Alfabet logike sudova *je unija skupova* A_1, A_2 *i* A_3, *pri čemu je:*

$A_1 = \{P_0,\ P_1,\ P_2,\ \ldots\}$ *prebrojiv skup čije elemente nazivamo* **propozicionalne varijable**;

$A_2 = \{\neg,\ \wedge,\ \vee,\ \rightarrow,\ \leftrightarrow\}$ *skup* **logičkih veznika**;

$A_3 = \{(,\ )\}$ *skup pomoćnih simbola (zagrade).*

Uočite da smo u definiciji naveli da alfabet logike sudova sadrži znakove koje nazivamo propozicionalne varijable. Možete zamišljati da se propozicionalne varijable interpretiraju sudovima, ali to ne mora nužno biti tako. Jedna interpretacija logike sudova su i npr. elektronički logički sklopovi. U sljedećoj točki ćemo formalno definirati interpretacije propozicionalnih varijabli.

Logičke veznike redom nazivamo: $\neg$ negacija, $\wedge$ konjunkcija, $\vee$ disjunkcija, $\rightarrow$ kondicional i $\leftrightarrow$ bikondicional.

Naravno, ne zanimaju nas sve riječi alfabeta. Svakako nećemo promatrati npr. riječ $\neg\wedge)P_2()$. Sada definiramo najvažnije riječi alfabeta logike sudova, a to su formule.

Definicija 1.3. Atomarna formula *je svaka propozicionalna varijabla. Pojam* **formule** *definiramo rekurzivno:*

a) svaka atomarna formula je formula;

b) ako su A *i* B *formule tada su i sve sljedeće riječi također formule:* $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ *i* $(A \leftrightarrow B)$.

c) riječ alfabeta logike sudova je formula ako je nastala primjenom konačno mnogo koraka uvjeta a) i b).

Napomena 1.4. *Primijetimo da u prethodnoj definiciji* A *i* B *nisu formule već oznake za formule, tj. to nisu simboli jezika već su* **meta–simboli**. *Po dogovoru ćemo s velikim slovima (npr.* A, B, C, F, G, F_1, F_2, $\ldots$*) označavati formule. Za propozicionalne varijable upotrebljavat ćemo oznake* P, Q, R, S, $\ldots$

Dogovor o pisanju zagrada. Način zapisivanja formula obzirom na zagrade, kako smo definirali, naziva se **sistem vanjskih zagrada**. Zapis formula se može definirati i u **sistemu unutarnjih zagrada** ili pak u **poljskoj notaciji**, tj. bez zagrada. Ako su A i B formule, u sistemu unutarnjih zagrada definirali bi da su tada sljedeće riječi formule: $\neg(A)$, $(A) \wedge (B)$, $(A) \vee (B)$, $(A) \to (B)$, i $(A) \leftrightarrow (B)$.
U daljnjem tekstu nećemo se strogo držati zapisivanja formula pomoću zagrada, već ćemo uvesti prioritet logičkih veznika. Najveći **prioritet** ima negacija, zatim veznici $\wedge$ i $\vee$, a najmanji prioritet (ali isti) imaju veznici $\to$ i $\leftrightarrow$.
No, to ne znači da ćemo se potpuno odreći zagrada prilikom zapisivanja formula. U nekim situacijama ćemo pisati zagrade kako bi istaknuli prioritet nekog veznika. Tako bi zapis formule $((\neg P) \wedge Q) \to R)$ u sistemu unutarnjih zagrada izgledao $((\neg(P)) \wedge (Q)) \to (R)$, dok ćemo je mi obično zapisivati kao $(\neg P \wedge Q) \to R$.

Kažemo da je formula B **potformula** formule A ako je riječ B podriječ od A.

Poglavlje 2

Drugo predavanje – logika sudova

2.1 Semantika i normalne forme

2.1.1 Interpretacije i istinitost

Neka je A formula te neka je $\{P_1, \ldots, P_n\}$ skup svih propozicionalnih varijabli koje se pojavljuju u A. To kratko označavamo sa $A(P_1, \ldots P_n)$. Ponekad ćemo skup svih varijabli koje se javljaju u formuli A označavati sa $Var(A)$.

Definicija 2.1. *Svako preslikavanje sa skupa svih propozicionalnih varijabli u skup* $\{0,1\}$, *tj.* $I : \{P_0, P_1, \ldots\} \to \{0,1\}$ *nazivamo* **totalna interpretacija** *ili kratko* **interpretacija**. *Ako je preslikavanje definirano na podskupu skupa propozicionalnih varijabli tada kažemo da je to* **parcijalna interpretacija.** *Kažemo da je parcijalna interpretacija* I **adekvatna** *za formulu* $A(P_1, \ldots, P_n)$ *ako je funkcija* I *definirana na* P_i *za sve* $i = 1, \ldots, n$.

Sada rekurzivno definiramo vrijednost interpretacije na proizvoljnoj formuli, tj. istinitost, odnosno neistinitost, formule za danu interpretaciju.

Definicija 2.2. *Neka je* I *interpretacija (totalna ili parcijalna). Ako se radi o parcijalnoj interpretaciji* I *smatramo da je* I *adekvatna za formule na kojima se definira njena vrijednost. Tada vrijednost interpretacije* I *na proizvoljnoj formuli definiramo rekurzivno:*

$$
\begin{aligned}
I(\neg A) = 1 \quad & \textit{ako i samo ako} \quad I(A) = 0; \\
I(A \wedge B) = 1 \quad & \textit{ako i samo ako} \quad I(A) = 1 \;\; \textit{i} \;\; I(B) = 1; \\
I(A \vee B) = 1 \quad & \textit{ako i samo ako} \quad I(A) = 1 \;\; \textit{ili} \;\; I(B) = 1; \\
I(A \to B) = 1 \quad & \textit{ako i samo ako} \quad I(A) = 0 \;\; \textit{ili} \;\; I(B) = 1; \\
I(A \leftrightarrow B) = 1 \quad & \textit{ako i samo ako} \quad I(A) = I(B).
\end{aligned}
$$

Napomena 2.3. *Istaknimo da veznik ili shvaćamo* **inkluzivno**, *tj. da "*$I(A) = 1$ *ili* $I(B) = 1$*" znači da je ili* $I(A) = 1$, *ili* $I(B) = 1$ *ili oboje. U prirodnom (hrvatskom) jeziku se veznik ili obično promatra* **ekskluzivno**.

Preglednije je vrijednost interpretacije na formulama definirati pomoću tablica koje se nazivaju **semantičke tablice.** Tada se vrijednosti interpretacije za složenije formule mogu definirati i ovako:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \to Q$	$P \leftrightarrow Q$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Definicija 2.4. *Ako je vrijednost interpretacije I na formuli jednaka* 1, *tj.* $I(F) = 1$, *tada kažemo da je formula F* **istinita za interpretaciju** *I.*
Ako je $I(F) = 0$ *tada kažemo da je formula F* **neistinita za interpretaciju** *I. Ako je S skup formula i I neka interpretacija, sa* $I(S) = 1$ *ćemo kratko označavati činjenicu da je* $I(F) = 1$, *za sve* $F \in S$. *Analogno s* $I(S) = 0$ *označavamo činjenicu da je svaka formula iz skupa S neistinita za interpretciju I.*

Ako su A i B oznake za istu formulu tada pišemo $A \equiv B$, i govorimo da su formule A i B jednake. Znak $\equiv$ nije znak alfabeta logike sudova već je pomoćni (tj. metasimbol). Za jednakost formula ne upotrebljavamo znak $=$ jer ćemo ga kod logike predikata koristiti kao osnovni znak alfabeta.

Definicija 2.5. *Neka je S skup formula, a F neka formula. Kažemo da formula F* **logički slijedi** *iz skupa S, u oznaci* $S \models F$, *ako za svaku interpretaciju I, za koju je* $I(S) = 1$, *vrijedi* $I(F) = 1$. *Ako je S jednočlani skup, tj.* $S = \{A\}$, *tada činjenicu* $\{A\} \models B$ *zapisujemo i kao* $A \Rightarrow B$.

Definicija 2.6. *Kažemo da su formule A i B* **logički ekvivalentne**, *i označavamo* $A \Leftrightarrow B$, *ako za svaku interpretaciju I vrijedi* $I(A) = I(B)$.

Definicija 2.7. *Za formulu F kažemo da je* **ispunjiva**, *odnosno* **oboriva**, *ako postoji interpretacija I tako da vrijedi* $I(F) = 1$, *odnosno* $I(F) = 0$. *Za formulu F kažemo da je* **valjana** *(***tautologija** *ili* **identički istinita***) ako je istinita za svaku interpretaciju. Za formulu F kažemo da je* **antitautologija** *ili* **identički neistinita** *ako je neistinita za svaku interpretaciju.*

2.1.2 Normalne forme

Definicija 2.8. *Atomarnu formulu i njezinu negaciju nazivamo* **literal**. *Formulu oblika* $A_1 \wedge A_2 \wedge \ldots \wedge A_n$ *nazivamo konjunkcija (A_i su proizvoljne formule).*
Formulu oblika $A_1 \vee A_2 \vee \ldots \vee A_n$ *nazivamo disjunkcija.*
Elementarna konjunkcija *je konjunkcija literala, a* **elementarna disjunkcija** *je disjunkcija literala.*
Konjunktivna normalna forma *je konjunkcija elementarnih disjunkcija.*
Disjunktivna normalna forma *je disjunkcija elementarnih konjunkcija.*

Primjer 2.9. *Promotrimo neke primjere formula koje su normalne forme. Formula* $(P_2 \vee \neg P_3 \vee P_4) \wedge (P_7 \vee \neg P_8) \wedge (P_2 \vee P_3 \vee \neg P_3)$ *je jedna konjunktivna normalna forma, a formula* $(P_3 \wedge \neg P_7 \wedge P_9) \vee (\neg P_3 \wedge P_7 \wedge P_9) \vee (P_3 \wedge P_7 \wedge P_9)$ *je disjunktivna normalna forma.*

Definicija 2.10. *Neka je* A *neka formula, te* B *konjunktivna normalna forma i* C *disjunktivna normalna forma. Kažemo da je* B **konjunktivna normalna forma za** A *ako vrijedi* $A \Leftrightarrow B$*. Kažemo da je* C **disjunktivna normalna forma za** A *ako vrijedi* $A \Leftrightarrow C$*.*

Lako je vidjeti da ako za neku formulu postoji konjunktivna normalna (ili disjunktivna), tada za nju postoji beskonačno normalnih konjunktivnih formi. Dakle, normalne forme, ako postoje, nisu jedinstvene.

Primjer 2.11. *Neka je* $F \equiv ((P \to Q) \to (R \to \neg P)) \to (\neg Q \to \neg R)$*. Kako bismo odredili sve parcijalne interpretacije za koje je formula* F *neistinita, napišimo prvo semantičku tablicu za formulu* F*.*

P	Q	R	$P \to Q$	$R \to \neg P$	$(P \to Q) \to (R \to \neg P)$	$\neg Q \to \neg R$	F
0	*0*	*0*	*1*	*1*	*1*	*1*	*1*
0	*0*	*1*	*1*	*1*	*1*	*0*	*0*
0	*1*	*0*	*1*	*1*	*1*	*1*	*1*
0	*1*	*1*	*1*	*1*	*1*	*1*	*1*
1	*0*	*0*	*0*	*1*	*1*	*1*	*1*
1	*0*	*1*	*0*	*0*	*1*	*0*	*0*
1	*1*	*0*	*1*	*1*	*1*	*1*	*1*
1	*1*	*1*	*1*	*0*	*0*	*1*	*1*

Kako bismo odredili jednu konjunktivnu normalnu formu za formulu F *promotrimo redove u tablici, tj. interpretacije, gdje je vrijednost formule* F *jednaka* 0*. To su drugi i šesti redak tablice. U drugom retku pripadna interpretacija* I *je definirana sa* $I(P) = I(Q) = 0 \quad i \quad I(R) = 1$*. Ta interpretacija određuje elementarnu disjunkciju* $P \vee Q \vee \neg R$ *u konjunktivnoj normalnoj formi. Analogno, promatrajući šesti redak tablice, tj. interpretaciju* $I(P) = I(R) = 1 \quad i \quad I(Q) = 0$*, dobivamo elementarnu disjunkciju* $\neg P \vee Q \vee \neg R$*. S dobivene dvije elementarne disjunkcije definiramo sljedeću konjunktivnu normalnu formu:* $(P \vee Q \vee \neg R) \wedge (\neg P \vee Q \vee \neg R)$*. Lako je vidjeti da je dobivena konjunktivna normalna forma logički ekvivalentna početnoj formuli* F*.*
Možemo kratko reći da smo **konjunktivnu normalnu formu** *za formulu* F *dobili:*

- *promatrajući u njenoj sematičkoj tablici* **"nule**,
- *a zatim smo* **negirali** *propozicionalne varijable koje su* **"jedan".**

Analogno bismo dobili **disjunktivnu normalnu formu** *za formulu* F

- *promatrajući u njenoj sematičkoj tablici* **"jedinice**,
- *a zatim bismo* **negirali** *propozicionalne varijable koje su* **"nule"**.

Primjenom tog postupka dobivamo sljedeću disjunktivnu normalnu formu za F :

$$(\neg P \wedge \neg Q \wedge \neg R) \ \vee \ (\neg P \wedge Q \wedge \neg R) \vee$$

$$(\neg P \wedge Q \wedge R) \ \vee \ (P \wedge \neg Q \wedge \neg R) \vee$$

$$(P \wedge Q \wedge \neg R) \ \vee \ (P \wedge Q \wedge R)$$

Teorem 2.12. (o normalnim formama)
Za proizvoljnu formulu A *logike sudova postoje formule* B *i* C *koje su logički ekvivalentne s* A*, te je* B *u konjunktivnoj normalnoj formi, a* C *je u disjunktivnoj normalnoj formi.*

Zadaci. Odredite normalne forme sljedećih formula:

1. $((P \wedge Q) \ \vee \ (\neg P \wedge \neg R)) \ \rightarrow \ (Q \leftrightarrow R)$
2. $((P \rightarrow Q) \ \vee \ (\neg P \wedge \neg R)) \ \leftrightarrow \ (R \rightarrow R)$
3. $(\neg(Q \wedge R) \ \leftrightarrow \ (P \vee \neg R)) \ \wedge \ (Q \rightarrow R)$
4. $((P \leftrightarrow \neg Q) \ \wedge \ \neg(\neg P \wedge \neg R)) \ \rightarrow \ (\neg Q \leftrightarrow R)$
5. $((P \rightarrow Q) \ \rightarrow \ (R \rightarrow \neg P)) \ \rightarrow \ (\neg Q \rightarrow \neg R)$
6. $((C \vee G) \rightarrow (K \wedge \neg P)) \ \wedge \ ((\neg K \vee C) \rightarrow (\neg G \wedge P)) \wedge (\neg(C \wedge \neg P) \rightarrow (G \wedge \neg K))$

Poglavlje 3

Treće predavanje – logika sudova

3.1 Glavni test za logiku sudova

Uveli smo nekoliko semantičkih pojmova. To su: implikacija, logička ekvivalencija, ispunjivost, oborivost, valjanost i antitautologija. Lako je vidjeti da su svi ti pojmovi međusobno definabilni. Pogledajmo kako se svi navedeni pojmovi svode na valjanost. Očito vrijedi:

$A \Rightarrow B$	ako i samo ako	$A \to B$ je valjana formula;
$A \Leftrightarrow B$	ako i samo ako	$A \leftrightarrow B$ je valjana formula;
A je ispunjiva	ako i samo ako	$\neg A$ nije valjana formula;
A je oboriva	ako i samo ako	A nije valjana formula;
A je antitautologija	ako i samo ako	$\neg A$ je valjana formula.

Ako treba npr. ispitati ispunjivost neke formule A dovoljno je vidjeti da formula $\neg A$ nije valjana. Ovdje ćemo govoriti o načinima ispitivanja valjanosti, tj. o **testovima valjanosti**.[1]

Jedan od testova valjannosti su semantičke tablice. No, nedostatak je tablica da ispituje vrijednost formule za svaku adekvatnu interpretaciju. Ako formula sadrži n različitih propozicionalnih varijabli tada semantička tablica sadrži 2^n redaka. To je vrlo nepraktično za malo veće n–ove.

Zapravo najveći nedostatak semantičkih tablica je da je to jedna "brute-forca" metoda. Iz semantičkih tablica mi ne možemo zaključiti koji su uzroci da je neka formula ispunjiva, oboriva, valjana ili antitautologija.

Važno je uočiti da se prilikom ispitivanja valjanosti formula sematičkom tablicom ispituju vrijednosti istine za svaku adekvatnu parcijalnu interpretaciju. Testovi kod

[1]Problem ispunjivosti formule logike sudova, tj. SAT (eng. satisfaction), je jedan NP–problem. To znači da postoji algoritam koji za proizvoljnu formulu F i nedeterministički odabranu interpretaciju I u polinomnom vremenu ispita vrijedi li $I(F) = 1$. Štoviše, SAT je jedan od NP–potpunih problema, tj. svaki drugi NP–problem može se svesti na SAT. Semantičke tablice su, primjerice, jedan algoritam za ispitivanje ispunjivosti formule. No, znamo ako formula sadrži n različitih propozicionalnih varijabli, tada semantička tablica sadrži 2^n redaka. To znači da semantičke tablice nisu polinomni algoritam, već eksponencijalni.

kojih se ne određuje vrijednost istine za svaku interpretaciju, već se samo traži jedna interpretacija koja bi imala određeno svojstvo, nazivaju se **ciljani testovi**.

Ako želimo odrediti je li neka formula ispunjiva tada nas zanima postoji li neka interpretacija za koju je dana formula istinita. Ako pak želimo npr. ispitati slijedi li neka formula F logički iz nekog danog konačnog skupa S tada moramo ispitati postoji li interpretacija I za koju vrijedi $I(S) = 1$ i $I(F) = 0$.

Sada dajemo jedan primjer ciljanog testa. Nazivamo ga **glavni test.**[2]
Opišimo kratko glavni test na primjeru kada za neku formulu F treba ispitati je li valjana. Na sličan način se ispituje ispunjivost, oborivost, antitautologičnost, implikacija i logička ekvivalencija formula. Prilikom ispitivanja valjanosti formule F polazimo od pitanja postoji li interpretacija propozicionalnih varijabli za koje je dana formula neistinita. Iz tog razloga test počinje s retkom oblika: $F\ \bot$. Tada primjenom određenih pravila (obzirom na "glavni" veznik) razgrađujemo danu formulu. Ovdje znak $\bot$ koristimo kao oznaku za "formula je neistinita". Analogno upotrebljavamo znak $\top$. Sada navodimo pravila koja koristimo prilikom glavnog testa.

$(\neg)$

$$\begin{array}{l} \neg B\ \textcircled{\top} \\ B\ \bot \end{array} \qquad \begin{array}{l} \neg B\ \textcircled{\bot} \\ B\ \top \end{array}$$

$(\wedge)$

$$\begin{array}{l} B \wedge C\ \textcircled{\top} \\ B\ \top \\ C\ \top \end{array} \qquad \begin{array}{l} B \wedge C\ \textcircled{\bot} \\ \quad / \quad \backslash \\ B\ \bot \quad C\ \bot \end{array}$$

$(\vee)$

$$\begin{array}{l} B \vee C\ \textcircled{\top} \\ \quad / \quad \backslash \\ B\ \top \quad C\ \top \end{array} \qquad \begin{array}{l} B \vee C\ \textcircled{\bot} \\ B\ \bot \\ C\ \bot \end{array}$$

$(\rightarrow)$

$$\begin{array}{l} B \rightarrow C\ \textcircled{\top} \\ \quad / \quad \backslash \\ B\ \bot \quad C\ \top \end{array} \qquad \begin{array}{l} B \rightarrow C\ \textcircled{\bot} \\ B\ \top \\ C\ \bot \end{array}$$

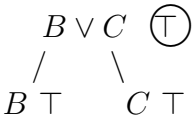
$(\leftrightarrow)$

$$\begin{array}{l} B \leftrightarrow C\ \textcircled{\top} \\ \quad / \quad \backslash \\ B\ \top \quad B\ \bot \\ C\ \top \quad C\ \bot \end{array} \qquad \begin{array}{l} B \leftrightarrow C\ \textcircled{\bot} \\ \quad / \quad \backslash \\ B\ \top \quad B\ \bot \\ C\ \bot \quad C\ \top \end{array}$$

Zaokruživanje konstante (npr. $\textcircled{\top}$) znači da se dani zahtjev svodi na nove

[2]U literaturi se glavni test naziva i semantičko stablo, odnosno semantički tableaux.

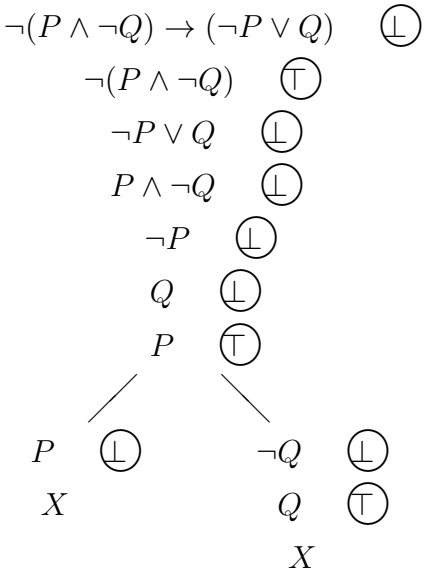
zahtjeve koji ga slijede. Za ilustraciju promotrimo sljedeće pravilo:



Iz njega čitamo da je formula $B \vee C$ istinita ako je istinita formula B ili C, tj. istinitost formule $B \vee C$ se svodi na istinitost formule B ili formule C.

Ako se prilikom ispitivanja na nekoj grani pojave reci oblika A $\textcircled{\top}$ i A $\textcircled{\bot}$ tada na toj grani prekidamo ispitivanje, te na kraj grane stavljamo oznaku X. Time smo označili da su uvjeti na egzistenciju interpretacije kontradiktorni na toj grani. Ako sve grane završe sa X tada zaključujemo da tražena interpretacija ne postoji. Inače s grane koja nije završila sa X očitavamo traženu interpretaciju.

Primjer 3.1. *Ispitajmo pomoću glavnog testa je li valjana sljedeća formula* $\neg(P \wedge \neg Q) \to (\neg P \vee Q)$.



Budući da su sve grane završile sa X zaključujemo da ne postoji interpretacija za koju bi dana formula bila neistinita. To znači da je početna formula valjana.

Glavni test uvijek završava u konačno mnogo koraka. Tada možemo vidjeti je li npr. formula valjana, ili pak možemo pročitati interpretaciju za koju je formula neistinita. To znači da za svaku formulu logike sudova možemo u konačno mnogo koraka odlučiti je li valjana. Zbog toga kažemo da je logika sudova **odlučiva teorija.**[3] To je jedna od najvećih razlika s logikom prvog reda koja nije odlučiva teorija.

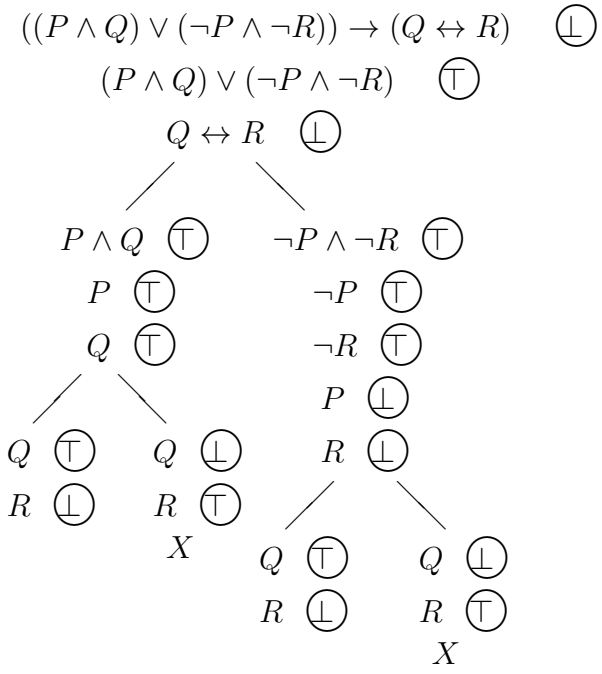
[3]Pojam odlučive teorije ne možemo ovdje strogo definirati. Trebali bismo prije definirati pojam algoritma.

Zadaci.

1. Pomoću glavnog testa ispitajte valjanost formule

$$((P \wedge Q) \vee (\neg P \wedge \neg R)) \rightarrow (Q \leftrightarrow R).$$

Rješenje:



Budući da postoje grane koje nisu završile kontradikcijom tada zaključujemo da dana formula nije valjana. Možemo očitati dvije interpretacije za koje je dana formula neistinita. Jedna je definirana sa $I(P) = I(Q) = 1$ i $I(R) = 0$. Druga je definirana sa $J(P) = J(R) = 0$ i $J(Q) = 1$.

2. Primjenom glavnog testa ispitajte je li:

   a) formula $(P_1 \rightarrow (P_1 \rightarrow P_2)) \rightarrow (P_1 \rightarrow P_2)$ valjana;

   b) formula $(P \rightarrow Q) \wedge (Q \rightarrow R) \wedge (\neg P \wedge R)$ ispunjiva;

   c) formula $\neg(\neg Q \vee P) \wedge (P \vee \neg R) \wedge (Q \rightarrow R)$ oboriva.

3. Ispitajte vrijedi li $((P \wedge Q) \vee (\neg P \wedge \neg R)) \Leftrightarrow (Q \leftrightarrow R)$

4. Ispitajte valjanost sljedećih formula:

   a) $(\neg(Q \wedge R) \leftrightarrow (P \vee \neg R)) \wedge (Q \rightarrow R)$

   b) $((P \leftrightarrow \neg Q) \wedge \neg(\neg P \wedge \neg R)) \rightarrow (\neg Q \leftrightarrow R)$

c) $((P \to Q) \ \to \ (R \to \neg P)) \ \to \ (\neg Q \to \neg R)$

5. Ispitajte vrijedi li $(P \wedge \neg Q) \Rightarrow (\neg P \leftrightarrow Q)$.

6. Danilo Blanuša bio je dugo godina profesor matematike na FER–u (točnije ETF–u). U jednoj od njegovih knjiga pod naslovom *Viša matematike* dana su sljedeća dva zadatka. U Blanušinoj knjizi zadaci su riješeni pomoću Booleovih algebri. Predlažemo da ih probate rješiti primjenom glavnog testa.

   U nekoj gostionici skupili su se mladići i djevojke na pokladnu zabavu. Gazdarica gostionice je rekla mladićima: "Svatko tko ispuni sljedeća tri uvjeta dobit će bocu šampanjca. Uvjeti glase:

   (1) Ako netko pleše s crnkom ili pleše sa mnom (inkluzivno "ili"!) onda mora plesati s konobaricom i ne smije plesati s plavušom.
   (2) Ako netko ne pleše s konobaricom ili ako pleše s crnkom, onda ne smije plesati sa mnom, ali mora plesati s plavušom.
   (3) Mora plesati sa mnom, ali ne smije s konobaricom, osim ako pleše s crnkom, ali ne s plavušom."

   Što mora učiniti mladić da besplatno dobije bocu šampanjca?

   Rješenje. Uvodimo redom sljedeće oznake za tvrdnje:

   P – "mladić je plesao s plavušom";

   C – "mladić je plesao s crnkom";

   K – "mladić je plesao s konobaricom";

   G – "mladić je plesao s gazdaricom".

   Ako simbole P, C, K i G shvatimo kao propozicionalne varijable tada se dani zadatak svodi na ispitivanje je li formula

   $$((C \vee G) \to (K \wedge \neg P)) \ \wedge \ ((\neg K \vee C) \to (\neg G \wedge P)) \wedge (\neg(C \wedge \neg P) \to (G \wedge \neg K))$$

   ispunjiva. Primjenom glavnog testa to učinite (formula je antitautologija, a to znači da niti jedan mladić ne može ispuniti gazdaričine uvjete).

7. Neki mladoženja poslije vjenčanja reće svojoj ženi: "Draga moja, dobro ćemo se slagati ako s obzirom na ručkove ispuniš ova tri uvjeta:

   1. Ako ne daš kruh na stol tada moraš dati sladoled.
   2. Ako daš kruh i sladoled, ne smiješ dati krastavce.
   3. Ako daš krastavce ili (inkluzivno 'ili'!) ne daš kruh, onda ne smiješ dati sladoled."

   Jesu li ovi uvjeti zajedno ispunjivi, i ako jesu, kako ih je moguće postići?

Poglavlje 4

Četvrto predavanje – logika sudova

4.1 Račun sudova (Frege–Łukasiewiczev sistem)

U ovoj točki prvo dajemo definiciju Frege–Łukasiewiczevog sistema, kojeg ovdje označavamo s RS (**račun sudova**). Nakon toga definiramo pojmove **dokaza, teorema i izvoda**. Prvo dokazujemo **teorem adekvatnosti** za sistem RS, kojim je iskazana korektnost sistema obzirom na semantiku definiranu u prethodnim točkama. Zatim nizom lema i **teoremom dedukcije** dajemo osnovna svojstva izvoda, odnosno dokaza. Ujedno nam te činjenice služe za dokaz glavnog teorema ove točke – **teorema potpunosti**. Sve detalje, posebno dokaze koji su ovdje ispušteni, možete pogledati u knjizi M. Vuković, Matematička logika, Element, Zagreb, 2009.

Važno je naglasiti da u ovoj točki ne promatramo isti alfabet kao prije, već samo sljedeći skup osnovnih znakova: $\{\neg, \rightarrow, \ (, \ )\} \ \cup \ \{P_0, P_1, \ldots\}$. To smanjenje nije nikakvo bitno smanjenje izražajnosti jezika logike sudova, jer se svi ispušteni veznici (tj. $\wedge$, $\vee$ i $\leftrightarrow$) mogu definirati pomoću veznika $\neg$ i $\rightarrow$. Veznike $\wedge$, $\vee$ i $\leftrightarrow$, koji nisu elementi alfabeta koristimo u zapisu formula ali ih shvaćamo kao pokrate, i to redom ovako:

$$A \wedge B \ \text{ označava } \ \neg(A \rightarrow \neg B);$$
$$A \vee B \ \text{ označava } \ \neg A \rightarrow B;$$
$$A \leftrightarrow B \ \text{ označava } \ \neg((A \rightarrow B) \rightarrow \neg(B \rightarrow A)).$$

Smanjeni alfabet bitno skraćuje dokaze koji se provode indukcijom po složenosti formule. Zbog promjene alfabeta mijenja se i definicija pojma formule. Ovdje nećemo navoditi tu izmijenjenu definiciju, već samo ističemo da sada u rekurzivnom koraku definicije imamo samo dva slučaja, i to obzirom na veznike $\neg$ i $\rightarrow$.

4.1.1 Sistem RS

Prije same definicije sistema RS objasnimo pojam aksioma i pravila izvoda. **Aksiom sistema RS je neka izabrana formula**. Zapravo, aksiome RS definiramo pomoću shema formula. Kako bi definirali pojam sheme formule moramo prvo uvesti oznake za

supstituciju. Neka je $A(P_1, \ldots, P_n)$ formula i B neka formula. Formulu dobivenu supstitucijom neke varijable P_i sa B u formuli A označavamo sa $A(B/P_i)$ Ako pak su $B_1, \ldots, B_n$ proizvoljne formule tada simultanu supstituciju varijabli P_i s formulama B_i označavamo sa $A(B_1/P_1, \ldots, B_n/P_n)$, ili pak kratko $A(B_1, \ldots, B_n)$.

Neka je $A(P_1, \ldots, P_n)$ formula. **Shema formule** A je skup svih formula oblika $A(B_1, \ldots, B_n)$, gdje su $B_1, \ldots, B_n$ oznake za proizvoljne formule. Shemu formule označavamo isto kao i formulu. Za danu shemu formule svaki njen element nazivamo **instanca**. Npr. ako je s $(P_3 \wedge P_4) \to (P_4 \leftrightarrow P_1)$ zadana shema formule, tada je jedna njena instanca $(P_7 \wedge (\neg\neg P_2 \vee P_8)) \to ((\neg\neg P_2 \vee P_8) \leftrightarrow P_1)$. Ponekad se shema formule zadaje pomoću meta–varijabli za formule. Primjerice, ako kažemo da je $A \to ((B \wedge C) \leftrightarrow \neg A)$ shema formule, tada mislimo na skup svih formula tog oblika, gdje su umjesto meta–varijabli A, B i C uvrštene proizvoljne formule. Obično ćemo sheme formula koristiti prilikom zadavanja aksioma, pa ćemo govoriti o **shemama aksioma**.

Pravilo izvoda je zadana transformacija kojom iz skupa formula dobivamo novu formulu. Obično se pravilo izvoda shematski zapisuje u obliku

$$\frac{A_1 \ \ldots \ A_n}{B}$$

a čitamo ga kao "iz skupa formula $\{A_1, \ldots, A_n\}$ slijedi formula B". Formule A_i se nazivaju **premise**, a formula B se naziva **konkluzija**.

Definicija 4.1. *Sistem RS zadan je svojim shemama aksioma i jednim pravilom izvoda. Sheme aksioma sistema RS su:*

(A1) $A \to (B \to A)$;

(A2) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$;

(A3) $(\neg B \to \neg A) \to (A \to B)$.

Jedino pravilo izvoda je **modus ponens**, *ili kratko* **mod pon** , *koje glasi:*

$$\frac{A \qquad A \to B}{B}$$

Svaku instancu neke od shema $(A1)$–$(A3)$ *nazivamo* **aksiom**.

Definicija 4.2. *Kažemo da je niz formula* $F_1, \ldots, F_n$ **dokaz** *za formulu* F *u sistemu RS ako vrijedi:*

a) formula F_n *je upravo* F, *tj. vrijedi* $F_n \equiv F$;

b) za sve $k \in \{1, \ldots, n\}$ *formula* F_k *je ili aksiom ili je nastala primjenom pravila modus ponens na neke formule* F_i *i* F_j, *gdje su* $i, j < k$.

Kažemo da je formula F **teorem** *sistema RS, u oznaci* $\vdash_{RS} F$ *(odnosno, kratko* $\vdash F$*), ako u RS postoji dokaz za* F.

Primijetimo da pojam teorema (a i dokaza) ovdje upotrebljavamo u dva smisla: teorem sistema RS i teorem koji nešto govori o sistemu RS. Ponekad se u literaturi za ovaj drugi smisao pojma teorema može naći izraz "**metateorem**". Mi ovdje nećemo činiti razliku, nadajući se da će značenje riječi "teorem" slijediti iz danog konteksta.

Teorem 4.3. (Teorem adekvatnosti za sistem RS)
Svaki teorem sistema RS je valjana formula.

Dokaz. Lako je dokazati da je svaki aksiom sistema RS valjana formula. Zatim, pravilo izvoda modus ponens **čuva istinitost**. Odnosno, ako je I interpretacija propozicionalnih varijabli takva da vrijedi $I(A) = 1$ i $I(A \to B) = 1$, tada očito mora vrijediti $I(B) = 1$.
Reći ćemo da je formula F n–**dokaziva** ($n \in \mathbb{N} \setminus \{0\}$) ako postoji barem jedan dokaz duljine n u sistemu RS za formulu F. Indukcijom po n dokazuje se da je svaka n–dokaziva formula valjana. Q.E.D.

Sada nam je glavni cilj dokazati obrat gornjeg teorema, tj. teorem potpunosti. U svrhu toga navodimo niz lema i propozicija.

Lema 4.4. *Vrijedi $\vdash A \to A$, tj. za sve formule A logike sudova formula $A \to A$ je teorem sistema RS.*

Jednom dokazani teorem od RS možemo upotrebljavati u novim dokazima. (Mogli bismo čitav dokaz teorema prepisati u novi traženi dokaz). To ćemo ilustrirati prilikom dokaza sljedećeg teorema sistema RS. Tvrdimo da za sve formule A vrijedi $\vdash \neg A \to (A \to A)$. Sljedećim nizom formula dajemo dokaz za $\neg A \to (A \to A)$.

1.	$(A \to A) \to (\neg A \to (A \to A))$	(aksiom $(A1)$)
2.	$A \to A$	(po prethodnoj lemi 4.4.)
3.	$\neg A \to (A \to A)$	(mod pon: 1. i 2.)

4.1.2 Teorem dedukcije za sistem RS

Sada prvo dajemo definiciju izvoda. U izvodima, za razliku od dokaza, osim aksioma imamo i posebno dozvoljene formule koje nazivamo pretpostavke.

Definicija 4.5. *Neka je S proizvoljan skup formula logike sudova i F neka formula. Kažemo da je niz formula $F_1, \ldots, F_n$* **izvod** *iz skupa S formule F u sistemu RS, u oznaci $S \vdash F$, ako vrijedi:*

a) formula F_n je upravo formula F, tj. imamo $F_n \equiv F$;

b) za sve $k \in \{1, \ldots, n\}$ vrijedi barem jedno od sljedećeg:

b_1) F_k je aksiom sistema RS;

b_2) $F_k \in S$ *(tada formulu* F_k *nazivamo* **pretpostavka***);*

b_3) *formula* F_k *je nastala iz nekih* F_i, F_j $(i, j < k)$ *pomoću pravila modus ponens.*

Dokaz sljedeće propozicije je sasvim analogan dokazu teorema adekvatnosti za sistem RS pa ga nećemo ovdje navoditi.

Propozicija 4.6. *Neka je* S *skup formula i* F *neka formula tako da vrijedi* $S \vdash F$. *Tada vrijedi* $S \models F$.

Sada nećemo odmah dati primjer nekog izvoda već ćemo prvo dokazati teorem dedukcije. Taj teorem nam bitno olakšava pronalaženje izvoda.

Teorem 4.7. (Teorem dedukcije za logiku sudova)
Neka je S *skup formula, te* A *i* B *formule logike sudova. Ako vrijedi* $S \cup \{A\} \vdash B$, *tada vrijedi i* $S \vdash A \to B$.

Dokaz. Za formulu F reći ćemo da je n–izvodljiva iz skupa formula $S \cup \{A\}$ ako postoji izvod formule F iz tog skupa koji je duljine n.

Indukcijom po n dokazuje se da za svaku n–izvodljivu formulu F iz skupa $S \cup \{A\}$ vrijedi $S \vdash A \to F$. Očito tada slijedi tvrdnja teorema. Q.E.D.

4.1.3 Konzistentnost

Definicija 4.8. *Za skup formula* S *kažemo da je* **konzistentan** *ako ne postoji formula* F *tako da vrijedi* $S \vdash F$ *i* $S \vdash \neg F$. *Ako skup formula nije konzistentan tada kažemo da je* **inkonzistentan**.

Primjer 4.9. *Neka je* $S = \{\neg A \to A,\ \neg A\}$. *Budući da formula* $\neg A$ *pripada skupu* S, *tada očito vrijedi* $S \vdash \neg A$. *No, iz* $S \vdash \neg A$ *i* $S \vdash \neg A \to A$ *primjenom pravila modus ponens slijedi* $S \vdash A$. *To znači da je skup formula* S *inkonzistentan.*
Ovim primjerom želimo naglasiti da prilikom definicije konzistentnosti nismo posebno istaknuli da formula F *ne mora pripadati skupu* S. *Upravo u ovom primjeru imamo situaciju da formula* $F \equiv A$ *ne pripada skupu* S, *a u drugu ruku vrijedi da su formule* F *i* $\neg F$ *izvedive iz* S.

Iz definicija konzistentnosti i izvoda lako slijedi tvrdnja sljedeće propozicije.

Propozicija 4.10. *Svaki podskup konzistentnog skupa je konzistentan. Svaki nadskup inkonzistentnog skupa je inkonzistentan.*

Kako bismo mogli navesti neki primjer konzistentnog skupa formula prvo definiramo sljedeći pojam.

Definicija 4.11. *Za skup formula S kažemo da je* **ispunjiv** *ako postoji interpretacija I tako da za sve formule $F \in S$ vrijedi $I(F) = 1$ (to kratko označavamo sa $I(S) = 1$).*

Propozicija 4.12. *Svaki ispunjiv skup formula S je konzistentan. Posebno je konzistentan skup svih teorema sistema RS.*

Dokaz. Pretpostavimo da je S inkonzistentan skup formula. Tada postoji formula F tako da vrijedi $S \vdash F$ i $S \vdash \neg F$. Iz propozicije 4.6. slijedi $S \models F$ i $S \models \neg F$. No, tada očito skup S ne može biti ispunjiv. Q.E.D.

Sada nam je cilj dokazati obrat prethodne propozicije. O tome govori generalizirani teorem potpunosti. No, prije moramo navesti još neka svojstva konzistentnih skupova formula.

Propozicija 4.13. *Skup formula je konzistentan ako i samo ako je svaki njegov konačan podskup konzistentan.*

Propozicija 4.14. *Skup formula S je konzistentan ako i samo ako iz S nije izvediva barem jedna formula.*

Propozicija 4.15. *Neka je S skup formula i F proizvoljna formula. Tada imamo:*

a) ako vrijedi $S \nvdash F$ tada je skup $S \cup \{\neg F\}$ konzistentan;

b) ako je S konzistentan skup formula i $S \vdash F$ tada je i skup formula $S \cup \{F\}$ konzistentan.

4.1.4 Potpuni skupovi formula

Kako bi dokazali generalizirani teorem potpunosti, tj. da je svaki konzistentan skup formula ispunjiv sada uvodimo pojam potpunog skupa formula.

Definicija 4.16. *Za skup formula S kažemo da je* **potpun** *ako za svaku formulu F vrijedi $S \vdash F$ ili $S \vdash \neg F$.*

Lema 4.17. (Lindenbaumova lema za logiku sudova)
Neka je S konzistentan skup formula. Tada postoji konzistentan i potpun skup formula S' takav da vrijedi $S \subseteq S'$.

Dokaz. Skup svih formula logike sudova je prebrojiv. Neka je $F_0,\ F_1,\ F_2,\ \ldots$ niz koji sadrži sve formule logike sudova. Rekurzivno definiramo niz skupova formula (S_n) na sljedeći način:

$$\begin{aligned} S_0 &= S \\ S_{n+1} &= \begin{cases} S_n \cup \{\neg F_n\}, & \text{ako } S_n \nvdash F_n \\ S_n \cup \{F_n\}, & \text{inače} \end{cases} \end{aligned}$$

Indukcijom je lako dokazati da je svaki skup formula S_n konzistentan. Tada konzistentnost skupa S' slijedi iz iz propozicije 4.13.. Lako je vidjeti da je skup S' potpun. Q.E.D.

Lema 4.18. (Lema o istinitosti za logiku sudova)
Ako je S konzistentan i potpun skup formula tada je S ispunjiv skup formula.

Dokaz. Definiramo totalnu interpretaciju I sa: $I(P) = 1$ ako i samo ako $S \vdash P$. Sada je lako indukcijom po složenosti formule F dokazati da vrijedi: $I(F) = 1$ ako i samo ako $S \vdash F$. Q.E.D.

4.1.5 Potpunost

Teorem 4.19. (Generalizirani teorem potpunosti za logiku sudova)
Skup formula je konzistentan ako i samo ako je ispunjiv.

Dokaz. Ako je skup formula ispunjiv tada iz propozicije 4.12. slijedi da je i konzistentan. Pretpostavimo sada da je skup formula S logike sudova konzistentan u odnosu na sistem RS. Iz Lindenbaumove leme, tj. leme 4.17., slijedi da postoji konzistentan i potpun skup formula S' takav da vrijedi $S \subseteq S'$. Iz leme 4.18. slijedi da je skup S' ispunjiv. Tada je očito i skup S ispunjiv. Q.E.D.

Teorem 4.20. (Jaki teorem potpunosti za sistem RS)
Neka je S skup formula i F neka formula. Tada vrijedi: $S \models F$ ako i samo ako $S \vdash F$.

Dokaz. Ako vrijedi $S \vdash F$ tada iz propozicije 4.6. slijedi $S \models F$. Dokažimo obrat. Pretpostavimo da $S \nvdash F$. Iz propozicije 4.15. a) slijedi da je skup formula $S \cup \{\neg F\}$ konzistentan. Iz generaliziranog teorema potpunosti slijedi da je taj skup i ispunjiv. To znači da postoji interpretacija I tako da vrijedi $I(S) = 1$ i $I(F) = 0$. No, tada $S \not\models F$. Q.E.D.

Teorem 4.21. (Teorem potpunosti za sistem RS)
Formula F je valjana ako i samo ako formula F je teorem sistema RS.

Teorem 4.22. (Teorem kompaktnosti za logiku sudova)
Skup formula S je ispunjiv ako i samo ako je svaki konačan podskup od S ispunjiv.

Poglavlje 5

Peto predavanje – modalna logika

5.1 Modalna logika

5.1.1 Motivacija

Prilikom definicije interpretacije kondicionala bili smo spomenuli da je u prvi mah pomalo čudan uvjet da iz $I(P) = 0$ i $I(Q) = 1$ slijedi $I(P \to Q) = 1$, odnosno da iz "laži" slijedi "sve". Obično se ta situacija naziva **paradoks materijalne implikacije**. U klasičnoj logici sudova ne možemo ispraviti nedostatke kondicionala. U tu svrhu promatraju se operatori na sudovima. Promotrimo prvo nekoliko primjera iz prirodnog jezika u kojima se pojavljuju operatori:

"Nužno će danas padati kiša."

"Moguće ću danas doći na posao."

"Ako gledam televiziju tada nužno žmirkam i zijevam."

U prethodnim rečenicama pojavljuju se dva operatora: "nužno" i "moguće," koje redom označavamo sa $\Box$ i $\Diamond$. Ti operatori su primjeri tzv. **modalnih operatora**, pa se iz tog razloga pripadne logike nazivaju **modalne logike**. Važno je reći da modalni operatori $\Box$ i $\Diamond$ nisu bulovski logički veznici, a to znači da ni njihova interpretacija ne može biti zadana pomoću neke istinosne funkcije. Pomoću modalnih operatora može se definirati tzv. **striktna implikacija**, koja se označava sa $\prec$, na sljedeći način: $A \prec B \equiv \Box\,(A \to B)$. Naravno, osnovno je pitanje koja svojstva ima upravo definirana striktna implikcija. Pojam nužnosti i mogućnosti je u nekim situacijama nejasan u intuitivnom smislu. Npr. hoćemo li formulu $\Box A \to \Box\Box A$ uzeti kao aksiom u modalnim sistemima ili ne? Upravo to je bio razlog da su početkom XX. stoljeća definirani mnogi modalni aksiomatski sistemi. Radi ilustracije mi ćemo definirati sintaksu i semantiku jednog od najjednostavnijih modalnih sistema koji se obično označava sa K.

5.1.2 Modalni sistem K

Definicija 5.1. *Alfabet modalnog sistema K sadrži alfabet klasične logike sudova i logičku konstantu $\perp$, te jedan unarni modalni operator $\Box$.*

Pojam modalne formule se definira analogno, pri čemu se još dodaje: ako je A formula tada je i $\Box\, A$ formula. Koristimo i unarni modalni operator $\Diamond$, pri čemu je $\Diamond A$ pokrata za formulu $\neg\Box\, \neg A$.

Definicija 5.2. *Modalni sistem K sadrži sljedeće aksiome:*

$$(A0) \quad \textit{sve tautologije (u novom jeziku!)}$$

$$(A1) \quad \Box\,(A \to B) \to (\Box\, A \to \Box\, B)$$

Pravila izvoda sistema K su:

$$\frac{A \qquad A \to B}{B}\ \textit{(mod pon)} \qquad i \qquad \frac{A}{\Box\, A}\ \textit{(nužnost)}$$

Sasvim analogno kao za sistem RS definiraju se pojmovi dokaza, izvoda i teorema.

Napomena 5.3. *U (A0) smo naveli da su sve tautologije aksiomi, ali promatrane u novom jeziku. Takve su očito sljedeće modalne formule: $\Box A \vee \neg\Box A$, $\Box\Diamond\Box A \to \Box\Diamond\Box A$ i $(\Diamond A \wedge \neg\Box\, A) \to \neg\Box\, A$.*

Radi ilustracije u sljedećoj propoziciji navodimo nekoliko teorema sistema K.

Propozicija 5.4. *Neka su A i B proizvoljne formule. Tada su sljedeće formule teoremi sistema K :*

a) $\Box\,(A \wedge B) \to (\Box\, A \wedge \Box\, B)$

b) $(\Box\, A \wedge \Box\, B) \to \Box\,(A \wedge B)$

c) $(\Box\, A \vee \Box\, B) \to \Box\,(A \vee B)$

d) $\Box\,(A \leftrightarrow B) \to (\Box\, A \leftrightarrow \Box\, B)$

e) $\Box\, \perp \to \Box\, A$

5.1.3 Semantika modalne logike

Definicija 5.5. *Neka je W neki neprazan skup, te $R \subseteq W \times W$ proizvoljna binarna relacija. Tada uređeni par (W, R) nazivamo* **Kripkeov okvir** *ili kratko* **okvir**. *Elemente skupa W nazivamo* **svijetovi**, *a relaciju R nazivamo* **relacija dostiživosti**.

Definicija 5.6. Kripkeov model $\mathfrak{M}$ *je uređena trojka $(W, R, \Vdash)$, gdje je (W, R) okvir, a $\Vdash$ je binarna relacija između svijetova i formula koja ima sljedeća svojstva:*

$w \nVdash \bot$
$w \Vdash \neg A$ *ako i samo ako* $w \nVdash A$
$w \Vdash A \wedge B$ *ako i samo ako* $w \Vdash A$ *i* $w \Vdash B$
$w \Vdash A \vee B$ *ako i samo ako* $w \Vdash A$ *ili* $w \Vdash B$
$w \Vdash A \to B$ *ako i samo ako* $w \nVdash A$ *ili* $w \Vdash B$
$w \Vdash A \leftrightarrow B$ *ako i samo ako* $w \Vdash A$ *je ekvivalentno sa* $w \Vdash B$
$w \Vdash \Box A$ *ako i samo ako* $\forall v(wRv$ *povlači* $v \Vdash A)$

Definicija 5.7. *Neka je $\mathfrak{M} = (W, R, \Vdash)$ Kripkeov model. Kažemo da je neka formula A* **istinita na modelu** $\mathfrak{M}$ *ako za sve svijetove $w \in W$ vrijedi $\mathfrak{M}, w \Vdash A$. To kratko označavamo sa $\mathfrak{M} \models A$. Kažemo da je formula A* **valjana** *ako za sve Kripkeove modele $\mathfrak{M}$ vrijedi $\mathfrak{M} \models A$.*

Neka je $\mathcal{F} = (W, R)$ neki Kripkeov okvir, te A neka formula. Kažemo da je formula A istinita na okviru $\mathcal{F}$ ako za svaku relaciju $\Vdash$ na okviru $\mathcal{F}$ vrijedi da za model $\mathfrak{M} = (W, R, \Vdash)$ vrijedi $\mathfrak{M} \models A$. To označavamo s $\mathcal{F} \models A$.

Kao i obično teorem adekvatnosti se lako dokazuje indukcijom po duljini dokaza.

Teorem 5.8. (Teorem adekvatnosti za sistem K)
Ako je A formula takva da $\vdash_K A$ tada je A valjana.

Ovdje samo iskazujemo teorem potpunosti za sistem K. Njegov dokaz je sličan dokazu potpunosti za logiku sudova, ali ima i specifičnosti.

Teorem 5.9. (Teorem potpunosti za sistem K)
Ako je A valjana formula tada je A teorem sistema K.

Napomena 5.10. *Za razliku od klasične logike sudova za modalnu logiku ne postoji samo jedan istaknuti sistem (kojemu su drugi sistemi ekvivalentni). Ovdje navodimo još nekoliko najčešće razmatranih proširenja sistema K.*

$$\begin{aligned} T &= K + \Box A \to A \\ S4 &= T + \Box A \to \Box \Box A \\ S5 &= T + \Diamond A \to \Box \Diamond A \end{aligned}$$

Zadaci.

1. Dokažite da je formula $\Box(p \wedge q) \to (\Box p \wedge \Box q)$ valjana.
   Rješenje. Neka je $\mathfrak{M} = (W, R)$ proizvoljan model, te $w \in W$ proizvoljan svijet. Pretpostavimo da vrijedi $\mathfrak{M}, w \Vdash \Box(p \wedge q)$. Neka je $u \in W$ proizvoljan svijet za kojeg vrijedi wRu. Iz $\mathfrak{M}, w \Vdash \Box(p \wedge q)$ i wRu slijedi $\mathfrak{M}, u \Vdash p \wedge q$. Iz ovog posljednjeg imamo $\mathfrak{M}, u \Vdash p$ i $\mathfrak{M}, u \Vdash q$. Budući da je svijet u bio proizvoljni R–sljedbenik od w tada vrijedi $\mathfrak{M}, w \Vdash \Box p$ i $\mathfrak{M}, w \Vdash \Box q$. Time smo dokazali da vrijedi $\mathfrak{M}, w \Vdash \Box p \wedge \Box q$.

2. Dokažite da je formula $\Diamond(p \vee q) \to (\Diamond p \vee \Diamond q)$ valjana.

3. Dokažite da formula $\Box p \to \Box\Diamond p$ nije valjana.
   Rješenje. Formula $\Box p \to \Box\Diamond p$ nije valjana ako postoji Kripkeov model $\mathfrak{M} = (W, R)$ i svijet $w \in W$ tako da vrijedi: $\mathfrak{M}, w \nVdash \Box p \to \Box\Diamond p$. Tada vrijedi:

$$\mathfrak{M}, w \Vdash \Box p \qquad (1)$$

$$\mathfrak{M}, w \nVdash \Box\Diamond p \qquad (2)$$

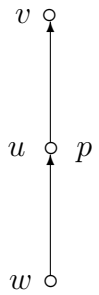
   Uvjet (1) je ekvivalentan sa:

$$\forall x \in W(wRx \Rightarrow x \Vdash p) \qquad (*)$$

   Uvjet (2) je ekvivalentan sa: $\forall x(wRx \Rightarrow x \Vdash \Diamond p)$, tj.

$$\forall x(wRx \Rightarrow \exists y(xRy \;\&\; y \Vdash p)) \qquad (**)$$

   Iz $(*)$ i $(**)$ lako slijedi da jedan model na kojem nije istinita zadana formula izgleda ovako:



   Dakle, $W = \{w, u, v\}$, $R = \{(w, u), (u, v)\}$ i $v \Vdash p$. Uočite da prepostavljamo da ne vrijedi wRv. Tada očito $w \Vdash \Box p$, a budući da wRu i $u \nVdash \Diamond p$, tada $w \nVdash \Box\Diamond p$.

4. Dokažite da su sljedeće formule teoremi sistema K :

   a) $\Box(p \wedge q) \to (\Box p \wedge \Box q)$

   b) $\Box(p \leftrightarrow q) \to (\Box p \leftrightarrow \Box q)$

   c) $\Box\bot \to \Box p$

   d) $\neg\Box\bot \to (\Box p \to \Diamond p)$

   e) $\Diamond(p \vee q) \leftrightarrow (\Diamond p \vee \Diamond q)$

   f) $(\Box p \vee \Box q) \to \Box(p \vee q$

   g) $(\Diamond p \wedge \Diamond q) \to \Diamond(p \wedge q)$

   Uputa. Iz teorema potpunosti slijedi da je dovoljno dokazati da su formule valjane.

5. Dokažite da sljedeće formule nisu teoremi sistema K :

   a) $\Box(p \vee q) \to (\Box p \vee \Box q)$

   b) $(\Box p \to \Box q) \to \Box(p \to q)$

   c) $(\Box p \leftrightarrow \Box q) \to \Box(p \leftrightarrow q)$

   d) $\Diamond\Diamond p \to \Diamond p$

   e) $p \to \Diamond p$

   f) $p \to \Box\Diamond p$

   Rješenje. Iz teorema adekvatnosti za sistem K slijedi da je dovoljno dokazati da dane formule nisu valjane.

   Rješenje zadatka 5a). Kako bi dokazali da formula $\Box(p \vee q) \to (\Box p \vee \Box q)$ nije valjana, trebamo naći barem jedan Kripkeov model $\mathfrak{M} = (W, R)$ i svijet $w \in W$ tako da vrijedi:

$$\mathfrak{M}, w \nVdash \Box(p \vee q) \to (\Box p \vee \Box q)$$

   Tada vrijedi:

$$\mathfrak{M}, w \Vdash \Box(p \vee q) \qquad (1)$$

$$\mathfrak{M}, w \nVdash \Box p \vee \Box q \qquad (2)$$

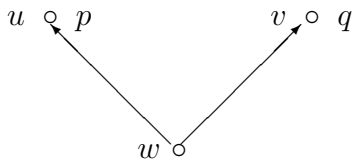
   Uvjet (1) je ekvivalentan sa:

$$\forall x \in W(wRx \Rightarrow x \Vdash p \vee q) \qquad (*)$$

   Uvjet (2) je ekvivalentan sa:

$$\exists x \exists y(wRx \ \&\ wRy \ \&\ x \nVdash p \ \&\ y \nVdash q) \qquad (**)$$

   Iz $(*)$ i $(**)$ lako slijedi da jedan model na kojem nije istinita zadana formula izgleda ovako:

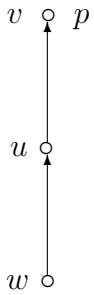


Dakle, $W = \{w, u, v\}$, $R = \{(w, u), (w, v)\}$, te $u \Vdash p$ i $v \Vdash q$. Tada očito $w \Vdash \Box(p \vee q)$. Budući da wRu i $u \nVdash q$, tada $w \nVdash \Box q$. Zatim, budući da wRv i $v \nVdash p$, tada $w \nVdash \Box p$. Dakle $w \nVdash \Box p \vee \Box q$.

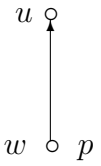
Uputa za rješenje zadatka 5b): protuprimjer je model kao u rješenju zadatka 3. uz dodatak da je relacija R tranzitivna.

Uputa za rješenje zadatka 5c): protuprimjer može biti isti model kao u rješenju zadatka 5b).

Rješenje zadatka 5d). Jedan Kripkeov model $\mathfrak{M}$ na čijem svijetu w nije istinita formula $\Diamond\Diamond p \to \Diamond p$ dan je sljedećom slikom:



Rješenje zadatka 5e). Primjer Kripkeovog modela $\mathfrak{M}$ na čijem svijetu w nije istinita formula $p \to \Diamond p$ dan je sljedećom slikom:



Uputa za rješenje zadatka 5f): Kao jedan protuprimjer može poslužiti model iz rješenja zadatka 5d).

6. Neka je $\mathcal{F} = (W, R)$ proizvoljni okvir. Dokažite:

   a) $\mathcal{F} \Vdash (\Diamond\Diamond p \to \Diamond p)$ ako i samo ako $\mathcal{F}$ je tranzitivan okvir.
   Zatim, dokažite da postoji okvir $\mathcal{F}$ koji nije tranzitivan, valuacija na $\mathcal{F}$ i $w \in \mathcal{F}$ tako da vrijedi $(\mathcal{F}, V), w \Vdash \Diamond\Diamond p \to \Diamond p$.

Uputa. Dovoljno je konstruirati model tako da vrijedi $w \nVdash \Diamond\Diamond p$. Primjerice, neka je $W = \{w, u, v\}$, $R = \{(w, u),\ (u, v))\}$ i $V(p) = \emptyset$. Uočite da relacija R nije tranzitivna, te vrijedi $w \Vdash \Diamond\Diamond p \to \Diamond p$.

b) $\mathcal{F} \Vdash (p \to \Diamond p)$ ako i samo ako $\mathcal{F}$ je refleksivan okvir.
Zatim, dokažite da postoji okvir $\mathcal{F}$ koji nije refleksivan, valuacija na $\mathcal{F}$ i $w \in \mathcal{F}$ tako da vrijedi $(\mathcal{F}, V), w \Vdash p \to \Diamond p$.

c) $\mathcal{F} \Vdash (p \to \Box\Diamond p)$ ako i samo ako $\mathcal{F}$ je simetričan okvir.
Zatim, dokažite da postoji okvir $\mathcal{F}$ koji nije simetričan, valuacija na $\mathcal{F}$ i $w \in \mathcal{F}$ tako da vrijedi $(\mathcal{F}, V), w \Vdash p \to \Box\Diamond p$.

7. Dokažite da je svaka tautologija klasične logike sudova valjana modalna formula.
Uputa. Neka je φ neka formula logike sudova koja nije valjana modalna formula. Tada postoji Kripkeov model $\mathfrak{M} = (W, R)$ i svijet $w \in W$ tako da vrijedi $\mathfrak{M}, w \nVdash \varphi$. Definiramo parcijalnu interpretaciju $I : Var(\varphi) \to \{0, 1\}$ sa:

$$I(p) = \begin{cases} 1, & \text{ako } \mathfrak{M}, w \Vdash p, \\ 0, & \text{ako } \mathfrak{M}, w \nVdash p \end{cases}$$

Dokažite da tada mora vrijediti $I(\varphi) = 0$.

Poglavlje 6

Šesto predavanje – Logika prvog reda

6.1 Uvod

Prisjetimo se glavnih rezultata klasične logike sudova. To su prije svega: teorem kompaktnosti, teoremi adekvatnosti i potpunosti. No, bez obzira na to logika sudova je vrlo slaba teorija. U logici sudova je nemoguće opisati neka logička zaključivanja, te opisati neke pojmove. Pokušat ćemo to ilustrirati sljedećim primjerom. Neka je $f : \mathbb{R} \to \mathbb{R}$ neprekidna funkcija u točki $x_\circ$, tj. istinita je formula

$$\forall \epsilon \; \exists \delta \; \forall x \; (\mid x - x_\circ \mid < \delta \quad \to \mid f(x) - f(x_\circ) \mid < \epsilon).$$

Negacija prethodne tvrdnje, tj. formula

$$\neg \forall \epsilon \; \exists \delta \; \forall x \; (\mid x - x_\circ \mid < \delta \quad \to \mid f(x) - f(x_\circ) \mid < \epsilon),$$

je formalni zapis činjenice da funkcija f ima prekid u nekoj točki $x_\circ$. Primjenom **pravila prijelaza** za kvantifikatore (sistematski ćemo ih proučavati u nekoj od sljedećih točaka) dobivamo

$$\exists \epsilon \; \forall \delta \; \exists x \; (\mid x - x_\circ \mid < \delta \;\; \wedge \;\; \mid f(x) - f(x_\circ) \mid \geq \epsilon).$$

U logici prvog reda (**predikatna logika** ili **kvantifikacijska logika**) kao osnovne znakove alfabeta imamo i varijable za objekte ili **individualne varijable**, relacijske simbole, funkcijske simbole i konstantske simbole, te kvantifikatore: $\forall$ i $\exists$.

Prelaskom na opsežniji alfabet neka dobra svojstva se gube. Jedno takvo svojstvo koje ima logika sudova, ali ne i logika prvog reda, je **odlučivost**. Definiciju pojma odlučivosti moguće je dati tek nakon proučavanja pojma izračunljivosti (npr. rekurzivne funkcije, RAM–strojevi, Turingovi strojevi, λ–račun, ...). Za svaku formulu logike sudova možemo u konačno mnogo koraka provjeriti je li valjana, tj. je li istinita za svaku interpretaciju (npr. tablica istinitosti ili glavni test). No, to nije moguće za formule logike prvog reda, tj. ne postoji algoritam koji bi primijenjen na

proizvoljnu formulu u konačno mnogo koraka davao odgovor je li dana formula valjana (**Churchov teorem**).

Objasnimo što znači logika prvog reda. U tu svrhu promotrimo sljedeća dva primjera formula.

$$\forall x \exists R \exists f (R(x) \to f(x) = x) \tag{1}$$
$$\forall R \exists P \forall g (R(x, y) \to g(P, R) = y) \tag{2}$$

U primjeru (1) je dana formula logike drugog reda jer kvantificiramo po relacijama i funkcijama koje se odnose na individualne varijable. U primjeru (2) je formula logike trećeg reda jer kvantificiramo po funkciji g čiji su argumenti funkcije i relacije. Dobar primjer formule logike drugog reda je aksiom matematičke indukcije:

$$\forall P((P(0) \ \wedge \ \forall x(P(x) \to P(x+1))) \ \to \ \forall x P(x)).$$

Čak i u logici prvog reda ne možemo izraziti sve osnovne matematičke pojmove kao što su npr. konačan skup ili pak prebrojiv skup. Važno je još reći da u ovom drugom poglavlju ne proučavamo samo jednu teoriju – logiku prvog reda, već općenito teorije prvog reda.

6.2 Jezik teorija prvog reda

U čitavoj ovoj točki govorimo **teorija prvog reda** iako taj pojam nismo definirali (i još ne možemo!). Da bi se zadala teorija prvog reda treba definirati pripadni alfabet i skup aksioma.

Definicija 6.1. *Alfabet neke teorije prvog reda je unija skupova $A_1, \ldots, A_6$ gdje su redom skupovi A_i definirani s:*

$A_1 = \{v_0, \ v_1, \ \ldots\}$, *prebrojiv skup čije elemente nazivamo* **individualne varijable**.

$A_2 = \{\neg, \ \wedge, \ \vee, \ \to, \ \leftrightarrow, \ \forall, \ \exists\}$, *skup* **logičkih simbola**, *koje redom nazivamo: negacija, konjunkcija, disjunkcija, kondicional, bikondicional, univerzalni i egzistencijalni kvantifikator.*

$A_3 = \{R_k^{n_k} \ : \ k \in I\}$, *skup čije elemente nazivamo* **relacijski simboli.** *Skup I je neki podskup $\mathbb{N}$. Prirodan broj n_k se naziva mjesnost relacijskog simbola. Pretpostavljamo da ovaj skup sadrži barem jedan dvomjesni relacijski simbol.*

$A_4 = \{f_k^{m_k} \ : \ k \in J\}$, *skup čije elemente nazivamo* **funkcijski simboli.** *Skup J je neki podskup $\mathbb{N}$, možda i prazan. Prirodan broj m_k se naziva mjesnost funkcijskog simbola.*

$A_5 = \{c_k \ : \ k \in K\}$, *skup čije elemente nazivamo* **konstantski simboli.** *Skup K je neki podskup $\mathbb{N}$, možda i prazan.*

$A_6 = \{(\ )\ ,\}$, *skup pomoćnih simbola (lijeva i desna zagrada, te zarez).*

Bili smo naveli da elemente skupa A_1 nazivamo **individualne varijable**. To znači da će te varijable poprimati vrijednosti nekih individua, odnosno objekata. To mogu biti brojevi, vektori, pravci, riječi nekog alfabeta, ... Upravo definirani alfabet je prebrojiv. Može se promatrati i neprebrojiv skup individualnih varijabli (to je razuman zahtjev kada proučavamo skup realnih brojeva). U definiciji smo naveli da pretpostavljamo da skup A_3 sadrži barem jedan dvomjesni relacijski simbol (rezerviran je za relaciju jednakosti). Skupovi funkcijskih i konstantskih simbola neke teorije prvog reda mogu biti i prazni.

Obično se unija skupova relacijskih, funkcijskih i konstantskih simbola naziva **skup nelogičkih simbola** ili **signatura**.

Smatrat ćemo da je definiran alfabet neke teorije prvog reda ako smo zadali skup nelogičkih simbola, tj. signaturu σ.

Definicija 6.2. *Skup nelogičkih simbola logike prvog reda je unija sljedećih skupova:*

- *prebrojiv skup relacijskih simbola,*
- *prebrojiv skup funkcijskih simbola i*
- *prebrojiv skup konstantskih simbola.*

Štoviše, smatramo da za svaki $k \in \mathbb{N}$ *postoji prebrojivo mnogo relacijskih i funkcijskih simbola mjesnosti* k.

Za **Peanovu aritmetiku** skup nelogičkih simbola je: $\sigma_{PA} = \{0, s, +, \cdot, =\}$, gdje je 0 konstantski simbol, s je jednomjesni funkcijski simbol (interpretira se funkcijom sljedbenika), a $+$ i $\cdot$ su dvomjesni funkcijski simboli čije su standardne interpretacije jasne.

Skup nelogičkih simbola **Zermelo–Fraenkelove teorije skupova** uz simbol za jednakost sadrži još samo jedan dvomjesni relacijski simbol. Obično se taj simbol označava s $\in$. Dakle, $\sigma_{ZF} = \{\epsilon, =\}$.

Dogovor o oznakama (1). Prilikom pisanja raznih riječi alfabeta neke teorije prvog reda nećemo koristiti samo simbola iz alfabeta. Tako ćemo obično umjesto individualnih varijabli v_i pisati znakove $x,\ y,\ z,\ x_1,\ x_2,\ \ldots$ Umjesto relacijskih simbola pisat ćemo znakove $P,\ Q,\ R,\ \ldots,$ umjesto funkcijskih simbola pisat ćemo $f,\ g,\ h,\ \ldots$ Mjesnost ćemo obično ispuštati ako će se iz samog zapisa moći lako odrediti.

U daljnjem tekstu smatramo da je fiksirana neka signatura σ **neke teorije prvog reda**.

Definicija 6.3. σ**–term** *ili kratko* **term** *je riječ pripadnog alfabeta definirana sljedećom rekurzivnom definicijom:*

a) svaka individualna varijabla i konstantski simbol koji pripada σ su σ–termi;

b) ako je f^n neki $n-$mjesni funkcijski simbol koji je element od σ, i $t_1, \ldots, t_n$ σ–termi, tada je riječ $f^n(t_1, \ldots, t_n)$ σ–term;

c) riječ je σ–term ako i samo ako je nastala pomoću konačno mnogo primjena prethodno dva navedena pravila.

Navodimo neke primjere terma: x, z, c_{23}, c_7, $f(x)$, $g(x, y, z)$ Uočite da će interpretacija svakog terma biti neki objekt skupa u kojem valuiramo individualne varijable.

Definicija 6.4. *Ako je R^n neki $n-$mjesni relacijski simbol koji pripada zadanoj signaturi σ, i $t_1, \ldots, t_n$ su σ–termi, tada riječ $R^n(t_1, \ldots, t_n)$ nazivamo* **atomarna formula.**

Definicija 6.5. *σ–***formula***, ili kratko* **formula***, je riječ pripadnog alfabeta $\mathcal{A}$ definirana sljedećom rekurzivnom definicijom:*

a) svaka atomarna formula je formula;

b) ako su A i B formule tada su $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \to B)$ i $(A \leftrightarrow B)$ također formule;

c) ako je A formula, a x varijabla, tada su riječi $(\forall x A)$ i $(\exists x A)$ također formule;

d) riječ alfabeta $\mathcal{A}$ je formula ako i samo ako je nastala primjenom konačno mnogo puta prethodno navedena tri pravila.

Uočite da za svaku signaturu σ teorije prvog reda postoje atomarne formule jer smo u definiciji alfabeta zahtijevali da je skup relacijskih simbola neprazan. U uvjetu c) ne zahtijevamo da formula A sadrži varijablu x.

Ovdje ćemo upotrebljavati istu **konvenciju za ispuštanje zagrada** kao kod logike sudova, tj. prioritet logičkih veznika i kvantifikatora je od najvećeg do najmanjeg dan sljedećom slikom:

$$\forall \quad \exists \quad \neg$$

$$\wedge \quad \vee$$

$$\to \quad \leftrightarrow$$

U isti redak su stavljeni simboli koji imaju isti prioritet. Radi izbjegavanja zabune (kad npr. u formuli dolaze simboli s istim prioritetom) ponekad ćemo pisati zagrade.

Kod formula oblika $\forall x A$ i $\exists x A$ formulu A nazivamo **doseg kvantifikatora** $\forall x$, odnosno $\exists x$. **Složenost formule** je broj kvantifikatora i logičkih veznika koji se u njoj javljaju. **Potformula** dane formula je podriječ formule koja je i sama formula.

Shema formule je riječ sagrađena od meta–varijabli za formule ($A,\ B,\ C, \ldots$) pomoću pravila koja vrijede za formule. Uvrštavanjem u shemu formule umjesto meta–varijabli konkretnih formula dobiva se formula koju nazivamo **instanca.** Shema formule je zapravo oznaka za određeni skup formula.

Definicija 6.6. *U svakoj atomarnoj formuli svaki nastup varijable je* **slobodan**. *U formulama oblika $\forall xA$ i $\exists xA$ svaki nastup varijable x je* **vezan.** *Ako varijabla x ima slobodan (vezan) nastup u formuli A, i B je proizvoljna formula, tada je taj nastup varijable x slobodan (odnosno vezan) i u formulama $\neg A,\ A \wedge B,\ A \vee B,\ A \to B,$ $A \leftrightarrow B,\ \forall yA\ \ i\ \ \exists yA$, gdje je y varijabla različita od x.*

Primjer 6.7. *U formuli $P(x, y) \to \forall xR(x)$ imamo tri nastupa varijable x. Prvi nastup je slobodan, a sljedeća dva su vezana. Nastup varijable y je slobodan. U formuli $\forall x(P(x, y) \to \forall xR(x))$ svi nastupi varijable x su vezani, a nastup varijable y je slobodan.*

Definicija 6.8. *Za neku varijablu x kažemo da je* **slobodna varijabla** *u formuli A ako postoji barem jedan njezin nastup u formuli koji je slobodan. Inače kažemo da je to* **vezana varijabla** *u formuli. Po dogovoru smatramo da je svaki nastup varijable u proizvoljnom termu slobodan. Formula koja ne sadrži slobodne varijable naziva se* **zatvorena formula** *ili* **rečenica**. *Formula koja ne sadrži kvantifikatore naziva se* **otvorena formula**.

Za formulu A sa $A(x_1,\ \ldots,\ x_n)$ označavamo da varijabe $x_1,\ \ldots,\ x_n$ mogu doći slobodne u formuli A. To ne znači da svaka od varijabli $x_1,\ \ldots,\ x_n$ dolazi slobodna u formuli A, niti znači da se u nizu $x_1,\ \ldots,\ x_n$ nalaze sve slobodne varijable formule A. Oznaka $A(x_1,\ \ldots,\ x_n)$ služit će nam da nakon supstitucije neke varijable x_i s termom t pišemo

$$A(x_1,\ \ldots,\ x_{i-1},\ t/x_i,\ x_{i+1},\ \ldots,\ x_n).$$

Sada želimo definirati **supstituciju varijable u formuli s danim termom**. No, moramo postaviti još jedan uvjet da bi nam supstitucije bile ispravne, tj. da čuvaju istinitost. U vezi toga promotrimo sljedeći primjer.

Primjer 6.9. *Neka je $A(y)$ formula $\forall x\exists yR(x, y) \to \exists xR(x, y)$, a term t neka je varijabla x. Potformula $\forall x\exists yR(x, y)$ od $A(y)$ se posebno može promatrati kao zapis o funkcionalnosti relacije R. Promotrimo što dobivamo supstitucijom varijable y s termom t. Nakon supstitucije imamo formulu $A(t/y)$, tj. $\forall x\exists yR(x, y) \to \exists xR(x, x)$. Ova posljednja formula posebno izražava da svaka funkcija ima fiksnu točku, što naravno nije istina. To znači da ne možemo dozvoliti proizvoljne supstitucije terma u formulu. Uočimo da je drugi nastup varijable y u početnoj formuli bio slobodan, a varijabla x iz terma t nakon supstitucije postaje vezana.*

Definicija 6.10. *Kažemo da je* **term** t **slobodan za varijablu** x **u formuli** A *ako niti jedan slobodan nastup varijable* x *ne leži u dosegu kvantifikatora* $\forall y$ *ili* $\exists y$*, gdje je* y *proizvoljna varijabla terma* t*.*

Ako je term t *slobodan za varijablu* x *u formuli* A *tada pod supstitucijom varijable* x *termom* t *podrazumijevamo zamjenu svakog slobodnog nastupa varijable* x *termom* t*.*

Ako formula ne sadrži slobodnih nastupa varijable x *smatramo da je supstitucija moguća, ali je nakon supstitucije početna formula nepromijenjena.*

Primjer 6.11. *a) Term* $f(x_1, x_3)$ *je slobodan za varijablu* x_1 *u formuli* $\forall x_2 P(x_1, x_2) \to R(x_1)$*, ali nije slobodan za varijablu* x_1 *u formuli* $\exists x_3 \forall x_2 P(x_1, x_2) \to R(x_1)$*.*

b) Svaki term koji ne sadrži varijable (izgrađen je iz konstantskih i funkcijskih simbola) slobodan je za svaku varijablu u svakoj formuli.

c) Promotrimo na kraju jedan primjer iz matematičke analize. Neka je s $F(y) = \int_0^1 (x + y)dx$*. Lako je vidjeti da vrijedi* $F(y) = y + \frac{1}{2}$*. Ako bismo u izrazu* $F(y)$ *varijablu* y *zamijenili sa* x *dobili bismo* $\int_0^1 2x\, dx$*, tj.* 1*. Naravno, takvu supstituciju ne bismo nikada napravili, jer "term* x *nije slobodan za varijablu* y *u formuli* $F(y)$*."*

Poglavlje 7

Sedmo predavanje - logika prvog reda

7.1 Semantika teorija prvog reda

7.1.1 Interpretacije i modeli

U ovoj točki ćemo definirati **semantiku za teorije prvog reda**. Interpretacija svakom nelogičkom simbolu pridružuje neki objekt: konstantu, relaciju ili funkciju. Pomoću pojma interpretacije definirat ćemo istinitost formule. To više neće biti jednostavno kao kod logike sudova jer će istinitost formule ovisiti i o valuaciji slobodnih varijabli. Na kraju ćemo definirati pojam ispunjive, oborive i valjane formule.

U čitavoj ovoj točki sa σ označavamo proizvoljan, ali fiksiran, skup nelogičkih simbola (za neku teoriju prvog reda).

Definicija 7.1. *σ–**struktura** je uređeni par $\mathfrak{M} = (M, \varphi)$, gdje je M neprazni skup koji nazivamo* **nosač**, *a φ je preslikavanje sa skupa nelogičkih simbola σ koje ima sljedeća svojstva:*

a) svakom relacijskom simbolu $R_k^{n_k}$ iz σ pridružuje se n_k-mjesna relacija $\varphi(R_k^{n_k})$ na M;

b) svakom funkcijskom simbolu $f_k^{m_k}$ iz σ pridružuje se m_k-mjesna funkcija $\varphi(f_k^{m_k})$ sa M^{m_k} u M;

c) svakom konstantskom simbolu c_k iz σ pridružuje se neki element $\varphi(c_k)$ iz M.

Primjer 7.2. *Za teoriju PA (Peanovu aritmetiku), čiji je skup nelogičkih simbola $\sigma_{PA} = \{0, s, +, \cdot, =\}$, jedna σ_{PA}–struktura je $(\mathbb{N}, \varphi)$, gdje imamo redom: $\varphi(0)$ je broj nula, $\varphi(+)$ je zbrajanje na skupu $\mathbb{N}$, $\varphi(\cdot)$ je množenje na skupu $\mathbb{N}$, $\varphi(s)$ je funkcija sljedbenika i binarni relacijski simbol $=$ se interpretira relacijom jednakosti na $\mathbb{N}$.*

Tu strukturu za teoriju PA označavamo s $(\mathbb{N}, 0, s, +, \cdot)$, i nazivamo je **standardni model** *za PA.*

Definicija 7.3. *Kardinalnost σ-strukture $\mathfrak{M} = (M,\ \varphi)$ je kardinalni broj skupa M, pa ćemo tako govoriti o konačnoj, prebrojivoj i beskonačnoj σ-strukturi.*

Definicija 7.4. *Za danu σ-strukturu $\mathfrak{M} = (M,\ \varphi)$ svaku funkciju sa skupa individualnih varijabli u nosač strukture nazivamo* **valuacija.**

Lema 7.5. *Neka je $\mathfrak{M} = (M,\ \varphi)$ σ–struktura i v neka valuacija. Postoji jedinstveno proširenje v' od v koje je definirano na skupu svih σ–terma, te v' ima sljedeća svojstva:*

$$\begin{aligned} v'(v_k) &= v(v_k), \\ v'(c_k) &= \varphi(c_k), \\ v'(f(t_1,\ \ldots,\ t_n)) &= \varphi(f)(v'(t_1),\ \ldots,\ v'(t_n)), \end{aligned}$$

za sve varijable v_k, sve konstantske simbole c_k i sve funkcijske simbole f iz S, te za sve terme t_i koji su definirani pomoću skupa S.

Nadalje smatramo da je svaka valuacija definirana na skupu svih terma, i to na način kao što je definirano u iskazu prethodne leme.

Definicija 7.6. *Svaki uređeni par neke σ-strukture $\mathfrak{M} = (M, \varphi)$ i proizvoljne valuacije v na M nazivamo* **σ-interpretacija**, *ili kratko* **interpretacija**.

Za danu valuaciju v i varijablu x sa v_x označavamo svaku valuaciju koja se podudara sa v na svim varijablama osim možda na varijabli x.

Definicija 7.7. *Neka je $(\mathfrak{M}, v)$ neka σ-interpretacija, gdje je $\mathfrak{M} = (M,\ \varphi)$.*

Istinitost σ–formule *F za danu interpretaciju, u oznaci $\mathfrak{M} \models_v F$, definiramo rekurzivno po složenosti formule F :*

a) ako je F atomarna formula, tj. F je oblika $R(t_1,\ \ldots,\ t_n)$, tada definiramo:

$$\mathfrak{M} \models_v F \quad \text{ako i samo ako} \quad (v(t_1),\ \ldots,\ v(t_n)) \ \in \varphi(R);$$

b) ako je F formula oblika $\neg G$ tada definiramo:

$\mathfrak{M} \models_v F$ *ako i samo ako nije* $\mathfrak{M} \models_v G$;

c) ako je F formula oblika $A \wedge B$ tada definiramo:

$\mathfrak{M} \models_v F$ *ako i samo ako* $\mathfrak{M} \models_v A$ *i* $\mathfrak{M} \models_v B$;

d) ako je F formula oblika $A \vee B$ tada definiramo:

$\mathfrak{M} \models_v F$ *ako i samo ako* $\mathfrak{M} \models_v A$ *ili* $\mathfrak{M} \models_v B$;

e) ako je F formula oblika $A \to B$ tada definiramo:

$\mathfrak{M} \models_v F$ *ako i samo ako ne vrijedi* $\mathfrak{M} \models_v A$ *ili vrijedi* $\mathfrak{M} \models_v B$;

f) ako je F formula oblika $A \leftrightarrow B$ tada definiramo:

$\mathfrak{M} \models_v F$ *ako i samo ako vrijedi da je* $\mathfrak{M} \models_v A$ *ekvivalentno s* $\mathfrak{M} \models_v B$;

g) ako je F formula oblika $\forall x G$ tada definiramo:

$\mathfrak{M} \models_v F$ *ako i samo ako* $\mathfrak{M} \models_{v_x} G$ *za sve valuacije* v_x;

h) ako je F formula oblika $\exists x G$ tada definiramo:

$\mathfrak{M} \models_v F$ *ako i samo ako* $\mathfrak{M} \models_{v_x} G$ *za neku valuaciju* v_x.

U daljnjem tekstu umjesto "nije $\mathfrak{M} \models_v F$" pisat ćemo kratko $\mathfrak{M} \not\models_v F$, i govorit ćemo da je formula F **neistinita** za danu interpretaciju.

Uočimo da definicija istinitosti formule oblika $\forall x F(x)$ ne može biti: "za sve $m \in M$ vrijedi $\mathfrak{M} \models F(m)$". Problem je u tome da elementi nosača M nisu elementi alfabeta, tj. riječ $F(m)$ nije formula.

Neka je Γ skup formula, te $\mathfrak{M} = (M, \varphi)$ struktura za Γ i v valuacija na M. Sa $\mathfrak{M} \models_v \Gamma$ kratko označavamo da za sve $F \in \Gamma$ vrijedi $\mathfrak{M} \models_v F$.

Definicija 7.8. *Kažemo da je σ–formula F* **ispunjiva** *(***oboriva***) ako postoji σ–interpretacija $(\mathfrak{M}, v)$ tako da vrijedi $\mathfrak{M} \models_v F$ $(\mathfrak{M} \not\models_v F)$.*

Kažemo da je σ–struktura $\mathfrak{M}$ **model** *za σ–formulu F ako vrijedi $\mathfrak{M} \models_v F$ za sve valuacije v. Tu činjenicu označavamo sa $\mathfrak{M} \models F$.*

Kažemo da je σ–formula **valjana** *ako je istinita za svaku σ–interpretaciju.*

Definicija 7.9. *Neka je F neka σ–formula i neka je Γ neki skup σ–formula. Kažemo da formula F* **logički slijedi** *iz skupa formula Γ ako za svaku σ–strukturu $\mathfrak{M}$ vrijedi da $\mathfrak{M} \models \Gamma$ povlači $\mathfrak{M} \models F$. To kratko označavamo sa $\Gamma \models F$. Ako je skup Γ jednočlan, tj. $\Gamma = \{A\}$, tada umjesto $\{A\} \models B$ ponekad pišemo $A \Rightarrow B$.*

Definicija 7.10. *Kažemo da su σ–formule A i B* **logički ekvivalentne** *ako vrijedi $A \Rightarrow B$ i $B \Rightarrow A$. Tu činjenicu označavamo sa $A \Leftrightarrow B$.*

7.1.2 Preneksna normalna forma

U prvom poglavlju dokazali smo da za svaku formulu logike sudova postoji njoj ekvivalentna konjuktivna i disjunktivna normalna forma. U ovoj točki cilj nam je dokazati sličan rezultat za teorije prvog reda. No, ovdje nas više zanimaju kvantifikatori, tj. kako kvantifikatore staviti ispred formule.

U čitavoj točki pretpostavljamo da je zadana neka signatura σ.

Lema 7.11. (Lema o pravilima prijelaza za kvantifikatore) *Neka su A, B i C σ–formule, te neka formula B ne sadrži slobodne nastupe varijable x. Tada vrijedi:*

1. $\neg\exists x A \Leftrightarrow \forall x(\neg A)$;
2. $\neg\forall x A \Leftrightarrow \exists x(\neg A)$;
3. $(\forall x A \to B) \Leftrightarrow \exists x(A \to B)$;
4. $(\exists x A \to B) \Leftrightarrow \forall x(A \to B)$;
5. $(B \to \forall x A) \Leftrightarrow \forall x(B \to A)$;
6. $(B \to \exists x A) \Leftrightarrow \exists x(B \to A)$;
7. $(B \wedge \forall x A) \Leftrightarrow \forall x(B \wedge A)$;
8. $(\forall x A \wedge B) \Leftrightarrow \forall x(A \wedge B)$;
9. $(B \wedge \exists x A) \Leftrightarrow \exists x(B \wedge A)$;
10. $(\exists x A \wedge B) \Leftrightarrow \exists x(A \wedge B)$;
11. $(B \vee \forall x A) \Leftrightarrow \forall x(B \vee A)$;
12. $(\forall x A \vee B) \Leftrightarrow \forall x(A \vee B)$;
13. $(B \vee \exists x A) \Leftrightarrow \exists x(B \vee A)$;
14. $(\exists x A \vee B) \Leftrightarrow \exists x(A \vee B)$;
15. $(\forall x A \wedge \forall x C) \Leftrightarrow \forall x(A \wedge C)$;
16. $(\exists x A \vee \exists x C) \Leftrightarrow \exists x(A \vee C)$.

Dokaz. Pomoću definicija istinosti i ekvivalencija formula lako je provjeriti svaku od navedenih ekvivalencija. Q.E.D.

Lema 7.12. (Lema o zamjeni vezane vaijable) *Neka je A σ–formula i x varijabla za koju postoji vezani nastup u formuli A. Zatim, neka je y varijabla koja ne nastupa u formuli A. Označimo s A' formulu dobivenu iz A zamjenom svakog vezanog nastupa varijable x s y. Tada vrijedi $A \Leftrightarrow A'$.*

Definicija 7.13. *Za σ–formulu $Q_1x_1 \ \ldots \ Q_mx_mA$ kažemo da je u* **preneksnoj normalnoj formi**, *ako je A otvorena formula, a Q_i je simbol $\forall$ ili $\exists$, za $i = 1, \ldots, m$. Po definiciji smatramo da je svaka otvorena formula u preneksnoj normalnoj formi.*

Teorem 7.14. (Teorem o preneksnoj normalnoj formi) *Za svaku σ–formulu F postoji σ–formula F' u preneksnoj normalnoj formi tako da vrijedi $F \Leftrightarrow F'$. Formulu F' nazivamo preneksna normalna forma za formulu F.*

Primjer 7.15. *Neka je*

$$F \equiv \forall x \exists y R(x,y) \to \neg \forall z P(z),$$

gdje je R dvomjesni, a P jednomjesni, relacijski simbol. Odredimo preneksnu normalnu formu formule F. Uočimo da nije potrebno mijenjati vezane varijable, tj. primjenjivati lemu o zamjeni vezanih varijabli. Primjenom pravila prijelaza redom imamo ekvivalencije:

$$F \Leftrightarrow \forall x \exists y R(x,y) \to \exists z(\neg P(z)) \Leftrightarrow$$

$$\exists x(\exists y R(x,y) \to \exists z(\neg P(z))) \Leftrightarrow$$

$$\exists x \forall y(R(x,y) \to \exists z(\neg P(z))) \Leftrightarrow$$

$$\exists x \forall y \exists z(R(x,y) \to (\neg P(z))).$$

Posljednja formula je u preneksnoj normalnoj formi i logički je ekvivalentna s formulom F.

Primjer 7.16. *Neka je sada $F \equiv \forall x R(x) \wedge \forall x P(x)$, gdje su R i P jednomjesni relacijski simboli. Primjenom pravila prijelaza 15 iz prve leme slijedi $F \Leftrightarrow \forall x(R(x) \wedge P(x))$, i to je jedna preneksna normalna forma za F.*

To možemo napraviti i drugačije. Primjenom leme o zamjeni vezanih varijabli imamo $F \Leftrightarrow \forall x R(x) \wedge \forall y P(y)$. Tada primjenom pravila prijelaza imamo $F \Leftrightarrow \forall x \forall y(R(x) \wedge P(y))$. Time smo dobili još jednu preneksnu normalnu formu formule F. Ovim primjerom željeli smo istaknuti da preneksna normalna forma dane formule nije jedinstvena.

Zadaci. Odredite preneksne normalne forme sljedećih formula

1. $(\forall z F(z) \ \vee \ (\exists y(\neg F(y)) \to \forall x G(x))) \to \forall w(F(w) \wedge G(w))$
2. $\forall x\Big(A(x) \to B(x)\Big) \to \forall y\Big(\exists x(A(y) \wedge C(y,x)) \to \exists x(B(y) \wedge C(x,x))\Big)$
3. $(\exists z F(z) \ \wedge \ (\exists y F(y) \to \forall x G(x))) \to \exists w(F(w) \wedge G(w))$

Rješenje zadataka 1.

$$(\forall z F(z) \ \vee \ (\exists y(\neg F(y)) \to \forall x G(x))) \to \forall w(F(w) \wedge G(w)) \ \Leftrightarrow$$

$$\forall z(F(z) \ \vee \ (\exists y(\neg F(y)) \to \forall x G(x))) \to \forall w(F(w) \wedge G(w)) \ \Leftrightarrow$$

$$\forall z(F(z) \ \vee \ \forall y(\neg F(y) \to \forall x G(x))) \to \forall w(F(w) \wedge G(w)) \ \Leftrightarrow$$

$$\forall z \forall y(F(z) \ \vee \ (\neg F(y) \to \forall x G(x))) \to \forall w(F(w) \wedge G(w)) \ \Leftrightarrow$$

$$\forall z \forall y(F(z) \ \vee \ \forall x(\neg F(y) \to G(x))) \to \forall w(F(w) \wedge G(w)) \ \Leftrightarrow$$

$$\forall z \forall y \forall x(F(z) \ \vee \ (\neg F(y) \to G(x))) \to \forall w(F(w) \wedge G(w)) \ \Leftrightarrow$$

$$\forall w \exists z \exists y \exists x[(F(z) \ \vee \ (\neg F(y) \to G(x))) \to (F(w) \wedge G(w))]$$

Rješenje zadatka 3.

$$(\exists z F(z) \ \wedge \ (\exists y F(y) \to \forall x G(x))) \to \exists w(F(w) \wedge G(w)) \Leftrightarrow$$

$$\exists z(F(z) \ \wedge \ (\exists y F(y) \to \forall x G(x))) \to \exists w(F(w) \wedge G(w)) \Leftrightarrow$$

$$\exists z(F(z) \ \wedge \ \forall x \forall y(F(y) \to G(x))) \to \exists w(F(w) \wedge G(w)) \Leftrightarrow$$

$$\exists z \forall x \forall y(F(z) \ \wedge \ (F(y) \to G(x))) \to \exists w(F(w) \wedge G(w)) \Leftrightarrow$$

$$\forall z \exists x \exists y \exists w[(F(z) \ \wedge \ (F(y) \to G(x))) \to (F(w) \wedge G(w))]$$

Poglavlje 8

Osmo predavanje - logika prvog reda

8.1 Glavni test za logiku prvog reda

8.1.1 Uvod

Sada razmatramo jedan postupak koji može poslužiti prilikom rješavanja sljedećih problema:

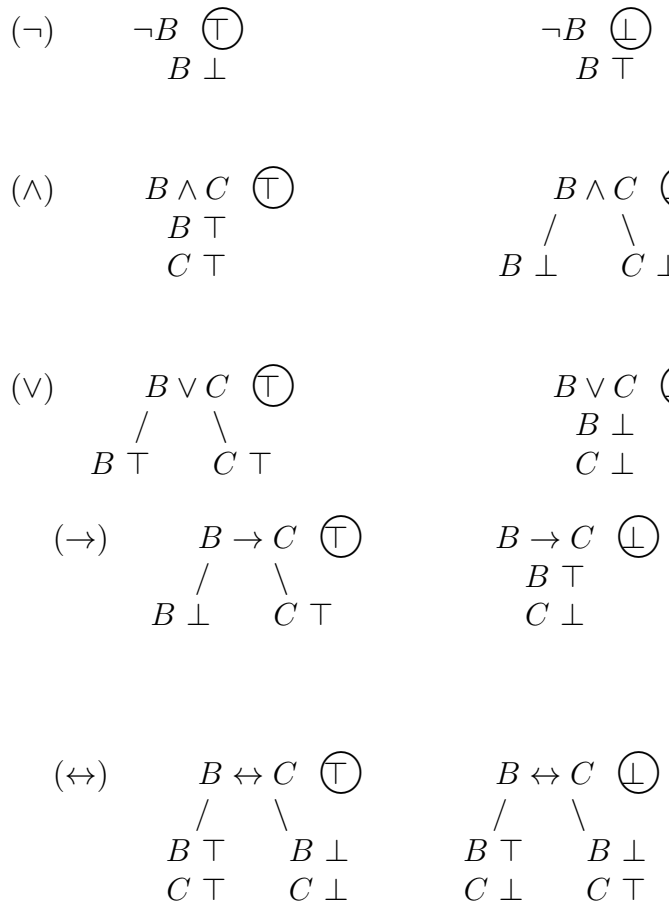
- Ispitivanje **valjanosti** formule;
- Ispitivanje **ispunjivosti** formule;
- Ispitivanje **oborivosti** formule;
- Određivanje je li neka formula **logička posljedica** zadanog konačnog skupa formula;
- Određivanje jesu li zadane dvije formule **logički ekvivalentne**.

Kod logike sudova glavni test uvijek završava u konačno mnogo koraka i korektno odgovaraju na postavljeno pitanje. Kod logike prvog reda to više nije slučaj. Postoji oboriva formula logike prvog reda za koju je svaka konačna struktura model. To znači da će ponekad biti nemoguće odrediti (beskonačnu) strukturu u konačno mnogo koraka.

U ovoj točki promatramo samo zatvorene formule koje ne sadrže konstantske i funkcijske simbole. Vidjet ćemo da je i taj smanjeni alfabet dovoljan kako bi se naglasila sva složenost problema ispitivanja valjanosti u logici prvog reda.

8.1.2 Pravila glavnog testa

Sada prvo navodimo pravila glavnog testa za propozicionalne veznike.



Preostalo je napisati pravila glavnog testa za kvantifikatore. No, opišimo prvo što zapravo znači ispitati valjanost neke formule F pomoću glavnog testa. Ako ispitujemo je li neka formula F valjana tada je prvi redak testa oblika $F\ \bot$. To znači da mi pokušavamo odrediti postoji li struktura koja nije model za formulu F. Iz definicije strukture slijedi da moramo odrediti nosač $|\mathfrak{M}|$ i preslikavanje φ. Opišimo prvo na koji način određujemo nosač $|\mathfrak{M}|$, tj. navedimo u kojim koracima ispitivanja **"punimo" nosač** s novim elementima. Postoje dva takva osnovna oblika. **To su:** $\forall x G(x)\ \bot$ **i** $\exists x G(x)\ \top$. Za svaki od ta dva navedena oblika moramo u $|\mathfrak{M}|$ dodati novi element, jer npr. istinitost formule $\exists x G(x)$ znači da postoji element u $|\mathfrak{M}|$ koji je "svjedok" istinitosti. Nakon analize retka oblika $\exists x G(x)\ \top$, prvo zaokružujemo znak $\top$, tj. pišemo $\textcircled{\top}$, te u taj redak dopisujemo $(..a..)$. Element a mora biti novi, tj. ne smije biti uveden u nekom prethodnom koraku. Oznaka $(..a..)$ nam sugerira da smo element a upravo uveli u tom koraku. Zatim, u (nekom) sljedećem retku pišemo $G(a)\ \top$. Analogno postupamo za retke oblika $\forall x G(x)\ \bot$. Time smo opisali dva oblika pravila za kvantifikatore. Važno je istaknuti da u ovoj točki znakove $a, b, c \ldots, a_1, a_2, \ldots$

smatramo elementima alfabeta, tj. to su konstantski simboli u jednu ruku. U drugu ruku te oznake koristimo za elemente nosača interpretacije. Sada opisujemo pravila za slučajeve $\forall xG(x)\ \top$ i $\exists xG(x)\ \bot$. Što zapravo znači istinitost formule oblika $\forall xG(x)$ za neku interpretaciju? Po definiciji to znači istinitost formule $G(x)$ za svaku valuaciju. Malo neprecizno, ali kraće zapisano, to zapravo znači istinitost formule $G(m)$, za svaki element $m \in |\mathfrak{M}|$. Dakle, za svaki uvedeni element a u nosaču mi moramo ispitati istinitost formule $G(a)$. Posebno to znači da ispitujemo za elemente koji su uvedeni prije i poslije retka $\forall xG(x)\ \top$. To pak povlači da redak oblika $\forall xG(x)\ \top$ možda nikad neće biti do kraja analiziran, jer se moguće poslije njega uvodi novi element u nosač. Analizu retka oblika $\forall xG(x)\ \top$ u odnosu na element a označavamo sa $\forall xG(x)\ \top\ @$. Zatim, u (nekom) sljedećem retku pišemo $G(a)\ \top$. Analogno postupamo za retke oblika $\exists xG(x)\ \bot$.

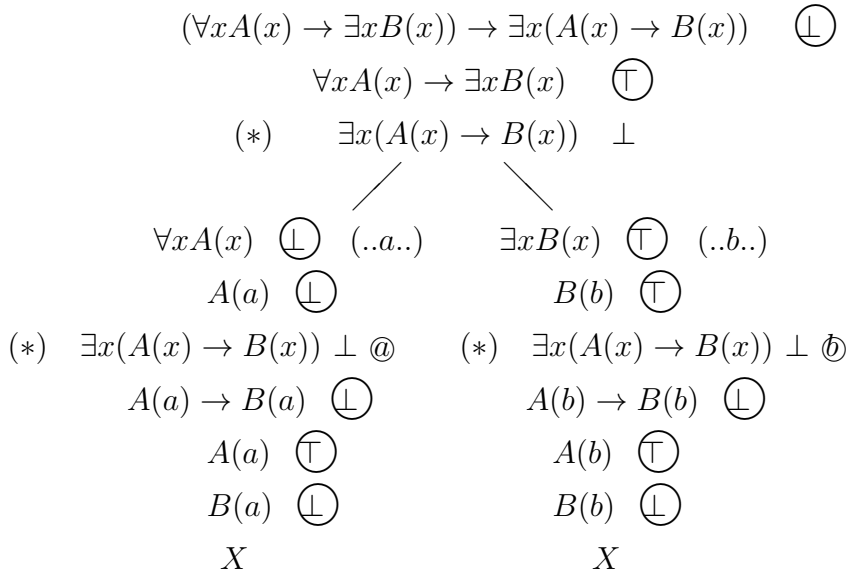
Sada navodimo pravila za kvantifikatore. Oznaka $(\uparrow\ a \downarrow\ a)$ bi nam trebala sugerirati da novo uvedeni element a moramo dopisati kod svih redaka oblika $\forall xG(x)\ \top$ i $\exists xG(x)\ \bot$ koji su bili prije i koji će se pojaviti kasnije.

$$(\forall) \qquad \frac{\forall xB\ \top\ @, \ldots}{B(a)\ \top} \qquad\qquad \frac{\forall xB\ \textcircled{\bot}\ (..a..)\ (\uparrow\ a \downarrow\ a)}{B(a)\ \bot}$$

$$(\exists) \quad \frac{\exists xB\ \textcircled{\top}\ (..a..)\ (\uparrow\ a \downarrow\ a)}{B(a)\ \top} \qquad\qquad \frac{\exists xB\ \bot\ @, \ldots}{B(a)\ \bot}$$

8.1.3 Primjeri

Primjer 8.1. *Ispitajmo valjanost formule* $(\forall xA(x) \to \exists xB(x)) \to \exists x(A(x) \to B(x))$ *Na sljedećoj slici dano je jedno stablo glavnog testa za zadanu formulu.*

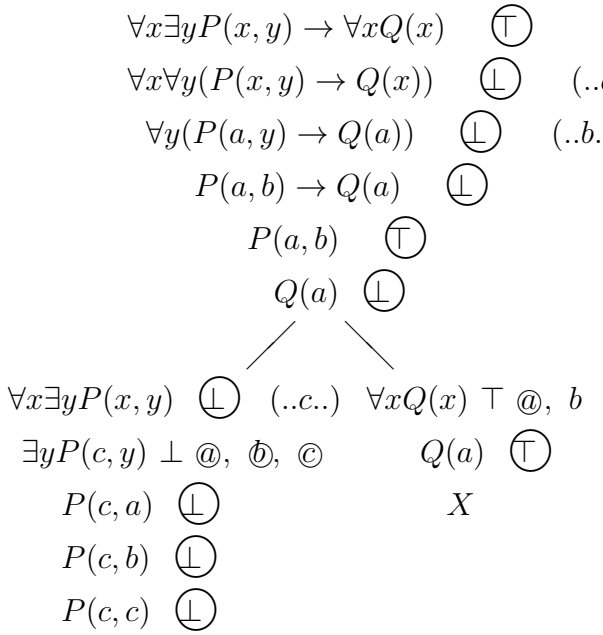


Sljedećim primjerom želimo istaknuti nužnost uvođenja novih elemenata u nosač prilikom analize formula oblika $\forall xG(x)\ \bot$ i $\exists xG(x)\ \top$.

Primjer 8.2. *Pomoću glavnog testa ispitajmo vrijedi li*

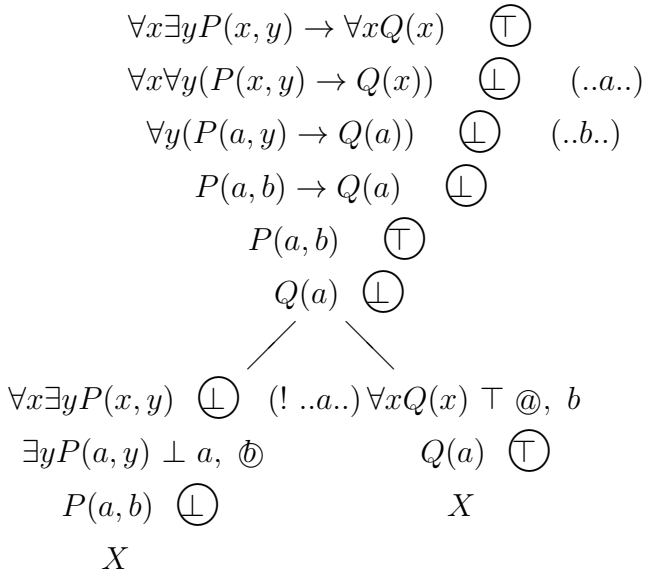
$$\forall x\exists yP(x,y) \to \forall xQ(x) \models \forall x\forall y(P(x,y) \to Q(x)).$$

Ispitivanje pomoću glavnog testa kratko je zapisano u obliku sljedećeg grafa.



Pošto lijeva grana nije završila oznakom za kontradikciju zaključujemo da dana tvrdnja nije istinita. S te lijeve grane možemo pročitati strukturu za koju početna tvrdnja nije istinita. Nosač strukture je $|\mathfrak{M}| = \{a, b, c\}$, *te je* $P^{\mathfrak{M}} = \{(a, b)\}$ *i* $Q^{\mathfrak{M}} = \emptyset$. *Uočimo još da na desnoj grani prvi redak oblika* $\forall x Q(x) \top$ *nismo analizirali u odnosu na element b. To nije nužno jer smo već naišli na kontradikciju.*

Pogledajmo sada što se događa kada u rješavanju gornjeg zadatka koristimo "stare" elemente.



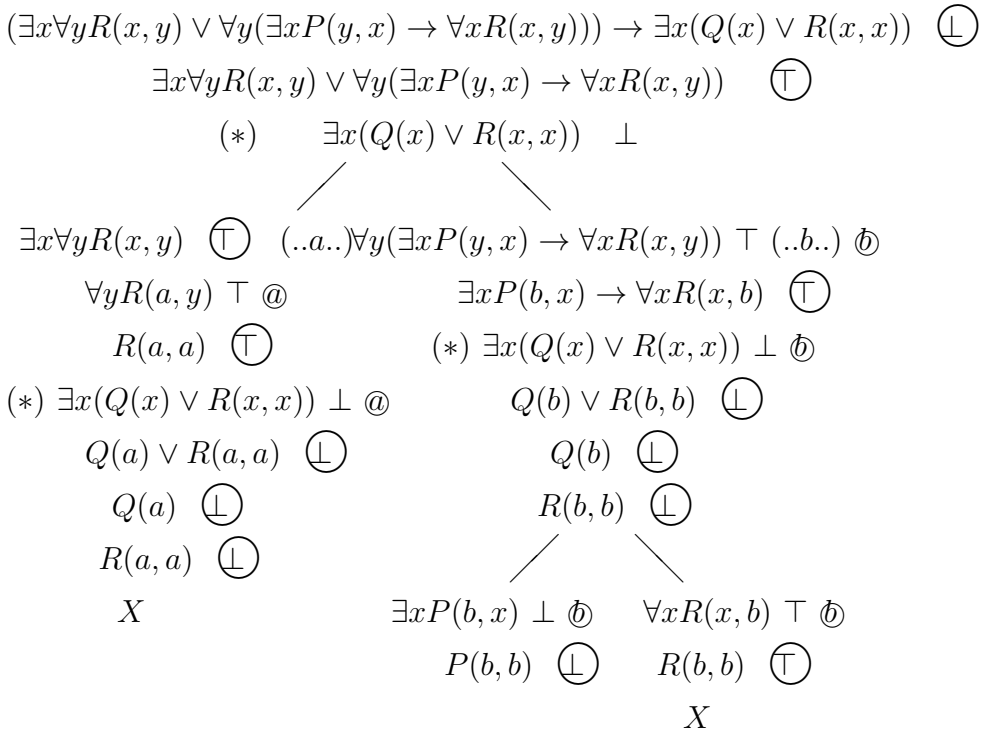
U sedmom retku smo sa (! ..*a*..) označili da ne uvodimo novi element već koristimo stari. Dani test je na svim granama završio kontradikcijom, pa bi brzopleto (i krivo) mogli zaključiti da je dana formula valjana. Iz prethodnog testa znamo da formula nije valjana. Korištenjem "starog" elementa a mi smo posljednjim testom zapravo dokazali da ne postoji strukura s točno dva elementa koja nije model za F.

U sljedećem primjeru želimo istaknuti kako se glavni test koristi za ispitivanje je li neka formula F **oboriva**. Početni redak u testu je oblika $F\bot$. To znači da pokušavamo odrediti strukturu koja nije model za formulu F.

Primjer 8.3. *Ispitajmo pomoću glavnog testa je li sljedeća formula oboriva:*

$$(\exists x \forall y R(x, y) \vee \forall y(\exists x P(y, x) \to \forall x R(x, y))) \to \exists x(Q(x) \vee R(x, x)).$$

Rješenje.



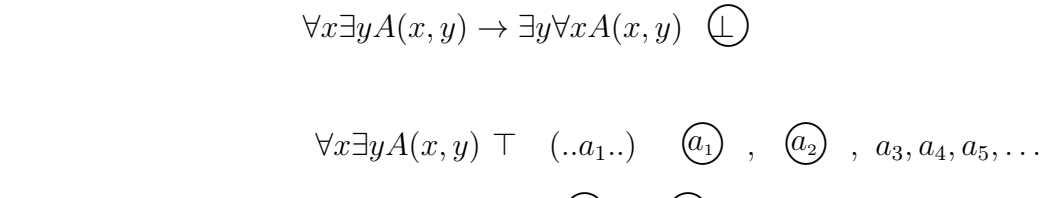
Zadana formula je oboriva. Struktura $\mathfrak{M}$ koja to dokazuje je zadana sa: $\mathfrak{M} = \{b\}$, te $Q^{\mathfrak{M}} = R^{\mathfrak{M}} = P^{\mathfrak{M}} = \emptyset$.

Svakako treba istaknuti i način uvođenja elementa b u prvom retku na desnoj grani. Primijetite da je taj redak oblika $\forall xG(x)$ ⊤. Iz pravila za kvantifikatore znamo da to nije oblik kod kojeg se uvodi novi element. No, ni u sljedećim recima nema oblika s kvantifikatorima kod kojih se uvodi novi element. Radi provođenja daljnje analize bili smo prinuđeni u tom prvom retku na desnoj grani uvesti novi element.

8.1.4 Neodlučivost logike prvog reda

Sljedeći primjer pokazuje da nekad test ne mora završiti, ali mi ipak možemo odrediti jednu traženu (beskonačnu) strukturu.

Primjer 8.4. *Ispitajmo je li formula* $\forall x\exists yA(x,y) \to \exists y\forall xA(x,y)$ *valjana.*



$$\begin{array}{lll}
\exists y A(a_1, y) & \textcircled{\top} & (..a_2..) \\
\forall x A(x, a_1) & \textcircled{\bot} & (..a_3..) \\
A(a_1, a_2) & \textcircled{\top} & \\
A(a_3, a_1) & \textcircled{\bot} & \\
\exists y A(a_2, y) & \textcircled{\top} & (..a_4..) \\
\forall x A(x, a_2) & \textcircled{\bot} & (..a_5..) \\
A(a_2, a_4) & \textcircled{\top} & \\
A(a_5, a_2) & \textcircled{\bot} & \\
\vdots & &
\end{array}$$

Neka je $|\mathfrak{M}| = \{a_n : n \in \mathbb{N} \setminus \{0\}\}$, *te* $A^{\mathfrak{M}} = \{(a_n, a_{2n}) : n \in \mathbb{N}\}$ *Nije teško vidjeti da vrijedi* $\mathfrak{M} \not\models \forall x \exists y A(x, y) \to \exists y \forall x A(x, y)$. *To znači da dana formula nije valjana.*

Prethodni primjer je prije svega važan kako bi istaknuli da glavni test ne mora uopće završti. Istaknimo samo da je moguće konstruirati konačnu strukturu koja nije model za danu formulu.

Sljedeća formula je primjer formule za koju je svaka konačna struktura model, ali ona ipak nije valjana:

$$\forall x_1 \forall x_2 \forall x_3 (R(x_1, x_1) \wedge (R(x_1, x_3) \to (R(x_1, x_2) \vee R(x_2, x_3)))) \to$$

$$\exists y \forall z R(y, z).$$

Pokušajte to dokazati pomoću glavnog testa. Analiza dosta brzo postaje jako složena. Nije uopće jasna strategija kojom bismo konstruirali beskonačnu strukturu.

Pokušajte zatim pomoću glavnog testa ispitati slijedi li logički formula

$$\forall x \forall y \exists z (R(x, y) \to (R(x, z) \to R(z, y))$$

iz skupa formula

$$\{\forall x \exists y R(x, y),\ \forall x \forall y (R(x, y) \to \neg R(y, x)\}.$$

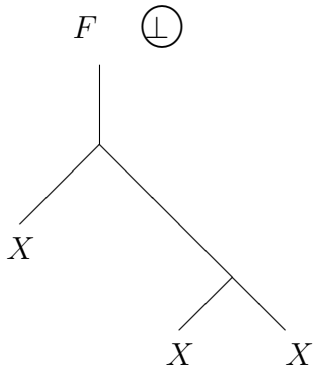
Analiza vrlo brzo postaje jako složena, te nije jasno hoće li glavni test uopće završiti. Naravno, vrlo lako je napisati još kompliciranije formule za koje će ispitivanje valjanosti biti jako složeno. No, to nije slučaj samo s glavnim testom, već se isti problemi javljaju kod svakog testa za ispitivanje valjanosti formula logike prvog reda. To ističemo u sljedećom teoremu Alonsa Churcha.

Teorem 8.5. (Churchov teorem) *Logika prvog reda je neodlučiva, tj. ne postoji test kojim bi se za svaku formulu u konačno mnogo koraka mogli ispitati je li valjana.*

Za dokaz prethodnog teorema morali bismo prvo uvesti osnovne pojmove i rezultate teorije izračunljivosti. O tome ćemo govoriti kasnije, tj. u trećem ciklusu ovih predvanja.

Rezimirajmo na kraju kako sve glavni test može završiti prilikom ispitivanja valjanosti neke formule. Moguće su sljedeće dvije situacije:

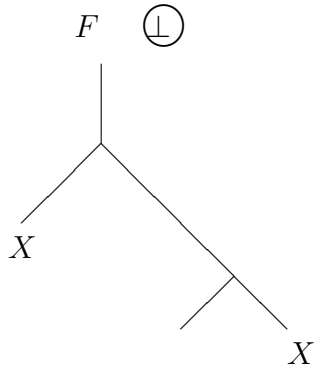
a) Test je završio u konačno mnogo koraka i sve grane su završile kontradikcijom. Jednu situaciju iz slučaja a) prikazujemo sljedećom slikom.



Tada zaključujemo da je dana formula F valjana.

b) Postoji grana koja nije završila kontradikcijom. Tu razlikujemo sljedeća dva podslučaja.

b_1) Postoji grana koja nije završila kontradikcijom gdje je test proveden do kraja. Na sljedećoj slici je prikazana jedna takva situacija.



Tada zaključujemo da je dana formula F oboriva. S grane koja nije završila kontradikcijom čitamo strukturu koja nije model za danu formulu.

b_2) Test nije proveden do kraja i nije jasno hoće li završiti u konačno koraka. U nekim specijalnim slučajevima moguće je na osnovu periodičkog ponavljanja odrediti beskonačnu strukturu koja nije model za danu formulu. No, većinom u takvim slučajevima ne možemo ništa zaključiti.

8.1.5 Zadaci

1. Odredite preneksnu normalnu formu formule i ispitajte valjanost pomoću glavnog testa

$$(\exists z F(z) \ \wedge \ (\exists y F(y) \to \forall x G(x))) \to \exists w (F(w) \wedge G(w))$$

Rješenje. Određujemo prvo preneksnu normalnu formu dane formule:

$$(\exists z F(z) \ \wedge \ (\exists y F(y) \to \forall x G(x))) \to \exists w (F(w) \wedge G(w)) \Leftrightarrow$$

$$\exists z (F(z) \ \wedge \ (\exists y F(y) \to \forall x G(x))) \to \exists w (F(w) \wedge G(w)) \Leftrightarrow$$

$$\exists z (F(z) \ \wedge \ \forall x \forall y (F(y) \to G(x))) \to \exists w (F(w) \wedge G(w)) \Leftrightarrow$$

$$\exists z \forall x \forall y (F(z) \ \wedge \ (F(y) \to G(x))) \to \exists w (F(w) \wedge G(w)) \Leftrightarrow$$

$$\forall z \exists x \exists y \exists w [(F(z) \ \wedge \ (F(y) \to G(x))) \to (F(w) \wedge G(w))]$$

Na posljednju formulu primjenjujemo glavni test:

$$\forall z\exists x\exists y\exists w[(F(z) \ \wedge \ (F(y) \to G(x))) \to (F(w) \wedge G(w))] \quad \textcircled{\perp} \quad (..a..)$$
$$\exists x\exists y\exists w[(F(a) \ \wedge \ (F(y) \to G(x))) \to (F(w) \wedge G(w))] \quad \perp @$$
$$\exists y\exists w[(F(a) \ \wedge \ (F(y) \to G(a))) \to (F(w) \wedge G(w))] \quad \perp @$$
$$\exists w[(F(a) \ \wedge \ (F(a) \to G(a))) \to (F(w) \wedge G(w))] \quad \perp @$$
$$(F(a) \ \wedge \ (F(a) \to G(a))) \to (F(a) \wedge G(a)) \quad \textcircled{\perp}$$
$$F(a) \ \wedge \ (F(a) \to G(a)) \quad \textcircled{\top}$$
$$F(a) \wedge G(a) \quad \textcircled{\perp}$$
$$F(a) \quad \textcircled{\top}$$
$$F(a) \to G(a) \quad \textcircled{\top}$$

$F(a)$ $\textcircled{\perp}$ — X ; $G(a)$ $\textcircled{\top}$

$F(a)$ $\textcircled{\perp}$ — X ; $G(a)$ $\textcircled{\perp}$ — X

Sve grane su završile kontradikcijom pa zaključujemo da je dana formula valjana.

2. Pomoću glavnog testa ispitajte valjanost sljedećih formula:

   a) $(B \ \to \ (\forall xA(x) \wedge \forall xC(x))) \ \to \ (\neg B \vee \forall x(A(x) \wedge C(x)))$, pri čemu je B zatvorena formula;

   b) $\forall x\forall yP(x,y) \to (\exists y\exists xP(y,x) \ \vee \ \exists x\exists yP(x,y))$;

   c) $(\neg A \wedge (\exists xB(x) \vee \exists xC(x))) \ \leftrightarrow \ \neg(A \vee \forall x(\neg B(x) \wedge \neg C(x)))$;

   d) $\forall x\forall y(P(x,y) \wedge Q(x)) \to (\forall x\forall yP(x,y) \wedge \forall xQ(x))$.

   Rješenje: Sve navedene formule su valjane.

3. Odredite prvo preneksnu normalnu formu formule

$$(\forall xF(x) \vee (\exists xF(x) \to \forall xG(x))) \to (\forall xF(x) \wedge \forall xG(x)),$$

   a zatim ispitajte je li dobivena preneksna normalna forma oboriva.

4. Pomoću glavnog testa ispitajte vrijedi li:

   a) $\{\forall x\forall y(F(x,y) \to \neg F(y,x))\} \models \neg\exists xF(x,x)$;

   b) $\{\forall x(A(x) \to B(x))\} \models \forall y(\exists x(A(x) \wedge C(y,x)) \to \exists x(B(x) \wedge C(y,x)))$;

c) $\{\exists x\forall y B(x,y) \rightarrow A,\ \neg A \vee \exists x\exists y B(x,y)\} \models A \vee \neg\forall y\forall x B(y,x)$, gdje je formula A zatvorena;

d) $\forall x\forall y(R(x,y) \rightarrow \neg R(y,x)) \models \forall x\forall y(R(x,y) \rightarrow (R(y,x) \rightarrow R(x,x)))$;

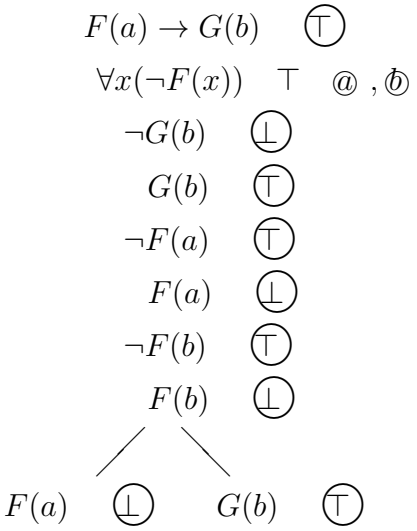
e) $F \rightarrow \neg\forall y\exists x R(x,y),\ \forall x\exists y R(y,x) \vee \exists x R(x,x) \models \neg F \vee \forall x\forall y R(x,y)$;

f) $\exists x(R(x,x) \rightarrow \forall y R(x,y)) \models \forall x\forall y(\neg R(x,y) \rightarrow R(y,x))$. Ako tvrdnja ne vrijedi odredite barem jednu interpretaciju koja to dokazuje.

5. Ispitajte pomoću glavnog testa vrijedi li

$$F(a) \rightarrow G(b),\ \forall x(\neg F(x)) \models \neg G(b).$$

Rješenje: Pošto dana formula sadrži konstantske simbole a i b moramo prvo reći što raditi s njima prilikom glavnog testa. Po definiciji strukture za svaki konstantski simbol mora postojati element u nosaču. To znači da prije početka testiranja smatramo da nosač strukture sadrži barem dva elementa. Interpretacije konstatskih simbola, kao i obično u ovoj točki, označavamo istim znakovima. Glavni test zapisujemo u obliku stabla ovako:



Pošto sve grane nisu završile kontradikcijom zaključujemo da dana tvrdnja nije istinita, tj. formula $\neg G(b)$ logički ne slijedi iz skupa formula $\{F(a) \rightarrow G(b),\ \forall x(\neg F(x))\}$.

6. Pomoću glavnog testa ispitajte je li ispunjiva formula

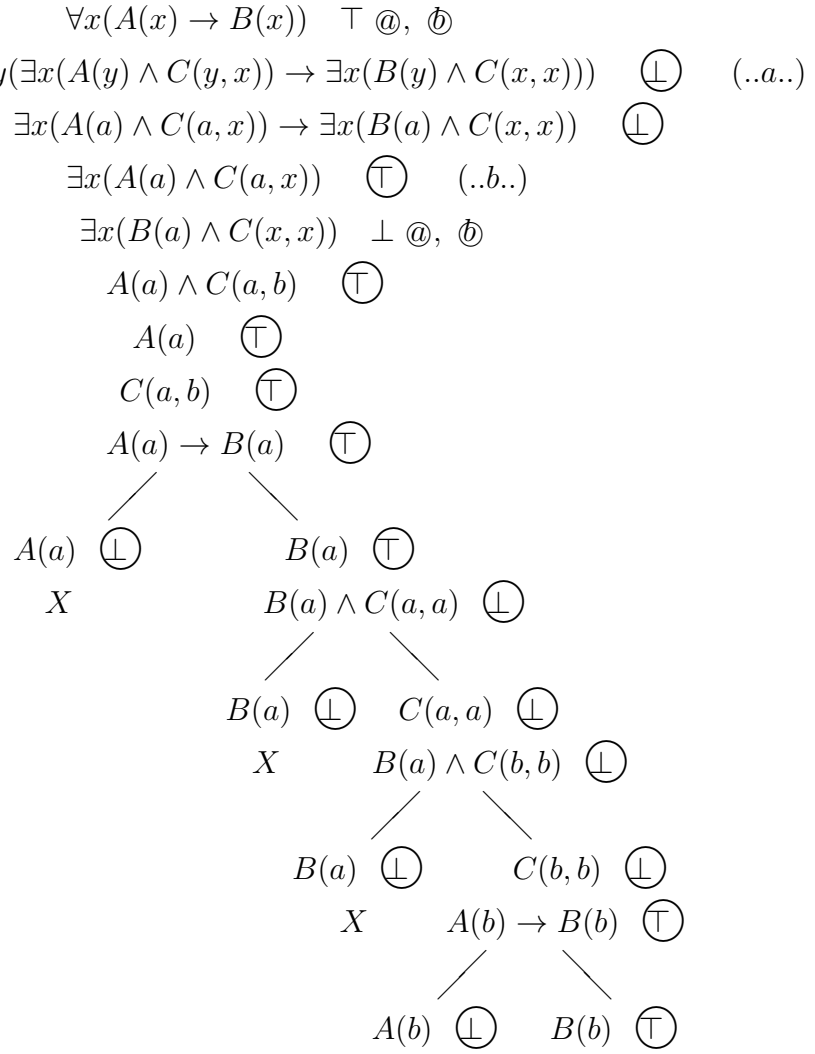
$$(\forall x\exists y P(x,y) \wedge \forall x Q(x)) \wedge \neg\forall x\exists y(P(x,y) \wedge Q(x)).$$

7. Neka je A zatvorena formula, a B formula s točno jednom slobodnom varijablom. Koristeći glavni test dokažite ili opovrgnite

$$\{\exists x B(x) \to A,\ \neg A \vee \exists x B(x)\} \models A \vee \neg \exists x B(x).$$

8. Pomoću glavnog testa odredite barem dvije strukture koje dokazuju

$$\forall x(A(x) \to B(x)) \not\models \forall y(\exists x(A(y) \wedge C(y,x)) \to \exists x(B(y) \wedge C(x,x))).$$



Neka je $|\mathfrak{M}| = \{a,b\}$, te $A^{\mathfrak{M}} = \{a\}$, $B^{\mathfrak{M}} = \{a\}$, i $C^{\mathfrak{M}} = \{(a,b)\}$. Zatim definiramo $|\mathfrak{N}| = \{a,b\}$, $A^{\mathfrak{N}} = \{a\}$, $B^{\mathfrak{N}} = \{a,b\}$ i $C^{\mathfrak{N}} = \{(a,b)\}$. Tada su $\mathfrak{M}$ i $\mathfrak{N}$ tražene dvije strukture.

9. Prilikom ispitivanje valjanosti formule

$$\forall x \exists y A(x,y) \to \exists y \forall x A(x,y)$$

proveden je sljedeći glavni test:

$$\begin{array}{rl}
\forall x \exists y A(x,y) \to \exists y \forall x A(x,y) & \textcircled{\bot} \\
\forall x \exists y A(x,y) \ \top & (..a_1..) \quad \textcircled{a_1} \ , \ \textcircled{a_2} \\
\exists y \forall x A(x,y) \ \bot & \textcircled{a_1} \ , \ \textcircled{a_2} \\
\exists y A(a_1,y) & \textcircled{\top} \quad (..a_2..) \\
A(a_1,a_2) & \textcircled{\top} \\
\forall x A(x,a_1) & \textcircled{\bot} \quad (!..a_1..) \\
A(a_1,a_1) & \textcircled{\bot} \\
\exists y A(a_2,y) & \textcircled{\top} \quad (!..a_1..) \\
A(a_2,a_1) & \textcircled{\top} \\
\forall x A(x,a_2) & \textcircled{\bot} \quad (!..a_2..) \\
A(a_2,a_2) & \textcircled{\bot}
\end{array}$$

Uočite da je test završen, tj. provedena je analiza za sve formule i sve elemente. Pošto test nije završen kontradikcijom možemo li zaključiti da dana formula nije valjana? (Uočite da smo u primjeru 8.4. također ispitivali valjanost iste formule. U gornjem testu smo s znakom ! označili da upotrebljavamo stari element, iako bismo po pravilu trebali uvoditi novi element.)

10. Neka je F zatvorena formula, a R dvomjesni relacijski simbol. Koristeći glavni test dokažite ili opovrgnite

$$\{\neg F \vee \exists x \forall y(\neg R(x,y)),\ \forall y \exists x R(y,x) \vee \exists x R(x,x)\} \models F \to \exists x \exists y R(x,y).$$

Rješenje:

$\neg F \vee \exists x \forall y(\neg R(x,y))$ $\top$
$\forall y \exists x R(y,x) \vee \exists x R(x,x)$ $\top$
$F \to \exists x \exists y R(x,y)$ $\bot$
F $\top$
(I) $\exists x \exists y R(x,y)$ $\bot$ @

Lijevo: $\neg F$ $\top$; F $\bot$; X

Desno: $\exists x \forall y(\neg R(x,y))$ $\top$ $(..a..)$
(II) $\forall y(\neg R(a,y))$ $\top$ @
$\neg R(a,a)$ $\top$
$R(a,a)$ $\bot$
(III) $\exists y R(a,y)$ $\bot$ @
$R(a,a)$ $\bot$

Lijevo: $\forall y \exists x R(y,x)$ $\top$@ , c
(I) $\exists x \exists y R(x,y)$ $\bot$ c
(II) $\forall y(\neg R(a,y))$ $\top$ c
(III) $\exists y R(a,y)$ $\bot$ ©
$\exists x R(a,x)$ $\top$ $(..c..)$
$R(a,c)$ $\top$
$R(a,c)$ $\bot$
X

Desno: $\exists x R(x,x)$ $\top$ $(..b..)$
(I) $\exists x \exists y R(x,y)$ $\bot$ ƀ
(II) $\forall y(\neg R(a,y))$ $\top$ b
(III) $\exists y R(a,y)$ $\bot$ b
$R(b,b)$ $\top$
$\exists y R(b,y)$ $\bot$ ƀ
$R(b,b)$ $\bot$
X

Pošto su sve grane završile kontradikcijom zaključujemo da je početna tvrdnja istinita.

Poglavlje 9

Deveto predavanje - logika prvog reda

9.1 Račun teorija prvog reda

Sadržaj predavanja:

- prvo dajemo definiciju jednog hilbertovskog sistema za logiku prvog reda, kojeg ovdje označavamo s RP (račun predikata).
- Nakon toga definiramo pojmove dokaza, teorema i izvoda.
- Prvo dokazujemo teorem adekvatnosti za sistem RP, kojim je iskazana korektnost sistema obzirom na semantiku definiranu u prethodnoj točki.
- Zatim navodimo osnovna svojstva izvoda, odnosno dokaza.
- Te sve činjenice služe nam kasnije za dokaz najvažnijeg teorema o teorijama prvog reda – generaliziranog teorema potpunosti.

9.1.1 Osnovne definicije

Važno je naglasiti da u ovoj točki ne promatramo isti skup logičkih simbola kao prije, već samo skup $\{\neg, \rightarrow, \forall\}$. Logičke simbole koji ne pripadaju alfabetu ipak ćemo koristiti prilikom zapisivanja nekih formula. No, ti simboli su samo pokrate za neke duže formule. Sada to točno definiramo:

$A \wedge B$ označava $\neg(A \rightarrow \neg B)$;
$A \vee B$ označava $\neg A \rightarrow B$;
$A \leftrightarrow B$ označava $\neg((A \rightarrow B) \rightarrow \neg(B \rightarrow A))$;
$\exists x A$ označava $\neg\forall x(\neg A)$.

Definicija 9.1. Račun logike prvog reda *zadan je s pet shema aksioma i dva pravila izvoda. Sheme aksioma su sljedeće:*

(A1) $A \to (B \to A)$;

(A2) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$;

(A3) $(\neg B \to \neg A) \to (A \to B)$;

(A4) $\forall x A(x) \to A(t/x)$, *gdje je term* t *slobodan za varijablu* x *u formuli* A;

(A5) $\forall x(A \to B) \to (A \to \forall x B)$, *gdje formula* A *ne sadrži slobodnih nastupa varijable* x.

Pravila izvoda su modus ponens i generalizacija, tj.

$$\frac{A \qquad A \to B}{B} \qquad i \qquad \frac{A}{\forall x A}.$$

Ovako definirani sistem kratko ćemo označavati s RP (skraćenica od "račun predikata").

Definicija 9.2. Teorija T **prvog reda** *definirana je svojim jezikom, skupom aksioma i pravilima izvoda. Smatramo da je definiran jezik teorije* T *ako smo definirali skup nelogičkih simbola. Po definiciji smatramo da svaka teorija* T *sadrži sve sheme aksioma sistema RP. Zatim, jedina pravila izvoda teorije prvog reda su modus ponens i generalizacija. Aksiomi teorije* T *koji nisu valjane formule nazivamo* **nelogički aksiomi. Smatramo da je zadan skup aksioma teorije** T **ako je zadan skup nelogičkih aksioma**.

Primjer 9.3. *Kao primjer teorije prvog reda definiramo* **teoriju parcijalno uređenih skupova**. *Signatura:* $\sigma = \{=, \leq\}$ *Nelogički aksiomi:*

aksiomi za jednakost
$\forall x(x \leq x)$
$\forall x \forall y(x \leq y \ \wedge y \leq x \ \to \ x = y)$
$\forall x \forall y \forall z(x \leq y \ \wedge \ y \leq z \ \to \ x \leq z)$

Definicija 9.4. *Neka je zadana neka teorija* T *prvog reda. Zatim, neka su* $A_1, \ldots, A_n$ *i* A *formule jezika teorije* T. *Kažemo da je niz formula* $A_1, \ldots, A_n$ **dokaz** *za formulu* A *u teoriji* T *ako vrijedi:*

a) formula A_n *je upravo* A;

b) za sve $i \in \{1, \ldots, n\}$ *vrijedi jedno od:*

- *formula* A_i *je aksiom od* T;

– *formula A_i je nastala primjenom pravila izvoda modus ponens ili generalizacije na neke formule A_j i A_k, pri čemu je $j, k < i$.*

Kažemo da je formula A **teorem sistema** *T ako u T postoji dokaz za A. To označavamo sa $\vdash_T A$.*

Obično ćemo u ovoj točki kratko pisati $\vdash A$ umjesto $\vdash_{RP} A$. Sa $\nvdash A$ ćemo označavati činjenicu da formula A nije teorem logike prvog reda, a $\nvdash_T A$ označava da formula A nije teorem neke teorije T prvog reda.

9.1.2 Adekvatnost i teorem dedukcije

Teorem 9.5. (Teorem adekvatnosti za sistem RP)
Svaki teorem sistema RP je valjana formula.

Dokaz je lako provesti indukcijom po duljini dokaza.

Teorem 9.6. (Teorem adekvatnosti za teoriju prvog reda)
Neka je T teorija prvog reda i F neka formula jezika teorije T. Ako vrijedi $\vdash_T F$ tada za sve modele $\mathfrak{M}$ teorije T vrijedi $\mathfrak{M} \models F$.

Definicija 9.7. *Neka je T neka teorija prvog reda. Zatim, neka je Γ skup formula jezika teorije T, te A formula istog jezika. Kažemo da je niz $A_1, \ldots, A_n$ formula teorije T* **izvod** *iz skupa Γ formule A u teoriji T, u oznaci $\Gamma \vdash_T A$, ako vrijedi:*

a) formula A_n je upravo formula A;

b) za sve $i \in \{1, \ldots, n\}$ vrijedi barem jedno od sljedećeg:

- b_1) *A_i je aksiom teorije T;*
- b_2) *$A_i \in \Gamma$;*
- b_3) *formula A_i je nastala iz nekih A_k, A_j $(k, j < i)$ pomoću pravila izvoda modus ponens ili generalizacije.*

Teorem 9.8. (Teorem dedukcije za teorije prvog reda)
Neka je Γ skup formula teorije T, A zatvorena, a B proizvoljna formula. Ako vrijedi $\Gamma \cup \{A\} \vdash_T B$ tada vrijedi i $\Gamma \vdash_T A \to B$.

Dokaz je sasvim analogan dokazu teorema dedukcije za račun sudova.

Napomena 9.9. *Sada teorem dedukcije više ne vrijedi u istom obliku kao što je bio izrečen i dokazan za račun sudova. Točnije, za proizvoljnu formulu A, za koju vrijedi $\Gamma \cup \{A\} \vdash B$, ne mora vrijediti $\Gamma \vdash A \to B$.*

9.1.3 Potpunost

Sada nam je glavni cilj dokazati **teorem potpunosti**, tj. da je svaka valjana formula teorem sistema *RP*. Prvo ćemo navesti jači teorem – **generalizirani teorem potpunosti**. Nakon toga će teorem potpunosti slijediti kao jednostavan korolar. Prije samih dokaza navedenih teorema dajemo definiciju **konzistentne teorije** prvog reda, te navodimo osnovna svojstva vezana uz konzistentnost.

Definicija 9.10. *Neka je T proizvoljna teorija prvog reda, te σ pripadna signatura. Kažemo da je teorija T* **konzistentna** *ako ne postoji σ–formula F tako da su F i $\neg F$ teoremi teorije T. Inače kažemo da je teorija T* **inkonzistentna**. *Za skup σ–formula Γ kažemo da je* **konzistentan** *u teoriji T ako ne postoji σ–formula F tako da vrijedi $\Gamma \vdash_T F$ i $\Gamma \vdash_T \neg F$. Inače kažemo da je skup formula Γ* **inkonzistentan** *u teoriji T.*

Teorem 9.11. *Teorija RP je konzistentna.*

Dokaz. Pretpostavimo da su za neku formulu F istovremeno F i $\neg F$ teoremi logike prvog reda. Iz teorema adekvatnosti tada slijedi da su formule F i $\neg F$ valjane, što je nemoguće. Q.E.D.

U sljedećoj propoziciji navodimo svojstva konzistentnih skupova. Dokazi svih tvrdnji su sasvim analogni dokazima u logici sudova.

Propozicija 9.12. *Neka je T teorija prvog reda, σ pripadna signatura, i Γ skup σ–formula. Tada vrijede sljedeće tvrdnje:*

a) Skup Γ je konzistentan u teoriji T ako i samo je svaki konačan podskup od Γ konzistentan u teoriji T;

b) Skup formula Γ je konzistentan u teoriji T ako i samo ako postoji σ–formula F tako da vrijedi $\Gamma \nvdash_T F$;

c) Ako je F zatvorena σ–formula i vrijedi $\Gamma \nvdash_T F$, tada je skup $\Gamma \cup \{\neg F\}$ konzistentan u teoriji T;

d) Ako je F zatvorena σ–formula i vrijedi $\Gamma \nvdash_T \neg F$, tada je skup formula $\Gamma \cup \{F\}$ konzistentan u teoriji T;

e) Ako postoji model teorijie T koji je model i za skup formula Γ tada je skup Γ konzistentan u teoriji T;

f) Ako je Γ konzistentan skup formula u teoriji T i F zatvorena σ–formula, tada je bar jedan od skupova $\Gamma \cup \{F\}$ i $\Gamma \cup \{\neg F\}$ konzistentan u teoriji T;

g) Ako je Γ konzistentan skup formula u teoriji T, te je F σ–formula takva da vrijedi $\Gamma \vdash F$, tada je i skup $\Gamma \cup \{F\}$ konzistentan u teoriji T.

Prilikom dokaza potpunosti u logici sudova nakon razmatranja konzistentnih skupova, ključne su bile Lindenbaumova lema i lema o istinitosti. Slična je situacija i u logici prvog reda. No, osim tih dviju lema treba razmatrati i tzv. Henkinove teorije. Sve detalje dokaza potpunosti za teorije prvog reda možete vidjeti u knjizi M. Vuković, Matematička logika, Element, Zagreb, 2009. Ovdje samo iskazujemo generalizirani teorem potpunosti, te navodimo njegove najvažnije posljedice.

Teorem 9.13. (Generalizirani teorem potpunosti za teorije prvog reda)
Za svaku konzistentnu teoriju T prvog reda postoji prebrojiv model.

Korolar 9.14. *Neka je T teorija prvog reda i F formula pripadnog jezika. Ako je formula F istinita u svakom modelu za T tada je F teorem od T.*

Dokaz. Dovoljno je dokazati tvrdnju za zatvorene formule, jer očito vrijedi da je neka struktura $\mathfrak{M}$ model za formulu A ako i samo ako je $\mathfrak{M}$ model za formulu $\overline{A}$. Zatim, znamo (!) da vrijedi $\vdash_T A$ ako i samo ako $\vdash_T \overline{A}$, za sve formule A. Tvrdnju korolara dokazujemo obratom po kontrapoziciji. Neka je F zatvorena formula koja nije teorem teorije T. Iz propozicije 9.12. slijedi da je tada teorija $T + \{\neg F\}$ konzistentna. Iz generaliziranog teorema potpunosti slijedi da postoji model $\mathfrak{M}$ za teoriju $T + \{\neg F\}$. To znači da formula F nije istinita u modelu $\mathfrak{M}$ za teoriju T. Q.E.D.

Neposredna posljedica prethodnog korolara je sljedeći Gödelov teorem potpunosti.

Teorem 9.15. (Gödelov teorem potpunosti)
Neka je T teorija prvog reda i F formula pripadnog jezika. Tada vrijedi:

$$\vdash_T F \text{ ako i samo ako } \text{ za sve modele } \mathfrak{M} \text{ od } T \text{ vrijedi } \mathfrak{M} \models F$$

Posebno vrijedi: $\vdash_{RP} F$ ako i samo ako je F valjana formula.

Teorem 9.16. (Jaki teorem potpunosti za sistem RP)
Neka je S skup formula logike prvog reda, a F neka formula. Vrijedi:

$$S \models F \quad \text{ako i samo ako} \quad S \vdash_{RP} F.$$

Dokaz. Ako vrijedi $S \vdash_{RP} F$ tada indukcijom po duljini izvoda $F_1, \ldots, F_n$ lako dokazujemo $S \models F_i$, za sve i. Tada posebno slijedi $S \models F_n$, tj. $S \models F$. Pretpostavimo sada da imamo $S \nvdash_{RP} F$. Bez smanjenja općenitosti možemo pretpostaviti da je F zatvorena formula. Tada iz propozicije 9.12. slijedi da je skup formula $S \cup \{\neg F\}$ konzistentan. Iz generaliziranog teorema potpunosti slijedi da postoji model $\mathfrak{M}$ za taj skup formula. Tada imamo $\mathfrak{M} \models S$ i $\mathfrak{M} \models \neg F$, tj. $\mathfrak{M} \not\models F$. To znači da $S \not\models F$. Q.E.D.

Teorem 9.17. (Teorem kompaktnosti)
Neka je S neki skup σ–formula. Tada vrijede sljedeće tvrdnje:

a) Za skup formula S postoji model ako i samo ako za svaki konačan podskup od S postoji model.

b) $S \models F$ ako i samo postoji konačan podskup S' od S tako da vrijedi $S' \models F$.

Teorem 9.18. (Löwenheim–Skolemov teorem "na dolje")
Svaka teorija prvog reda koja ima model ima i prebrojiv model.

Poglavlje 10

Deseto predavanje – izračunljivost

10.1 Uvod

Algoritam je jedan od osnovnih pojmova matematike. Prilikom proučavanja matematičke logike naveli smo sljedeći teorem.

> **Churchov teorem**. *Logika prvog reda je* **neodlučiva teorija**, *tj. ne postoji algoritam koji bi za svaku formulu u konačno mnogo koraka određivao je li dana formula valjana.*

Da bi tako nešto tvrdili moramo prvo točno definirati pojam algoritma kako bi mogli dokazati da za problem ispitivanja valjanosti algoritam ne postoji. U nastavku predavanja bavit ćemo se problem definicije izračunljive funkcije $f : S \to \mathbb{N}$, gdje je $S \subseteq \mathbb{N}^k$. Više detalja, te sve ovdje ispuštene dokaze možete pogledati u nastavnom materijalu posvećenom kolegiju *Izračunljivost* koji sam niz godina predavao na PMF–u. Navedeni nastavni materijal možete pogledati na mrežnoj adresi https://www.math.pmf.unizg.hr/sites/default/files/pictures/izn-skripta-2009.pdf.

U nastavku navodimo još neke primjere iz matematike gdje se susrećemo s algoritmima.

10.1.1 Primjeri algoritama

- Najpoznatiji su algoritmi oni za zbrajanje, oduzimanje, množenje i dijeljenje decimalnih brojeva koje se uče još u osnovnoj školi. Prvi ih je zapisao arapski matematičar Al-Khwarizmi u IX. stoljeću.
- Jedan od najstarijih algoritama je svakako **Euklidov algoritam.** Euklidov algoritam rješava sljedeći problem: za dane prirodne brojeve n i m treba odrediti najveću zajedničku mjeru.
- Algoritam za određivanje **drugog korijena** proizvoljnog decimalnog broja. (vidi I. Gusić, *Matematički rječnik*, str. 6). (Postoji i algoritam za određivanje trećeg korijena!)

- Algoritam za rješavanje **Cramerovog sustava** linearnih algebarskih jednadžbi.

Pojam algoritma je precizno definiran tek u XX.-tom stoljeću. Prirodno se postavlja pitanje kako to da su se tako dugo matematičari bezbrižno snalazili s nepreciznim pojmom algoritma. Jednostavno, nije bilo potrebe za tom definicijom. Potreba za definicijom pojavila se tek kada se željelo dokazati da ne postoji algoritam za rješavanje nekog problema. Sada ističemo jedan problem za koji se pokazalo da ne postoji algoritam za njegovo rješavanje.

Neka je $P(x_1, \ldots, x_n)$ polinom s cjelobrojnim koeficijentima. Jednadžba oblika $P(x_1, \ldots, x_n) = 0$ naziva se **diofantska jednadžba**. Takve su, primjerice, sljedeće jednadžbe: $2x+3y = 1$, $x^2+y^2-z^2 = 0$ i $6x^{18}-x+3 = 0$. Postavlja se pitanje ima li dana diofantska jednadžba cjelobrojna rješenja. Za neke posebne slučajeve odavno su poznati algoritmi.

a) Diofantske jednadžbe s jednom nepoznanicom.
To su jednadžbe obilika $a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 = 0$, gdje su a_i cijeli brojevi. Ako je x_0 cjelobrojno rješenje gornje jednadžbe tada je očito x_0 djeljitelj slobodnog koeficijenta a_0. Dakle, da bi ispitali ima li dana jednadžba cjelobrojnih rješenja dovoljno je uvrstiti u jednadžbu sve djeljitelje koeficijenta a_0 (pozitivne i negativne), te ispitati je li neki od njih rješenje.

b) Linearne diofantske jednadžbe s dvije nepoznanice.
To su jednadžbe oblika: $ax + by = c$, gdje su $a, b, c \in \mathbb{Z}$. Očito najveća zajednička mjera brojeva a i b, tj. $M(a, b)$ mora dijeliti c. Iz tog razloga dovoljno je promatrati slučaj kada je $M(a, b) = 1$. Euklidov algoritam daje sva rješenja linearnih diofantskih jednadžbi s dvije nepoznanice.

Na drugom svjetskom kongresu matematičara u Parizu 1900. godine poznati njemački matematičar David Hilbert je dao popis 23 teška problema i upozorio matematičare na njihovu važnost za dalji razvoj matematike. Deseti Hilbertov problem glasi:

> *Odrediti algoritam koji za svaku diofantsku jednadžbu određuje ima li ona cjelobrojno rješenje.*

J. V. Matijasević je 1969. godine dokazao da traženi algoritam ne postoji.

10.1.2 Intuitivni opisi nekih pojmova

Sada nam je cilj prvo intuitivno objasniti pojmove kojima ćemo se baviti, odnosno pojmove koje imamo namjeru kasnije formalno definirati. **Izračunavanje** je proces kod kojeg iz nekih početno danih objekata s fiksiranim skupom pravila dobivamo krajnji rezultat. Početne objekte nazivamo **ulazni podaci.** Fiksirani skup pravila naziva se **algoritam**. Krajnji rezultati se nazivaju **izlazni podaci.** Mi ćemo uvijek

pretpostavljati da postoji najviše **jedan** izlazni podatak. (Ako želimo promatrati izračunavanje s k izlaznih podataka možemo promatrati k izračunavanja koje svako ima samo jedan izlazni rezultat.) U drugu ruku **dozvoljavamo svaki konačan broj ulaznih podataka,** uključujući i nula ulaznih podataka. Pretpostavljamo da je za svaki pojedini algoritam **fiksiran broj ulaznih podataka.** Ne zahtijevamo da za sve ulazne podatke svaki algoritam daje izlazni rezultat. To znači da neki algoritam za neke ulazne podatke može računati bez da ikad stane. Kod algoritma mora biti točno precizirano što je akcija koja se izvodi u svakom koraku. Te akcije moraju biti u dovoljnoj mjeri mehaničke (kako bi se mogle izvoditi i na računalu).

Neka je $f : S \subseteq \mathbb{N}^k \to \mathbb{N}$. Smatramo da algoritam A sa k ulaznih podataka **izračunava** funkciju f ako vrijedi:

> *prirodni brojevi $x_1, \ldots, x_k$ su u domeni funkcije f ako i samo ako algoritam A prilikom izračunavanja s ulaznim podacima $x_1, \ldots, x_k$ stane, tj. daje izlazni rezultat. Ako je algoritam A stao tada je izlazni rezultat jednak $f(x_1, \ldots, x_k)$.*

Bilo je dosta teško te pojmove opisati precizno. Iz tog razloga ćemo dati definicije tih pojmova na nekoliko različitih načina. Definirat ćemo klase **RAM-izračunljivih funkcija** i **parcijalno rekurzivnih funkcija**. Iz definicije će biti jasno da je svaka ta funkcija izračunljiva u prije navedenom intuitivnom smislu. Nakon proučavanja obiju klasa (prije svega da se klase poklapaju) dat ćemo argumente za tvrdnju da svaka izračunljiva funkcija (u intuitivnom smislu) pripada tim klasama.

Mogli bismo pomisliti da, čim je zadana neka funkcija $f : \mathbb{N} \to \mathbb{N}$, onda automatski imamo i "efektivni postupak" za izračunavanje njenih vrijednosti $f(n)$. Sljedeći jednostavan primjer pokazuje da je pojam "**efektivne izračunljivosti**" vrlo suptilan. Neka je funkcija $f : \mathbb{N} \to \mathbb{N}$ zadana ovako:

$$f(n) = \begin{cases} 1, & \text{ako postoji n uzastopnih petica u decimalnom} \\ & \text{zapisu broja } \sqrt{2} \\ 0, & \text{inače} \end{cases}$$

10.1.3 Termini i oznake

Mi ćemo promatrati samo algoritme čiji su ulazni podaci i izlazni podatak **prirodni brojevi.** Kada govorimo **skup** mislimo uvijek na neki podskup prirodnih brojeva. Često ćemo umjesto uređene k–torke prirodnih brojeva $(x_1, \ldots, x_k)$ pisati samo kratko $\vec{x}$. Kada kažemo k–**mjesna funkcija** tada mislimo na neku funkciju čija je domena podskup od $\mathbb{N}^k$, a kodomena je skup $\mathbb{N}$. Ako kažemo samo **funkcija** tada mislimo na neku k–mjesnu funkciju. Funkcija je **totalna** ako je njena domena skup $\mathbb{N}^k$. Ako želimo naglasiti da neka funkcija moguće nije totalna tada kažemo da je ona **parcijalna**. Kada kažemo "k–mjesna relacija" tada mislimo na neki podskup od $\mathbb{N}^k$,

za neki $k \in \mathbb{N}$. Ako je R neka relacija tada činjenicu $\vec{x} \in R$ zapisujemo kao i $R(\vec{x})$. Ako je R dvomjesna relacija tada umjesto $R(x, y)$ pišemo i xRy. Ako je R relacija tada sa χ_R označavamo **karakterističnu funkciju** relacije R, koja je definirana na sljedeći način:

$$\chi_R(\vec{x}) = \begin{cases} 1, & \text{ako vrijedi } \ R(\vec{x}); \\ \\ 0, & \text{inače.} \end{cases}$$

Primijetite da je χ_R totalna funkcija za svaku relaciju R. Smatramo da je relacija R izračunljiva ako je pripadna funkcija χ_R izračunljiva. Kada relaciji pridjeljujemo neka funkcijska svojstva mislimo uvijek o svojstvu karakteristične funkcije.

10.2 RAM–stroj

RAM–stroj (end. random access machine) je idealizirano računalo s beskonačno velikom memorijom koje nikad ne radi greške. Definicija RAM-stroja koja slijedi je opisna. Mogli bi dati definiciju koja bi bila formalnija ("RAM-stroj je uređena n-torka ...") i "više" matematička, ali tada taj pojam ne bi bio toliko jasan.

Definicija 10.1. *Osnovni dijelovi RAM–stroja su:*

- *traka s registrima;*
- *spremnik za program;*
- *brojač.*

Za svaki prirodan broj k stroj ima registar koji označavamo sa $\mathcal{R}_k$. U svakom trenutku rada stroja svaki registar $\mathcal{R}_k$ sadrži neki prirodan broj. U **spremniku za program** je smješten program. **Program** je konačan niz instrukcija. Ako je n broj instrukcija u programu tada su one numerirane sa $1, 2, \ldots, n$. U **brojaču** se u svakom trenutku rada RAM-stroja nalazi redni broj instrukcije koja se izvršava. Postoje četiri tipa instrukcija za RAM–stroj:

1) INC $\mathcal{R}_k$
   Kada stroj izvodi tu instrukciju tada povećava broj u registru $\mathcal{R}_k$ za jedan, te broj u brojaču poveća za jedan.

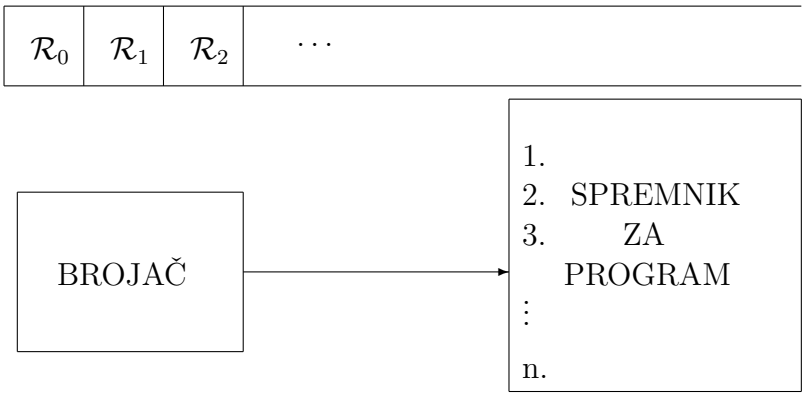
2) DEC $\mathcal{R}_k,\ m$.
   Broj m je obavezno broj neke instrukcije u programu. Ako je broj u registru $\mathcal{R}_k$ različit od nule tada se prilikom izvršenja navedene instrukcije broj u $\mathcal{R}_k$ smanji za jedan, a broj u brojaču se poveća za jedan. Ako je broj u registru $\mathcal{R}_k$ jednak nuli tada se prilikom izvršenja navedene instrukcije samo broj u brojaču promijeni u m.

3) GO TO m
   Broj m je obavezno broj neke instrukcije u programu. Kada stroj izvodi tu instrukciju on jednostavno broj u brojaču mijenja u m.

4) STOP
   Ova instrukcija znači bezuvjetni prekid izvođenja programa.

Za primjenu stroja **prvo stavljamo program** u spremnik programa. Zatim, **upisujemo** odgovarajuće brojeve u registre (ulazni podaci). Ako se radi o programu sa k ulaznih podataka, tada smatramo da su oni redom zapisani u registrima $\mathcal{R}_1$, $\ldots$, $\mathcal{R}_k$. U **brojaču je na početku broj** 1 (to znači da se svaki program počinje izvršavati od prve instrukcije). Tada startamo stroj. Stroj tada počinje izvršavati instrukcije. U svakom koraku stroj izvršava instrukciju u programu čiji je redni broj u brojaču. Na kraju izvršenja instrukcije mijenja se broj u brojaču. Ako je u nekom trenutku **broj u brojaču veći od broja instrukcija** u programu ili pak treba biti izvršena instrukcija STOP, tada stroj staje. U tom slučaju je izlazni rezultat zapisan u registru $\mathcal{R}_0$. Ako se to nikad ne dogodi stroj radi "vječno".

Na početku smo rekli da je RAM–stroj neka vrsta idealiziranog računala koja nikad ne griješi. Ako stroj nikad ne stane prilikom izvršenja nekog programa to ne znači da stroj griješi, već je program takav. (Sjetimo se samo koliko puta su nam probleme stvarale beskonačne petlje koje se znaju dogoditi prilikom nepažljivog programiranja.) Beskonačno izvršavanja stroja ne dopuštamo samo kako bismo mogli opravdati grešku u programu, već će nam ta neodređenost biti oznaka za jedno svojstvo funkcije čija se vrijednost izračunava. Sljedećom slikom dajemo skicu RAM-stroja.



Sada dajemo neke jednostavne primjere RAM–programa.

a) Program koji broj u registru $\mathcal{R}_k$ povećava za tri.

1. INC $\mathcal{R}_k$
2. INC $\mathcal{R}_k$
3. INC $\mathcal{R}_k$

b) Program koji broj u registru $\mathcal{R}_k$ zamijeni s nulom.

1. DEC $\mathcal{R}_k$, 3
2. GO TO 1
3. STOP

c) Program koji nikad ne staje.

1. GO TO 2
2. GO TO 1

Važno je **naglasiti da nam nije cilj učenje principa programiranja** (s minimalnim brojem instrukcija!). Glavni cilj je dati dvije različite definicije koje opisuju izračunljive funkcije, te dokazati da se klase funkcija, koje te dvije definicije određuju, poklapaju.

Definicija 10.2. *Neka je* $f : S \subseteq \mathbb{N}^k \to \mathbb{N}$ *i* P *program za RAM–stroj. Kažemo da program* P **izračunava** *funkciju* f *ako za sve prirodne brojeve* $x_1, \ldots, x_k$ *vrijedi da je* $(x_1, \ldots, x_k) \in S$ *ako i samo ako RAM–stroj s programom* P *u spremniku i s* $(x_1, \ldots, x_k)$ *kao ulaznim podacima stane, i u tom slučaju je u registru* $\mathcal{R}_0$ *zapisan broj* $f(x_1, \ldots, x_k)$.
Kažemo da je **funkcija** f **RAM–izračunljiva** *ako postoji program za RAM-stroj koji je izračunava. Kažemo da je* **relacija** R **RAM–izračunljiva** *ako je njena karakteristična funkcija RAM–izračunljiva.*

Želimo posebno **naglasiti** da iz prethodne definicije slijedi da ako je P program koji izračunava funkciju f, te su u RAM–stroju ulazni podaci koji ne pripadaju domeni funkcije f, tada RAM-stroj nikada neće stati. **Smatramo** da je svaka RAM–izračunljiva funkcija izračunljiva. (Sve instrukcije RAM–programa su jasne i "mehaničke".) Kasnije ćemo dati nekoliko primjera RAM–izračunljivih funkcija, te dokazati da je jedna velika klasa funkcija RAM-izračunljiva.

Poglavlje 11

Jedanaesto predavanje – izračunljivost

11.1 Makro–stroj

Dosta je teško pisati programe za RAM–stroj jer na raspolaganju imamo samo četiri tipa instrukcija. Sada ćemo uvesti nove tipove instrukcije. To će zapravo biti pokrate za čitave nizove osnovnih instrukcija. Ideja uvođenja novih instrukcija je slična kao kod programiranja – primjenjujemo potprograme, ili kao što se danas češće nazivaju **makroi**.

Definicija 11.1. *Za svaki program P za RAM–stroj uvodimo novu instrukciju koju označavamo sa P^* i nazivamo makro za P.* **Makro-stroj** *se sastoji od istih dijelova kao i RAM–stroj, te za svaki program P za RAM–stroj prepoznaje instrukciju P^*. Kada makro–stroj počinje izvršavati instrukciju oblika P^* on počinje izvršavati zapravo program P (s podacima koji su trenutno u registrima).*

Ako izvršavanje program P s danim podacima nikad ne stane tada smatramo da izvršavanje instrukcije P^* nije kompletirano, tj. makro–stroj ne staje. Ako izvršavanje programa P stane tada se broj u brojaču poveća za jedan (u odnosu na broj u brojaču na početku izvršavanja instrukcije P^*), te makro–stroj nastavlja izvršavati daljnje instrukcije. Pojam **makro-izračunljive** funkcije se definira na isti način kao pojam RAM–izračunljive funkcije.

Definicija 11.2. *Neka su S i S' strojevi (RAM ili makro). Označimo sa $\mathcal{R}_k$ proizvoljni registar stroja S, a s $\mathcal{R}'_k$ proizvoljni registar stroja S'. Neka je P program za stroj S, a P' program za stroj S'. Kažemo da su programi P i P'* **ekvivalentni** *ako za sve $\vec{x}$, koji su zapisani u S i S' kao ulazni podaci, oba stroja ili rade beskonačno, ili pak oba stanu i za sve $k \in \mathbb{N}$ u registrima $\mathcal{R}_k$ i $\mathcal{R}'_k$ je zapisan isti broj.*

Propozicija 11.3. *Za svaki program za makro–stroj postoji program za RAM–stroj koji je ekvivalentan s njim.*

Dokaz. Neka je Q program za makro-stroj. Za svaki makro P^* u programu Q zamijenimo makro P^* s nizom instrukcija programa P. Zatim, prenumeriramo na odgovarajući način redne brojeve instrukcija, te brojeve u instrukcijama oblika DEC $\mathcal{R}_i,\ m$ i GO TO m. Očito je dobiveni RAM–program ekvivalentan s makro–programom Q. Q.E.D

Korolar 11.4. *Klase RAM–izračunljivih funkcija i makro–izračunljivih funkcija su jednake.*

Sada uvodimo oznake za makroe koje ćemo češće koristiti. Makro programa iz prvog primjera RAM–programa, koji broj u registru $\mathcal{R}_k$ izjednačava s nulom, označavat ćemo sa **ZERO** $\mathcal{R}_k$.

Sljedeći program za makro–stroj broj iz registra $\mathcal{R}_i$ prepisuje u registar $\mathcal{R}_j$, pri čemu na kraju izvršavanja programa broj u registru $\mathcal{R}_i$ nije promijenjen. Za izvršavanje tog programa koristimo kao pomoćni registar $\mathcal{R}_k$.

1. ZERO $\mathcal{R}_j$
2. ZERO $\mathcal{R}_k$
3. DEC $\mathcal{R}_i,\ 7$
4. INC $\mathcal{R}_j$
5. INC $\mathcal{R}_k$
6. GO TO 3
7. DEC $\mathcal{R}_k,\ 10$
8. INC $\mathcal{R}_i$
9. GO TO 7
10. STOP

Makro za prethodni program označavamo sa **MOVE** $\mathcal{R}_i$ **TO** $\mathcal{R}_j$ **USING** $\mathcal{R}_k$. Obično ćemo ispuštati pisanje pomoćnog registra $\mathcal{R}_k$. Smatrat ćemo da je pomoćni registar izabran tako da je različit od svih drugih registara koji se koriste u programu. To znači da ćemo obično koristiti oznaku **MOVE** $\mathcal{R}_i$ **TO** $\mathcal{R}_j$.

Sada uvodimo još jednu makro–instrukciju. Neka je f neka k–mjesna RAM–izračunljiva funkcija, te neka je P program za RAM–stroj koji je izračunava. Neka je m najmanji prirodni broj koji ima svojstvo da se prilikom izvršavanja programa P ne koriste registri $\mathcal{R}_i$ za svaki $i \geq m$. Označimo sa Q sljedeći program za makro–stroj:

1. MOVE $\mathcal{R}_1$ TO $\mathcal{R}_{m+1}$ USING $\mathcal{R}_m$
$\vdots$
(m-1). MOVE $\mathcal{R}_{m-1}$ TO $\mathcal{R}_{2m-1}$ USING $\mathcal{R}_m$
(Brojevi iz registara $\mathcal{R}_1, \ldots, \mathcal{R}_{m-1}$ se prepisuju redom u registre $\mathcal{R}_{m+1}, \ldots, \mathcal{R}_{m+(m-1)}$.)
m. ZERO $\mathcal{R}_0$
(m+1). ZERO $\mathcal{R}_{k+1}$
$\vdots$
(2m-k-1). ZERO $\mathcal{R}_{m-1}$
(Brojevi u registrima $\mathcal{R}_0$, $\mathcal{R}_{k+1}, \ldots, \mathcal{R}_{m-1}$ se brišu.)
(2m-k). P^*
(Program P koristi registre $\mathcal{R}_0$, $\mathcal{R}_1$, …, $\mathcal{R}_k$, …, $\mathcal{R}_{m-1}$).
(2m-k+1). MOVE $\mathcal{R}_{m+1}$ TO $\mathcal{R}_1$ USING $\mathcal{R}_m$
$\vdots$
(3m-k+1). MOVE $\mathcal{R}_{2m-1}$ TO $\mathcal{R}_{m-1}$ USING $\mathcal{R}_m$
(Brojevi iz registara $\mathcal{R}_{m+1}, \ldots, \mathcal{R}_{m+(m-1)}$ se vraćaju redom u registre $\mathcal{R}_1, \ldots, \mathcal{R}_{m-1}$.)

Provjerite da vrijedi: ako je program Q pokrenut s ulaznim podacima $x_1, \ldots, x_k$ u registrima $\mathcal{R}_1, \ldots, \mathcal{R}_k$ tada će se makro–stroj zaustaviti ako i samo ako je $f(x_1, \ldots, x_k)$ definirano, i u tom slučaju je broj u registru $\mathcal{R}_0$ jednak $f(x_1, \ldots, x_k)$, a brojevi u registrima $\mathcal{R}_i$ su nepromijenjeni osim možda u registrima s indeksom i, gdje je $i = 0$ ili vrijedi $m \leq i < 2m$. Makro za prethodni program označavamo sa $f(\mathcal{R}_1, \ldots, \mathcal{R}_k) \to \mathcal{R}_0$.

11.1.1 Zadaci

1. Neka je $f : \mathbb{N}^2 \to \mathbb{N}$ funkcija definirana sa $f(x, y) = 1$, ako je $x = y$, a inače je $f(x, y) = 0$. Napišite program za RAM–stroj koji računa funkciju f.
Rješenje.

1. DEC $\mathcal{R}_0$, 3
2. GO TO 1
3. DEC $\mathcal{R}_1$, 6
4. DEC $\mathcal{R}_2$, 9
5. GO TO 3
6. DEC $\mathcal{R}_2$, 8
7. GO TO 9
8. INC $\mathcal{R}_0$
9. STOP

2. Neka je u registru $\mathcal{R}_1$ zapisan broj n, a u registru $\mathcal{R}_2$ broj m. Napišite program za makro–stroj koji provjerava je li n strogo veći od m. Ako jest neka stroj

stane i u registru $\mathcal{R}_0$ neka bude zapisan broj 0. Ako pak nije, neka također stroj stane, a u registru $\mathcal{R}_0$ neka bude zapisan broj 1.
Rješenje.

1. ZERO $\mathcal{R}_0$
2. INC $\mathcal{R}_0$
3. MOVE $\mathcal{R}_1$ TO $\mathcal{R}_3$
4. MOVE $\mathcal{R}_2$ TO $\mathcal{R}_4$
5. DEC $\mathcal{R}_3$, 9
6. DEC $\mathcal{R}_4$, 8
7. GO TO 5
8. ZERO $\mathcal{R}_0$
9. STOP

3. Napišite program za makro–stroj koji prepoznaje je li broj n u registru $\mathcal{R}_1$ djeljiv s tri. Ako je djeljiv tada neka stroj stane i neka u registru $\mathcal{R}_0$ bude zapisana nula. Ako pak broj n nije djeljiv s tri neka stroj ne stane.
Rješenje.

1. ZERO $\mathcal{R}_2$
2. MOVE $\mathcal{R}_1$ TO $\mathcal{R}_2$
3. DEC $\mathcal{R}_2, 9$
4. DEC $\mathcal{R}_2, 7$
5. DEC $\mathcal{R}_2, 7$
6. GO TO 3
7. GO TO 8
8. GO TO 7
9. STOP

4. Napišite program za makro–stroj koji izračunava funkciju $n \mapsto \lfloor \sqrt{n} \rfloor$ (najveće cijelo drugog korijena od n).
Rješenje. Označimo s $\mathcal{R}_i^2 \to \mathcal{R}_j$ makro za program koji broj iz registra $\mathcal{R}_i$ kvadrira i zapisuje u registar $\mathcal{R}_j$ (smatramo da je nakon toga broj u registru $\mathcal{R}_i$ nepromijenjen). Označimo s IF $\mathcal{R}_i \leq \mathcal{R}_j < \mathcal{R}_k$ THEN p ELSE q makro za program koji provjerava je li broj zapisan u registru $\mathcal{R}_i$ manji od broja zapisanog u registru $\mathcal{R}_j$, a taj je strogo manji od broja u zapisanog u registru $\mathcal{R}_k$. Ako je to ispunjeno program se nastavlja izvršavati od p–te instrukcije, a ako nije program se nastavlja izvršavati od q. instrukcije. Sada prvo dajemo program za makro–stroj koji u registar $\mathcal{R}_0$ zapisuje $\lfloor \sqrt{n} \rfloor$ (ako je na početku u registru $\mathcal{R}_1$ zapisan broj n.) Nakon toga ćemo dati programe za makro–stroj koji će

opravdati uvođenje gornjih makroa.

1. ZERO $\mathcal{R}_0$
2. ZERO $\mathcal{R}_2$
3. INC $\mathcal{R}_2$
4. $\mathcal{R}_0^2 \to \mathcal{R}_3$
5. $\mathcal{R}_2^2 \to \mathcal{R}_4$
6. IF $\mathcal{R}_3 \leq \mathcal{R}_1 < \mathcal{R}_4$ THEN 9 ELSE 7
7. INC $\mathcal{R}_0$
8. GO TO 2
9. STOP

Sada navodimo jedan program za makro–stroj koji broj zapisan u registru $\mathcal{R}_i$ kvadrira i rezultat zapisuje u registar $\mathcal{R}_j$. (Registri $\mathcal{R}_k$ i $\mathcal{R}_m$ su pomoćni u ovom programu).

1. ZERO $\mathcal{R}_j$
2. ZERO $\mathcal{R}_m$
3. MOVE $\mathcal{R}_i$ TO $\mathcal{R}_k$
4. GO TO 7
5. $\mathcal{R}_i + \mathcal{R}_j \to \mathcal{R}_m$
6. MOVE $\mathcal{R}_m$ TO $\mathcal{R}_j$
7. DEC $\mathcal{R}_k$, 9
8. GO TO 5
9. STOP

Preostalo je napisati program za makro–stroj koji opravdava makro IF $\mathcal{R}_i \leq \mathcal{R}_j < \mathcal{R}_k$ THEN p ELSE q koji smo prije koristili. Registre $\mathcal{R}_l$, $\mathcal{R}_m$ i $\mathcal{R}_s$ koristimo kao pomoćne.

1. MOVE $\mathcal{R}_i$ TO $\mathcal{R}_l$
2. MOVE $\mathcal{R}_j$ TO $\mathcal{R}_m$
3. MOVE $\mathcal{R}_k$ TO $\mathcal{R}_s$
4. DEC $\mathcal{R}_s$, q
5. DEC $\mathcal{R}_m$, 8
6. DEC $\mathcal{R}_l$, 4
7. GO TO 4
8. DEC $\mathcal{R}_l$, p
9. GO TO q

Uočite da ovo posljednje nije korektni program za makro–stroj jer u njemu ne postoje instrukcije s rednim brojevima p i q. No, to nije problem, jer smo mogli ovaj posljedni program napisati u programu koji rješava zadatak umjesto makro–instrukcije IF $\mathcal{R}_i \leq \mathcal{R}_j < \mathcal{R}_k$ THEN 9 ELSE 7.

5. Napišite program za makro–stroj koji izračunava sljedeću funkciju:

$$f(n) = \begin{cases} 0, & \text{ako je } n = 0; \\ \lfloor \log n \rfloor, & \text{inače} \end{cases}$$

Rješenje. Označimo s $10^{\mathcal{R}_i} \to \mathcal{R}_j$ makro za program koji čita broj iz registra $\mathcal{R}_i$ i računa potenciju broja 10, te rezultat zapisuje u registar $\mathcal{R}_j$ (smatramo da je nakon toga broj u registru $\mathcal{R}_i$ nepromijenjen). Označimo s IF $\mathcal{R}_i \leq \mathcal{R}_j < \mathcal{R}_k$ THEN p ELSE q makro kao u rješenju prethodnog zadatka. Prvo dajemo program za makro–stroj koji u registar $\mathcal{R}_0$ zapisuje $f(n)$ (ako je na početku u registru $\mathcal{R}_1$ zapisan broj n).

1. ZERO $\mathcal{R}_0$
2. MOVE $\mathcal{R}_1$ TO $\mathcal{R}_5$
3. DEC $\mathcal{R}_5$, 12
4. ZERO $\mathcal{R}_2$
5. INC $\mathcal{R}_2$
6. $10^{\mathcal{R}_0} \to \mathcal{R}_3$
7. $10^{\mathcal{R}_2} \to \mathcal{R}_4$
8. IF $\mathcal{R}_3 \leq \mathcal{R}_1 < \mathcal{R}_4$ THEN 11 ELSE 9
9. INC $\mathcal{R}_0$
10. GO TO 5
11. STOP

Sada dajemo program za makro–stroj koji računa i zapisuje broj 10^n u registar $\mathcal{R}_j$, ako je na početku u registru $\mathcal{R}_i$ zapisan broj n. (Registri $\mathcal{R}_k$ i $\mathcal{R}_m$ su pomoćni u ovom programu).

1. ZERO $\mathcal{R}_j$
2. INC $\mathcal{R}_j$
3. ZERO $\mathcal{R}_m$
4. MOVE $\mathcal{R}_i$ TO $\mathcal{R}_k$
5. GO TO 8
6. $10 \cdot \mathcal{R}_j \to \mathcal{R}_m$
7. MOVE $\mathcal{R}_m$ TO $\mathcal{R}_j$
8. DEC $\mathcal{R}_k$, 10
9. GO TO 6
10. STOP

Makro $10 \cdot \mathcal{R}_j \to \mathcal{R}_m$ nećemo posebno opisivati.

6. Neka je $f : \{1, 2, 3\} \to \mathbb{N}$ funkcija definirana sa $f(i) = i$. Napišite program za RAM–stroj koji računa funkciju f.

7. Neka je $S = \{(x, x) : x \in \mathbb{N}\}$. Neka je $f : S \to \mathbb{N}$ funkcija definirana sa $f(s) = 0$, za svaki $s \in S$. Napišite program za RAM–stroj koji računa funkciju f.

8. Neka je $f : \mathbb{N}^2 \setminus \{(1, 2)\} \to \mathbb{N}$ funkcija definirana sa $f(x, y) = x$. Napišite program za RAM–stroj koji računa funkciju f.

9. Neka je $S = \{(i, j) : i, j \in \mathbb{N},\ i \leq j\}$. Neka je funkcija $f : S \to \mathbb{N}$ definirana sa $f(i, j) = i$. Napišite program za RAM–stroj koji računa funkciju f.

10. Neka je S podskup od $\mathbb{N}^3$ definiran sa $S = \{(n, 1, k) \mid n, k \in \mathbb{N}\}$. Neka je $f : S \to \mathbb{N}$ funkcija definirana sa $f(n, 1, k) = n$. Napišite program za RAM–stroj koji računa funkciju f.

11. Neka je $f : \mathbb{N}^2 \to \mathbb{N}$ funkcija definirana sa $f(a, b) = a^b$, $(a, b) \neq (0, 0)$, $f(0, 0) = 0$. Napišite program za makro–stroj koji računa funkciju f.

12. Neka je funkcija $f : 2\mathbb{N} \to \mathbb{N}$ definirana sa $f(x) = x$. Napišite program za RAM–stroj koji računa funkciju f.

13. Neka je $S = \{(n, n) : n \in \mathbb{N}\}$. Neka je $f : \mathbb{N}^2 \setminus S \to \mathbb{N}$ funkcija definirana sa $f(x, y) = \min\{x, y\}$. Napišite program za RAM–stroj koji računa funkciju f.

14. Napišite program za makro–stroj koji izračunava funkciju $f : \mathbb{N} \to ,\ f(x) = x^2$. Postoji li program za RAM–stroj koji izračunava funkciju f?

15. Neka je funkcija $\dot{-} : \mathbb{N}^2 \to \mathbb{N}$ definirana ovako:

$$x \dot{-} y = \begin{cases} 0, & \text{ako je } x \leq y; \\ x - y, & \text{inače.} \end{cases}$$

Napišite program za makro–stroj koji izračunava funkciju $\dot{-}$.

11.2 Rekurzivne funkcije

Sada definiramo još jednu klasu funkcija za koje će odmah iz definicije biti jasno su izračunljive u intuitivnom smislu. Dokazat ćemo da se ta nova klasa funkcija poklapa s klasom svih RAM–izračunljivih funkcija.

11.2.1 Inicijalne funkcije

Definicija 11.5. *Funkciju $Z : \mathbb{N} \to \mathbb{N}$ definiranu sa $Z(x) = 0$ nazivamo* **nul-funkcija.** *Funkciju $Sc : \mathbb{N} \to \mathbb{N}$ definiranu sa $Sc(x) = x + 1$ nazivamo* **funkcija sljedbenika** *(eng. successor). Neka je $n \in \mathbb{N}$ i $k \in \{1, \ldots, n\}$. Funkciju $I_k^n : \mathbb{N}^n \to \mathbb{N}$ definiranu sa $I_k^n(x_1, \ldots, x_n) = x_k$ nazivamo* **projekcija.** *Funkcije Z, Sc i I_k^n $(n \in \mathbb{N},\ k \leq n)$ nazivamo* **inicijalne funkcije**.

Propozicija 11.6. *Svaka inicijalna funkcija je RAM–izračunljiva.*

Dokaz. Redom za svaku inicijalnu funkciju pišemo program za makro–stroj koji je izračunava. Za nul–funkciju Z jedan program za makro–stroj koji je izračunava je: 1. ZERO $\mathcal{R}_0$. Za funkciju Sc jedan program za makro–stroj koji je izračunva je:

1. MOVE $\mathcal{R}_1$ TO $\mathcal{R}_0$
2. INC $\mathcal{R}_0$

Jedan program za makro–stroj koji izračunava funkciju I_k^n je: 1. MOVE $\mathcal{R}_k$ TO $\mathcal{R}_0$ Q.E.D.

11.2.2 Parcijalne funkcije

U daljnjim izlaganjima promatrat ćemo i funkcije koje nisu totalne, pa ćemo se često susretati s izrazima koji za neke prirodne brojeve nisu definirani. Ako je izraz X nedefiniran za neki $\vec{x} \in \mathbb{N}^k$ tada pišemo $X(\vec{x}) \uparrow$, a inače $X(\vec{x}) \downarrow$. Analogno, ako je f parcijalna funkcija, te $\vec{x} \in \mathbb{N}^k$ koji nije u domeni funkcije f, tada to kratko označavamo s $f(\vec{x}) \uparrow$. Ako pak je $\vec{x}$ u domeni funkcije tada to kratko označavamo s $f(\vec{x}) \downarrow$. Neka su X i Y neki izrazi. Sa $X \simeq Y$ označavamo da za svaku uređenu k–torku prirodnih brojeva vrijedi:

$$X(\vec{x}) \downarrow,\ Y(\vec{x}) \downarrow \quad \text{i} \quad X(\vec{x}) = Y(\vec{x});$$

ili

$$X(\vec{x}) \uparrow \quad \text{i} \quad Y(\vec{x}) \uparrow$$

Primjer 11.7. *1. Vrijedi* $2 \simeq \frac{2(x+1)}{x+1}$

2. Ne vrijedi $7 \simeq \frac{7(x-y)}{x-y}$ *jer u slučaju* $x = y$ *izraz* $\frac{7(x-y)}{x-y}$ *je nedefiniran.*

3. Neka je $S = \{0, 2, 4, 6, 8\}$ *i* $F, G : S \to \mathbb{N}$ *funkcije koje su definirane ovako:*

$$F(n) = n + 1 \qquad i \qquad G(n) = \frac{n^2 - 1}{n - 1}$$

Tada vrijedi $F \simeq G$.

11.2.3 Primitivno rekurzivne funkcije

Definicija 11.8. *Neka su* G, H_1, $\ldots$ H_n *funkcije. Neka je funkcija* F *definirana sa:*

$$F(\vec{x}) \simeq G(H_1(\vec{x}), \ldots, H_n(\vec{x})).$$

Tada kažemo da je funkcija F *definirana pomoću* **kompozicije** *funkcija.*

Propozicija 11.9. *Klasa RAM–izračunljivih funkcija je zatvorena za kompoziciju, tj. točnije ako su i* G, H_1, $\ldots$ H_n *neke RAM–izračunljive funkcije, tada je i funkcija* F *koja je definirana sa:*

$$F(\vec{x}) \simeq G(H_1(\vec{x}), \ldots H_n(\vec{x}))$$

također RAM-izračunljiva.

Dokaz. Navodimo jedan program za makro–stroj koji izračunava funkciju F.

$$
\begin{array}{rl}
1. & H_1(\mathcal{R}_1, \ldots, \mathcal{R}_k) \to \mathcal{R}_{k+1} \\
\vdots & \\
\text{n.} & H_n(\mathcal{R}_1, \ldots, \mathcal{R}_k) \to \mathcal{R}_{k+n} \\
\text{(n+1).} & G(\mathcal{R}_{k+1}, \ldots, \mathcal{R}_{k+n}) \to \mathcal{R}_0
\end{array}
$$

(Uočite da je ovdje važno da nakon niti jednog koraka prilikom izvršavanja programa nisu promijenjeni ulazni podaci u registrima $\mathcal{R}_1, \ldots, \mathcal{R}_k$). Q.E.D.

Definicija 11.10. *Neka je G totalna k–mjesna funkcija, H $(k+2)$–mjesna totalna funkcija. Neka je $(k+1)$–mjesna funkcija F definirana na sljedeći način:*

$$
\begin{aligned}
F(0, \vec{x}) &= G(\vec{x}) \\
F(y+1, \vec{x}) &= H(F(y, \vec{x}), y, \vec{x})
\end{aligned}
$$

Tada kažemo da je funkcija F definirana pomoću **primitivne rekurzije**.
Ako je $k = 0$ tada definicija funkcije F pomoću primitivne rekurzije izgleda ovako:

$$
\begin{aligned}
F(0) &= a \qquad (a \in \mathbb{N}) \\
F(y+1) &= H(F(y), y)
\end{aligned}
$$

Propozicija 11.11. *Klasa RAM–izračunljivih funkcija je zatvorena za primitivnu rekurziju, tj. ako su G i H neke RAM–izračunljive totalne funkcije tada je funkcija F, koja je definirana pomoću primitivne rekurzije iz G i H, također RAM–izračunljiva.*

Dokaz. Radi kraćeg zapisivanja promatramo slučaj kada je $k = 1$. Sada navodimo program za makro–stroj koji izračunava funkciju F.

1. $G(\mathcal{R}_2) \to \mathcal{R}_0$
2. MOVE $\mathcal{R}_1$ TO $\mathcal{R}_3$
3. ZERO $\mathcal{R}_1$
4. DEC $\mathcal{R}_3, 9$
5. $H(\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2) \to \mathcal{R}_4$
6. MOVE $\mathcal{R}_4$ TO $\mathcal{R}_0$
7. INC $\mathcal{R}_1$
8. GO TO 4
9. STOP

Q.E.D.

Definicija 11.12. *Najmanja klasa totalnih funkcija koja sadrži inicijalne funkcije, te je zatvorena za kompoziciju i primitivnu rekurziju, naziva se klasa* **primitivno rekurzivnih funkcija.**

Uočeno je da inicijalne funkcije, te kompozicija i primitivna rekurzija, nisu dovoljni kako bi se definirala svaka izračunljiva funkcija. Jedan primjer izračunljive funkcije koja nije primitivno rekurzivna je **Ackermanova funkcija**. Neka je funkcija $B : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ definirana pomoću primitivne rekurzije na sljedeći način:

$$B(0,y) = 2 + y$$

$$B(x+1,0) = sg(x)$$

$$B(x+1,y+1) = B(x,B(x+1,y))$$

Sada Ackermanovu funkciju A definiramo sa $A(x) = B(x,x)$. Sa sg je označena funkcija signum, tj. funkcija predznaka, koja je definirana sa:

$$sg(x) = \begin{cases} 0, & \text{ako je } x = 0, \\ 1, & \text{ako je } x > 0 \end{cases}$$

11.2.4 Primjeri primitivno rekurzivnih funkcija

Sljedeća propozicija govori o "virtualnim" varijablama. Bit će nam vrlo korisna u kasnijim razmatranjima.

Propozicija 11.13. *Neka je g primitivno rekurzivna k-mjesna funkcija, te $x_1, \ldots, x_n$ različite varijable. Neka je za sve i, gdje je $1 \leq i \leq k$, z_i jedna od varijabli $x_1, \ldots, x_n$. Neka je funkcija f definirana s*

$$f(x_1, \ldots, x_n) = g(z_1, \ldots, z_k).$$

Tada je funkcija f primitivno rekurzivna.

Dokaz. Neka je $z_i = x_{j_i}$, gdje je $1 \leq j_i \leq n$, $1 \leq i \leq k$. Tada za sve $i \in \{1, \ldots, k\}$ vrijedi $I^n_{j_i}(x_1, \ldots, x_n) = z_i$. Time imamo

$$f(x_1, \ldots, x_n) = g(I^n_{j_1}(x_1, \ldots, x_n), \ldots, I^n_{j_k}(x_1, \ldots, x_n)).$$

To znači da je funkcija f definirana pomoću kompozicije primitivno rekurzivnih funkcija, pa je primitivno rekurzivna po definiciji. Q.E.D.

Propozicija 11.14. *Za sve prirodne brojeve k i n nul–funkcija N, koja je definirana sa*
$N(x_1, \ldots, x_k) = 0$, i konstantna funkcija C_n, koja je definirana sa $C_n(x_1, \ldots, x_k) = n$, su primitivno rekurzivne.

Dokaz. Primjenom prethodne propozicije 11.13. uzimajući $g(x) = Z(x)$ slijedi da je funkcija N primitivno rekurzivna. Za funkcije C_n, $n \in \mathbb{N}$, dokaz provodimo indukcijom po n. Za $n = 0$ to je funkcija N za koju smo upravo dokazali da je primitivno rekurzivna. Očito za svaki $n \in \mathbb{N}$ vrijedi $C_{n+1}(\vec{x}) = Sc(C_n(\vec{x}))$. Q.E.D.

Propozicija 11.15. *Funkcije $(x,y) \mapsto x+y$, $(x,y) \mapsto x \cdot y$, $x \mapsto x!$ i $(x,y) \mapsto x^y$ su primitivno rekurzivne. (Po dogovoru stavljamo $0^0 = 1$, kako bi funkcija potenciranja bila totalna).*

Dokaz. Radi ilustracije dokazujemo za zbrajanje. Očito vrijedi:

$$\begin{aligned} x+0 &= I_1^1(x) \\ x+(y+1) &= Sc(I_1^3(x+y,y,x)). \end{aligned}$$

Q.E.D.

11.2.5 Parcijalno rekurzivne funkcije

Kako bi definirali klasu parcijalno rekurzivnih funkcija moramo definirati još jedan operator.

Definicija 11.16. *Neka je f funkcija. Sa $\mu y(f(\vec{x},y) \simeq 0)$ označavamo funkciju definiranu na sljedeći način:*

$$\mu y(f(\vec{x},y) \simeq 0) \simeq \begin{cases} z, & \text{najmanji } z, \text{ ako postoji,} \\ & \text{takav da je } f(\vec{x},y)\downarrow \quad \text{za sve } y<z, \\ & \text{te je } f(\vec{x},z)=0; \\ \\ \uparrow, & \text{inače} \end{cases}$$

*Tada kažemo da je funkcija $\mu y(f(\vec{x},y) \simeq 0)$ definirana pomoću μ-***operatora.**

Definicija 11.17. *Najmanja klasa funkcija koja sadrži sve inicijalne funkcije, te je zatvorena za kompoziciju, primitivnu rekurziju i μ–operator, naziva se klasa* **parcijalno rekurzivnih funkcija.** *Funkcija iz klase parcijalno rekurzivnih funkcija koja je totalna naziva se i* **rekurzivna funkcija**.

Definicija 11.18. *Kažemo da je relacija $R \subseteq \mathbb{N}^k$* **rekurzivna relacija** *ako je njena karakteristična funkcija rekurzivna. Kažemo da je neki skup $S \subseteq \mathbb{N}^k$* **rekurzivan** *ako je njegova karakteristična funkcija rekurzivna.*

Iz definicije slijedi da je svaka primitivno rekurzivna funkcija ujedno i rekurzivna, a onda i parcijalno rekurzivna.

Napomena 11.19. *U daljnjim razmatranjima koristit ćemo i sljedeću definiciju funkcije pomoću μ–operatora i relacije. Neka je R neka $(k+1)$–mjesna relacija. Sa $\mu y R(\vec{x},y)$ označavamo funkciju definiranu na sljedeći način:*

$$\mu y R(\vec{x},y) \simeq \begin{cases} \text{najmanji } z \text{ tako da vrijedi } \neg R(\vec{x},y) \quad \text{za sve } y<z \text{ i} \\ \text{vrijedi} \quad R(\vec{x},z), \text{ ako takav postoji;} \\ \\ \uparrow, \text{ inače.} \end{cases}$$

Lako je vidjeti da time nismo proširili klasu parcijalno rekurzivnih funkciju jer za svaku relaciju R vrijedi:

$$\mu y R(\vec{x}, y) \;\simeq\; \mu y\Big(1\dot{-}\chi_R(\vec{x}, y) \simeq 0\Big)$$

Funkcija $\dot{-}$ je definirana ovako:

$$x\dot{-}y = \begin{cases} x - y, & \text{ako je } \; x \geq y; \\ \\ 0, & \text{inače.} \end{cases}$$

Sljedeća napomena je jako važna jer naglašava grešku koja se često radi prilikom definicije μ–operatora.

Napomena 11.20. *Neka je f parcijalno rekurzivna funkcija. Zatim, neka je sa M označen operator koji je definiran sa:*

$$M(f)(\vec{x}) = \begin{cases} \textit{najmanji} \;\; z \;\; \textit{takav da je} \;\; f(\vec{x}, z) = 0, \\ \qquad\qquad\qquad \textit{ako takav} \;\; z \;\; \textit{postoji}; \\ \\ \qquad\qquad \uparrow, \;\; \textit{inače} \end{cases}$$

Uočite da se operator M razlikuje od μ–operatora u tome što nema zahtjeva da je vrijednost funkcije f definirana na svim vrijednostima y koje su manje od z. Kasnije ćemo pokazati da klasa RAM–izračunljivih funkcija nije zatvorena za operator M.

Propozicija 11.21. *Klasa RAM–izračunljivih funkcija je zatvorena za μ–operator, tj. ako je funkcija f RAM–izračunljiva, tada je funkcija $\mu y(f(\vec{x}, y) \simeq 0)$ također RAM-izračunljiva.*

Dokaz. Neka je f jedna $(k+1)$–mjesna funkcija. Navest ćemo jedan program za makro–stroj koji izračunava funkciju $\mu y(f(\vec{x}, y) \simeq 0)$ (koja je k–mjesna).

1. ZERO $\mathcal{R}_0$
2. $f(\mathcal{R}_1, \ldots, \mathcal{R}_k, \mathcal{R}_0) \to \mathcal{R}_{k+1}$
3. DEC $\mathcal{R}_{k+1}$, 6
4. INC $\mathcal{R}_0$
5. GO TO 2
6. STOP

Q.E.D.

Tvrdnja sljedećeg teorema slijedi iz prethodnih propozicija 11.6., 11.9., 11.11. i 11.21..

Teorem 11.22. *Svaka parcijalno rekurzivna funkcija je RAM–izračunljiva.*

U daljnjim razmatranjima cilj nam je dokazati obrat prethodnog teorema.

Poglavlje 12

Dvanaesto predavanje – izračunljivost

12.1 Rekurzivne funkcije

12.1.1 Primjeri primitivno rekurzivnih funkcija (nastavak)

Sada navodimo primjere primitivno rekurzivnih funkcija koje ćemo kasnije koristiti. Oduzimanje nije totalna funkcija na skupu $\mathbb{N}$. Iz tog razloga definiramo sljedeću funkciju (**modificirano oduzimanje**).

$$x \dot{-} y = \begin{cases} x - y, & \text{ako je } x \geq y; \\ 0, & \text{inače.} \end{cases}$$

Da bismo dokazali da je funkcija $\dot{-}$ primitivno rekurzivna prvo definiramo funkciju pr pomoću primitivne rekurzije ovako:

$$\begin{aligned} pr(0) &= 0 \\ pr(x+1) &= x \end{aligned}$$

Sada funkciju $\dot{-}$ možemo definirati pomoću primitivne rekurzije ovako:

$$\begin{aligned} x \dot{-} 0 &= x \\ x \dot{-} (y+1) &= pr(x \dot{-} y) \end{aligned}$$

Sa sg označavamo **funkciju predznaka**, tj. funkciju signum, definiranu sa:

$$sg(x) = \begin{cases} 0, & \text{ako je } x = 0; \\ 1, & \text{inače.} \end{cases}$$

Budući da funkciju sg možemo definirati pomoću primitivne rekurzije na sljedeći način:

$$sg(0) = 0$$

$$sg(x+1) = 1.$$

Iz toga slijedi da je funkcija signum primitivno rekurzivna. Označimo sa $\overline{sg}$ funkciju definiranu sa:

$$\overline{sg}(x) = \begin{cases} 1, & \text{ako je } x = 0; \\ 0, & \text{inače.} \end{cases}$$

Očito vrijedi $\overline{sg}(x) = 1 \dot{-} sg(x)$, pa je funkcija $\overline{sg}$ primitivno rekurzivna.

12.1.2 Rekurzivne relacije i skupovi

Ponovit ćemo definiciju rekurzivnih relacija i rekurzivnih skupova.

Definicija 12.1. *Kažemo da je relacija $R \subseteq \mathbb{N}^k$, $k \geq 1$,* **rekurzivna relacija** *ako je njena karakteristična funkcija rekurzivna.*

Uočite da je time definiran i pojam rekurzivnog skupa $S \subseteq \mathbb{N}^k$.

Propozicija 12.2. *Neka su R i P (primitivno) rekurzivne relacije. Tada su i sljedeće relacije (primitivno) rekurzivne:*

$$\neg R, \qquad R \wedge P, \qquad R \vee P, \qquad R \to P \ \text{ i } \ R \leftrightarrow P$$

Dokaz. Očito vrijedi $\chi_{\neg R}(\vec{x}) = 1 \dot{-} \chi_R(\vec{x})$ i $\chi_{R \wedge P}(\vec{x}) = \chi_R(\vec{x}) \cdot \chi_P(\vec{x})$. Budući da je $\{\neg, \wedge\}$ baza za skup svih propozicionalnih veznika tada sve druge relacije, koje su navedene u iskazu propozicije, možemo shvatiti kao pokrate. Q.E.D.

Korolar 12.3. *Neka su A i B rekurzivni skupovi. Tada su i A^c, $A \cap B$ i $A \cup B$ rekurzivni skupovi. Presjek, odnosno unija, konačno mnogo rekurzivnih skupova je rekurzivan skup.*

Propozicija 12.4. *Relacije $\leq$, $\geq$, $<$, $>$ i $=$ su primitivno rekurzivne.*

Dokaz. Lako je provjeriti da redom vrijedi:

$$\chi_{>}(x, y) = sg(x \dot{-} y)$$

$$x \leq y \ \text{ ako i samo ako } \ \neg(y > x)$$

$$x \geq y \ \text{ ako i samo ako } \ y \leq x$$

$$x < y \ \text{ ako i samo ako } \ \neg(y \leq x)$$

$$x = y \ \text{ ako i samo ako } \ (x \leq y \ \wedge \ y \leq x)$$

Sada primjenom propozicije 12.2. slijedi tvrdnja ove propozicije. Q.E.D.

Propozicija 12.5. (Definicija funkcije po slučajevima – verzija 1) *Neka su $R_1, \ldots R_n$ (primitivno) rekurzivne relacije koje imaju svojstvo da za svaki $\vec{x} \in \mathbb{N}^k$ postoji točno jedan $i \in \{1, \ldots, n\}$ tako da vrijedi $R_i(\vec{x})$. Neka su $F_1, \ldots, F_n$ (primitivno) rekurzivne funkcije. Tada je funkcija $F : \mathbb{N}^k \to \mathbb{N}$ definirana sa:*

$$F(\vec{x}) = \begin{cases} F_1(\vec{x}), & \text{ako vrijedi } R_1(\vec{x}), \\ \vdots & \\ F_n(\vec{x}), & \text{ako vrijedi } R_n(\vec{x}) \end{cases}$$

također (primitivno) rekurzivna.

Dokaz. Očito vrijedi $F(\vec{x}) = F_1(\vec{x}) \cdot \chi_{R_1}(\vec{x}) + \ldots + F_n(\vec{x}) \cdot \chi_{R_n}(\vec{x})$. Sada primjenom propozicije 11.15. (primitivna rekurzivnost funkcije zbrajanja) slijedi tražena tvrdnja. Q.E.D.

Propozicija 12.6. *Neka je F (primitivno) rekurzivna funkcija, a G totalna funkcija koja ima svojstvo da vrijedi $G(\vec{x}) = F(\vec{x})$, osim možda za konačno mnogo $\vec{x}$. Tada je funkcija G također (primitivno) rekurzivna.*

Korolar 12.7. *Neka je R relacija za koju postoji najviše konačno mnogo $\vec{x}$ takvih da vrijedi $R(\vec{x})$. Tada je relacija R primitivno rekurzivna.*

Dokaz. Iz pretpostavke propozicije slijedi da je $\chi_R(\vec{x}) = 0$, osima možda za konačno mnogo $\vec{x}$. Sada primjenom propozicije 12.6. i propozicije 11.14. slijedi tvrdnja.Q.E.D.

Budući da svaki skup možemo promatrati kao unarnu relaciju, tada iz prethodnog korolara direktno slijedi sljedeća tvrdnja.

Korolar 12.8. *Svaki konačan skup je primitivno rekurzivan.*

12.1.3 Ograničene sume i produkti

Propozicija 12.9. *Neka su g, α i β (primitivno) rekurzivne funkcije. Tada su (primitivno) rekurzivne i sljedeće funkcije:*

a) $$f(\vec{x}, y) = \sum_{i=0}^{y} g(\vec{x}, i).$$

b) $$f(\vec{x}, y, z) = \begin{cases} \sum_{i=y}^{z} g(\vec{x}, i), & \textit{ako je } \ y \leq z; \\ \\ 0, & \textit{inače.} \end{cases}$$

c) $$f(\vec{x}) = \begin{cases} \sum_{i=\alpha(\vec{x})}^{\beta(\vec{x})} g(\vec{x}, i), & \textit{ako je } \ \alpha(\vec{x}) \leq \beta(\vec{x}); \\ \\ 0, & \textit{inače.} \end{cases}$$

d) $$f(\vec{x}, y) = \prod_{i=0}^{y} g(\vec{x}, i).$$

e) $$f(\vec{x}, y, z) = \begin{cases} \prod_{i=y}^{z} g(\vec{x}, i), & \text{ako je } \ y \leq z; \\ \\ 1, & \text{inače.} \end{cases}$$

f) $$f(\vec{x}) = \begin{cases} \prod_{i=\alpha(\vec{x})}^{\beta(\vec{x})} g(\vec{x}, i), & \text{ako je } \ \alpha(\vec{x}) \leq \beta(\vec{x}); \\ \\ 1, & \text{inače.} \end{cases}$$

(Uočite da je definirano da su "prazni" produkti jednaki 1).

Dokaz. Za ilustraciju dokazujemo tvrdnje a) i b). Danu funkciju f iz tvrdnje a) moguće je definirati pomoću primitivne rekurzije na sljedeći način:

$$f(\vec{x}, 0) = g(\vec{x}, 0)$$

$$f(\vec{x}, y+1) = f(\vec{x}, y) + g(\vec{x}, y+1).$$

Lako je provjeriti da za funkciju f iz tvrdnje b) vrijedi sljedeće:

$$f(\vec{x}, y, z) = \left(\sum_{i=0}^{z} g(\vec{x}, i) \right) \dot{-} \left(\sum_{i=0}^{y} g(\vec{x}, i) \right) + g(\vec{x}, y) \cdot \overline{sg}(y \dot{-} z) \qquad \text{Q.E.D.}$$

Ako je $R(\vec{x}, y)$ rekurzivna relacija tada općenito relacije $\exists y R(\vec{x}, y)$ i $\forall y R(\vec{x}, y)$ ne moraju biti rekurzivne. To ćemo pokazati kasnije. Nije teško dokazati da je primjenom ograničene kvantifikacije sačuvana rekurzivnost. To ističemo u sljedećoj propoziciji.

Propozicija 12.10. *Neka je R (primitivno) rekurzivna relacija. Tada su (primitivno) rekurzivne i sljedeće relacije: $\exists y < z\ R(\vec{x}, y)$, $\exists y \leq z\ R(\vec{x}, y)$, $\forall y < z\ R(\vec{x}, y)$ i $\forall y \leq z\ R(\vec{x}, y)$.*

Korolar 12.11. *Neka je R (primitivno) rekurzivna relacija. Tada su (primitivno) rekurzivne i sljedeće relacije: $\exists y_{z_1<y<z_2}\ R(\vec{x}, y)$, $\exists y_{z_1\leq y\leq z_2}\ R(\vec{x}, y)$, $\forall y_{z_1<y<z_2}\ R(\vec{x}, y)$ i $\forall y_{z_1\leq y\leq z_2}\ R(\vec{x}, y)$.*

Korolar 12.12. *Neka su α i β (primitivno) rekurzivne funkcije, a R (primitivno) rekurzivna relacija. Tada su (primitivno) rekurzivne i sljedeće relacije: $\exists y_{\alpha(\vec{x})<y<\beta(\vec{x})}\ R(\vec{x}, y)$, $\exists y_{\alpha(\vec{x})\leq y\leq\beta(\vec{x})}\ R(\vec{x}, y)$, $\forall y_{\alpha(\vec{x})<y<\beta(\vec{x})}\ R(\vec{x}, y)$ i $\forall y_{\alpha(\vec{x})\leq y\leq\beta(\vec{x})}\ R(\vec{x}, y)$.*

Propozicija 12.13. *Neka je R (primitivno) rekurzivna relacija. Tada je funkcija $f : \mathbb{N}^{k+1} \to \mathbb{N}$ definirana sa:*

$$f(\vec{x}, z) = \begin{cases} \text{najmanji } y \text{ takav da vrijedi } R(\vec{x}, y) \text{ i } y < z, \\ \text{ako takav } y \text{ postoji;} \\ \\ z, \text{ inače,} \end{cases}$$

(primitivno) rekurzivna. Obično se tako definirana funkcija f označava i sa $\mu y < z\ R(\vec{x}, y)$.

Korolar 12.14. *Neka su α i β (primitivno) rekurzivne funkcije, a R (primitivno) rekurzivna relacija. Neka je funkcija $f : \mathbb{N}^k \to \mathbb{N}$ definirana s:*

$$f(\vec{x}) = \begin{cases} \text{najmanji } y \text{ takav da vrijedi } R(\vec{x}, y) \text{ i} \\ \alpha(\vec{x}) < y < \beta(\vec{x}) \text{ ako takav } y \text{ postoji;} \\ \\ \beta(\vec{x}), \text{ inače} \end{cases}$$

Ovako definiranu funkciju f obično označavamo s $\mu y_{\alpha(\vec{x})<y<\beta(\vec{x})}\ R(\vec{x}, y)$.

Analogno se definiraju i funkcije: $\mu y_{\alpha(\vec{x})\leq y\leq\beta(\vec{x})}\ R(\vec{x}, y)$, $\mu y_{\alpha(\vec{x})<y\leq\beta(\vec{x})}\ R(\vec{x}, y)$ i $\mu y_{\alpha(\vec{x})\leq y<\beta(\vec{x})}\ R(\vec{x}, y)$. Sve te funkcije su (primitivno) rekurzivne.

12.1.4 Zadaci

1. Dokažite da je funkcija $\lfloor x/y \rfloor$ = *najveće cijelo prilikom dijeljenja* x *sa* y, pri čemu radi totalnosti funkcije definiramo $\lfloor x/0 \rfloor = x$, za sve $x \in \mathbb{N}$, primitivno rekuzivna.

   Rješenje. $\lfloor x/y \rfloor = \sum_{i=1}^{x} \overline{sg}(i \cdot y \dot{-} x)$.

   Drugi način:

$$\lfloor x/y \rfloor = \begin{cases} \mu z \leq x (z \cdot y \leq x \ \text{ i } \ (z+1) \cdot y > x), & \text{ako je } \ y \neq 0; \\ x, & \text{ako je } \ y = 0) \end{cases}$$

2. Dokažite da su sljedeće funkcije primitivno rekurzivne:

   a) $res(x, y)$ = *ostatak pri dijeljenju* x *sa* y, pri čemu definiramo $res(x, 0) = x$, za svaki $x \in \mathbb{N}$.

   b)
$$div(x, y) = \begin{cases} 1, & \text{ako } y \text{ dijeli } x; \\ 0, & \text{inače.} \end{cases}$$

   c) $nd(x)$ = *broj djelitelja od* x, pri čemu definiramo $nd(0) = 0$.

   Rješenje.

   a) $res(x, y) = x \dot{-} (y \cdot \lfloor x/y \rfloor)$.

   b) $div(x, y) = \overline{sg}(res(x, y))$.

   c)
$$nd(x) = \begin{cases} \sum_{i=1}^{x} div(x, i), & \text{ako je } \ x \neq 0; \\ 0, & \text{ako je } \ x = 0. \end{cases}$$

3. Dokažite da su sljedeće funkcije primitivno rekurzivne:

   a)
$$\sigma(x) = \begin{cases} \text{suma svih djelitelja broja } \ x, & \text{ako je } \ x \neq 0; \\ 0, & \text{ako je } \ x = 0. \end{cases}$$

   b)
$$f(n, m) = \begin{cases} \binom{n}{m}, & \text{ako je } \ n \geq m; \\ 0, & \text{inače.} \end{cases}$$

Rješenje.

a)
$$\sigma(x) = \begin{cases} \sum_{i=1}^{x} i \cdot div(x,i), & \text{ako je } x \neq 0; \\ 0, & x = 0. \end{cases}$$

b) Vrijedi:
$$f(n,m) = \begin{cases} \mu z \leq n!\big(z \cdot m! \cdot (n-m)! = n!\big), & \text{ako je } n \geq m; \\ 0, & \text{inače.} \end{cases}$$

4. Dokažite da je funkcija $p : \mathbb{N} \to \mathbb{N}$ koja svakom $n \in \mathbb{N}$ pridružuje n–ti po redu prosti broj, RAM–izračunljiva.
Rješenje. Znamo da je svaka rekurzivna funkcija RAM-izračunljiva, pa je dovoljno dokazati da je funkcija p rekurzivna. Označimo s Pr unarnu relaciju na $\mathbb{N}$ koja je definirana s
$$Pr(x) \quad \Leftrightarrow \quad x \text{ je prosti broj.}$$
Primitivna rekurzivnost relacije Pr slijedi iz ekvivalencije
$$Pr(x) \text{ ako i samo ako } nd(x) = 2$$
(primitivna rekurzivnost funkcije nd je dokazana u nekom prethodnom zadatku). Očito vrijedi:
$$\begin{aligned} p(0) &= 2 \\ p(n+1) &= \mu\, y[y > p(n) \ \wedge \ Pr(y)], \end{aligned}$$
pa je funkcija p rekurzivna, a onda i RAM–izračunljiva.

5. Označimo s $\pi(x)$ broj svih prostih djelitelja broja x, pri čemu definiramo $\pi(0) = 0$. Dokažite da je funkcija π primitivno rekurzivna.
Rješenje. Lako je provjeriti da vrijedi $\pi(x) = \sum_{i=0}^{x} \chi_{Pr}(i) \cdot div(x,i)$. (Relacija Pr je definirana u rješenju prethodnog zadatka).

6. Za svaki $n \in \mathbb{N}$, $n \geq 2$, definirane su funkcije min i max s:
$$min(x_1, \ldots, x_n) \ = \ \textit{najmanji element skupa } \ \{x_1, \ldots, x_n\};$$
$$max(x_1, \ldots, x_n) \ = \textit{najveći element skupa } \ \{x_1, \ldots, x_n\}.$$
Dokažite da su za svaki $n \in \mathbb{N}$ funkcije min i max primitivno rekurzivne.
Rješenje. Za $n = 2$ za funkciju min očito vrijedi $min(x,y) = x \dot{-} (x \dot{-} y)$, pa je ta funkcija primitivno rekurzivna. Primitivna rekurzivnost n–mjesne funkcije min lako se dokaže indukcijom po n. Uočite da vrijedi $max(x,y) = x + y - min(x,y)$.

7. Neka je za sve $x \in \mathbb{N}$ s $\varphi(x)$ označen broj svih prirodnih brojeva manjih od x koji su relativno prosti s x (uočite da je posebno $\varphi(0) = 0$). Dokažite da je funkcija φ primitivno rekurzivna. Funkcija φ se naziva Eulerova funkcija.
Rješenje. Označimo s rp dvomjesnu relaciju definiranu s:

$$rp(x,y) \;\Leftrightarrow\; \text{brojevi } \; x \; \text{ i } \; y \; \text{ su relativno prosti ili } x \cdot y = 0.$$

Budući da vrijedi:

$$\chi_{rp}(x,y) = \overline{sg} \sum_{i=2}^{min(x,y)} div(x,i) \cdot div(y,i),$$

tada je relacija rp primitivno rekurzivna. Sada iz $\varphi(x) = \sum_{i=1}^{x} \chi_{rp}(x,i)$ slijedi primitivna rekurzivnost Eulerove funkcije.

8. Neka je za $x \neq 0$ definirano $f(x) =$ *broj eksponenata različitih od nule u rastavu broja x na proste faktore*, te neka je $f(0) = 0$. Dokažite da je f primitivno rekurzivna funkcija.

9. Dokažite da je funkcija $x \mapsto \lfloor \sqrt{x} \rfloor$ primitivno rekurzivna. Je li funkcija $f : \mathbb{N} \to \mathbb{N}$ definirana s

$$f(x) = \begin{cases} \sqrt{x}, & \text{ako je } \; \sqrt{x} \in \mathbb{N}; \\ 0, & \text{inače,} \end{cases}$$

primitivno rekurzivna?
Rješenje. $\lfloor \sqrt{x} \rfloor = \mu y \leq x \big(y^2 \leq x \; \wedge (y+1)^2 > x \big)$.

10. Dokažite da je funkcija $\lfloor \sqrt[y]{x} \rfloor$ primitivno rekurzivna, gdje smo definirali $\lfloor \sqrt[0]{x} \rfloor = x$, za sve $x \in \mathbb{N}$.

11. Funkcija $f : \mathbb{N} \to \mathbb{N}$ definirana je s $f(x) = \lfloor \log_2 x \rfloor$. Dokažite da je funkcija f RAM–izračunljiva.

12. Neka je funkcija $f : \mathbb{N} \to \mathbb{N}$ definirana s:

$$f(x) = \Big\lfloor \; \sqrt[7]{\log_2 \tfrac{2x+1}{2} + 3} \; \Big\rfloor$$

Dokažite da je funkcija f primitivno rekurzivna.
Rješenje. Očito vrijedi

$$f(x) = (\mu y < x)(2^{y^7 \dot{-} 3} \cdot 2 \dot{-} 1 \leq 2x < 2^{(y+1)^7 \dot{-} 3} \cdot 2 \dot{-} 1),$$

iz čega slijedi primitivna rekurzivnost funkcije f.

13. Dokažite da je sljedeća funkcija f primitivno rekurzivna:

$$f(x_1, \ldots, x_n) = 2^{max(x_2,\ldots,x_n)} \ + \ \lfloor \log \ min(x_1 + 1, \ldots, x_{n-1} + 1) \rfloor.$$

14. Dokažite da su sljedeće relacije primitivno rekurzivne:

    a) *x je savršen broj*;

    b) *znamenka jedinica broja $3x$ je jednaka 7*;

    c) *x je kvadrat prostog broja*;

    d) *x je djeljiv s 2, 3 ili 5, i s niti jednim drugim prostim brojem*;

    e) *x u svom zapisu s bazom 3 nema znamenku 1.*

15. a) Relacija $P(x)$ je zadana s $(\exists n \in \mathbb{N})(x = 1 + 2 + \ldots + n)$. Je li relacija P primitivno rekurzivna?

    b) Relacija $P(x)$ je zadana s $(\exists n \in \mathbb{N})(x = 1^2 + 2^2 + \ldots + n^2)$. Dokažite da je relacija P primitivno rekurzivna.

    Rješenje b). Označimo s f funkciju definiranu s:

$$f(0) = 0$$

$$f(n + 1) = f(n) + (n + 1)^2.$$

    Očito je funkcija f primitivno rekurzivna. Karakteristična funkcija relacije P je dana sa $\chi_P(x) = \sum_{i=0}^{x} \chi_=(f(i), x)$. Iz ovog posljednjeg vidimo da je relacija P primitivno rekurzivna.

16. Dokažite da su sljedeći skupovi rekurzivni:

    a) prazan skup;

    b) skup svih prirodnih brojeva $\mathbb{N}$;

    c) svaki podskup od $\mathbb{N}$ čiji je komplement konačan;

    d) skup svih parnih;

    e) skup svih neparnih prirodnih brojeva.

17. Neka je f monotono padajuća funkcija. Dokažite da je tada slika od f rekurzivan skup, te da je f primitivno rekurzivna funkcija.

18. Dokažite da je svaka funkcija s konačnom domenom parcijalno rekurzivna.
    Rješenje. Neka je f n-mjesna, te neka je domena

$$Dom(f) = \{(a_{11}, \ldots, a_{1n}), \ldots, (a_{m1}, \ldots, a_{mn})\}.$$

Označimo $f(a_{i1}, \ldots, a_{in}) = b_i$, za sve $i = 1, \ldots, m$. Tada imamo:

$$f(\vec{x}) \simeq \sum_{i=1}^{m} b_i \cdot \overline{sg}\Big(|x_1 - a_{i1}| + \ldots + |x_n - a_{in}|\Big) + \mu z\Big(z + \prod_{j=1}^{m}(|x_1 - a_{j1}| + \ldots + |x_n - a_{jn}|) = 0\Big).$$

Iz ovog posljednjeg slijedi parcijalna rekurzivnost funkcije f.

19. Neka je $a_0, a_1 a_2 \ldots$ decimalni prikaz broja $\sqrt[3]{5}$. Dokažite da je funkcija $n \mapsto a_n$ primitivno rekurzivna.
Rješenje. Neka je $n \geq 1$. Iz

$$a_0, a_1 \ldots a_n < \sqrt[3]{5} < a_0, a_1 \ldots a_n + 10^{-n}$$

slijedi da je

$$(\overline{a_0 a_1 \ldots a_n})^3 < 5 \cdot 10^{3n} < (\overline{a_0 a_1 \ldots a_n} + 1)^3$$

pa je $\overline{a_0 a_1 \ldots a_n}$ najmanji prirodan broj k takav da je $(k+1)^3 > 5 \cdot 10^{3n}$. Stoga je a_n ostatak pri dijeljenju s 10 broja

$$\mu k((k+1)^3 > 5 \cdot 10^{3n}) = \mu k \leq 5 \cdot 10^{3n}((k+1)^3 > 5 \cdot 10^{3n})$$

iz čega slijedi da je funkcija $n \mapsto a_n$ primitivno rekurzivna.

20. Neka je $f : \{0, 3, 6, 9, \ldots\} \to \mathbb{N}$ funkcija definirana sa $f(i) = i$. Dokažite da je f parcijalno rekurzivna funkcija.
Rješenje. Lako je vidjeti da vrijedi $f(x) = [\mu y(|3y - x| = 0)] \cdot 3$.

21. Neka je $S = \{(x, y) : x < y\}$, te neka je $f : S \to \mathbb{N}$ funkcija definirana sa $f(x, y) = y - x - 1$. Dokažite da je f parcijalno rekurzivna funkcija.
Rješenje. Lako je vidjeti da vrijedi $f(x) = \mu z|z + x + 1 - y|$.

12.2 Kodiranje

12.2.1 Kodiranje konačnih nizova

Neka je I neka klasa, tj. skup koji nije podskup od $\mathbb{N}$. Funkciju $F : I \to \mathbb{N}$ ćemo smatrati **kodiranjem** ako vrijedi sljedeće:

a) funkcija F je **injekcija**;

b) postoji **algoritam koji određuje** $F(i)$, za svaki $i \in I$;

c) postoji algoritam koji za svaki $n \in \mathbb{N}$ određuje postoji li $i \in I$ tako da vrijedi $F(i) = n$. Ako takav i postoji tada ga algoritam efektivno određuje (**dekodiranje**).

Mi se nećemo baviti proučavanjem općenitih funkcija kodiranja. Za naša daljnja proučavanja posebno će biti značajno **kodiranje skupa svih konačnih nizova prirodnih brojeva**. U tu svrhu označimo s p_k $k-$ti prim broj, tj. neka je $p_0,\ p_1,\ p_2,\ \ldots$ rastući niz svih prim brojeva. (To znači da je $p_0 = 2,\ p_1 = 3,\ p_2 = 5,\ \ldots$).

Neka je $k \in \mathbb{N}$ proizvoljan, te $x_0,\ x_1,\ \ldots,\ x_{k-1}$ proizvoljan niz prirodnih brojeva duljine k. Tada danom nizu pridružujemo broj

$$\mathbf{p_0^{x_0+1}} \cdot \mathbf{p_1^{x_1+1}} \ \cdots\ \mathbf{p_{k-1}^{x_{k-1}+1}}$$

koji nazivamo **kod** konačnog niza $x_0,\ x_1,\ \ldots,\ x_{k-1}$. Po definiciji praznom nizu pridružujemo broj 0.

Primjer 12.15. *U ovom primjeru kod niza $x_0,\ x_1,\ \ldots,\ x_{k-1}$ označavamo sa $\langle x_0,\ x_1,\ \ldots,\ x_{k-1}\rangle$. Navodimo kodove nekih konačnih nizova.*

- $\langle 0,\ 0\rangle \ = \ 2^{0+1} \cdot 3^{0+1} \ = \ 6$
- $\langle 1,\ 1,\ 1\rangle \ = \ 2^{1+1} \cdot 3^{1+1} \cdot 5^{1+1} = \ 4 \cdot 9 \cdot 25$
- $\langle 0,\ 1,\ 2,\ 3\rangle \ = \ 2^{0+1} \cdot 3^{1+1} \cdot 5^{2+1} \cdot 7^{3+1} = \ 2 \cdot 9 \cdot 125 \cdot 7^4$

Iz teorema o jedinstvenoj dekompoziciji svakog prirodnog broja na prim faktore slijedi da različiti nizovi imaju različite kodove, tj. da je upravo definirana funkcija injekcija.

Napomena 12.16. *Želimo naglasiti da kod konačnog niza $x_0,\ x_1,\ \ldots,\ x_{k-1}$ nismo mogli definirati kao $p_0^{x_0} \cdot \ldots p_{k-1}^{x_{k-1}}$. Tada bi primjerice nizovima $2,\ 0$ i $2,\ 0,\ 0,\ 0,\ 0$ bio pridružen isti kod, tj. funkcija kodiranja ne bi bila injekcija.*

Sada nam je cilj dokazati da je upravo definirano preslikavanje kodiranje, tj. da je dano preslikavanje i dekodiranje efektivno. U tu svrhu redom definiramo funkcije i relacije, te dokazujemo da su rekurzivne.

Sa $\mathbf{Div}(x, y)$ označimo relaciju na $\mathbb{N}$ koja je definirana na sljedeći način:

$$Div(x, y) \text{ ako i samo ako } \text{"}broj\ x\ je\ djeljiv\ brojem\ y\text{"}.$$

Očito vrijedi: $Div(x, y)$ ako i samo ako $y \neq 0 \wedge \exists z(x = y \cdot z)$. No, iz toga ne možemo zaključiti da je relacija Div rekurzivna, jer se radi o neograničenoj kvantifikaciji $\exists z$. To možemo popraviti na sljedeći način:

$$Div(x, y) \text{ ako i samo ako } y \neq 0 \ \wedge \ (\exists z \leq x)(x = y \cdot z),$$

iz čega slijedi primitivna rekurzivnost relacije Div.

Označimo sa $\mathbf{Pr}(x)$ relaciju "*broj x je prim broj*". Primjenom relacije Div dobivamo da vrijedi:

$$Pr(x) \text{ ako i samo ako } x > 1 \ \wedge \ (\forall y < x)(y > 1 \rightarrow \neg Div(x, y)).$$

Primjenom primitivne rekurzivnosti relacije Div i prije dokazanih propozicija slijedi da je relacija Pr primitivno rekurzivna.

Primijetimo da tada vrijede sljedeće jednakosti:

$$p_0 = 2 \ \text{ i } \ p_{i+1} = \mu x \leq p_i! + 1(Pr(x) \ \wedge \ x > p_i).$$

No, to je upravo definicija funkcije $n \mapsto p_n$ pomoću primitivne rekurzije. To znači da je funkcija $n \mapsto p_n$, tj. funkcija koja prirodnom broju n pridružuje n–ti po redu prim broj, primitivno rekurzivna. Za svaki $k \in \mathbb{N}$, pri čemu je $k > 0$, definiramo k–mjesnu totalnu funkciju $\langle . \rangle$ na $\mathbb{N}^k$ na sljedeći način:

$$\langle x_0, \ldots, x_{k-1} \rangle = p_0^{x_0+1} \ \cdots \ p_{k-1}^{x_{k-1}+1}.$$

Zatim, definiramo da je $\langle\rangle = 0$ (tj. kod praznog niza je nula). Iz prethodnih rezultata slijedi da je funkcija kodiranja primitivno rekurzivna. (Uočite da koristimo istu oznaku za funkcije kodiranja različitih mjesnosti!)

Sada nam je cilj dokazati da je "dekodiranje" efektivan postupak, tj. da postoji algoritam koji za svaki $n \in \mathbb{N}$ određuje postoje li prirodni brojevi $k, x_0, \ldots, x_{k-1}$ tako da vrijedi $n = \langle x_0, \ldots, x_{k-1} \rangle$. U tu svrhu redom promatramo sljedeće relacije i funkcije, te dokazujemo da su rekurzivne. Neka je sa $\mathbf{exp}(x, i)$ označen eksponent prim broja p_i u rastavu broja x na prim faktore. Budući da želimo da funkcija exp bude totalna po dogovoru stavljamo $exp(0, i) = 0$, za sve $i \in \mathbb{N}$. Očito vrijedi:

$$exp(x, i) = \begin{cases} \mu y < x \ (Div(x, p_i^y) \wedge \neg Div(x, p_i^{y+1})), & \text{ako je } \ x \neq 0; \\ 0, & \text{ako je } \ x = 0, \end{cases}$$

iz čega slijedi da je funkcija *exp* primitivno rekurzivna. Definiramo jednomjesnu totalnu funkciju *lh* (eng. length) i dvomjesnu totalnu funkciju ( ) na sljedeći način:

$$lh(x) = \mu i < x(exp(x,i) = 0)$$

$$(x)_i = exp(x,i) \dot{-} 1.$$

Ako je $x = \langle x_0, x_1, \ldots, x_{k-1} \rangle$ tada očito vrijedi:

$$lh(x) = k,$$
$$(x)_i = \begin{cases} x_i, & \text{za sve } \ i < k; \\ \\ 0, & \text{za sve } \ i \geq k. \end{cases}$$

Očito su funkcije lh i $(.)$ primitivno rekurzivne.

Označimo sa *Seq* jednomjesnu relaciju definiranu sa:

$Seq(x)$ ako i samo ako
"*x je kod nekog konačnog niza prirodnih brojeva*

Budući da očito vrijedi:

$Seq(x)$ ako i samo ako

$$(x = 0 \ \vee \ (x \neq 0 \ \wedge \ (\forall i < x)(Div(x, p_i) \rightarrow i < lh(x))))$$

tada je relacija *Seq* primitivno rekurzivna.

Važna binarna operacija na konačnim nizovima je **konkatenacija**. To je dvomjesna funkcija na $\mathbb{N}$ koju označavamo sa $*$ i ima sljedeće svojstvo. Ako je $x = \langle x_0, \ldots, x_{k-1} \rangle$ i $y = \langle y_0, \ldots, y_{l-1} \rangle$ tada je $x * y$ kod niza $x_0, \ldots, x_{k-1}, y_0, \ldots, y_{l-1}$. Nije teško vidjeti da vrijedi

$$\begin{aligned} x * y \ &= \ \mu z(Seq(z) \ \wedge \ lh(z) = lh(x) + lh(y) \wedge \\ &\quad (\forall i < lh(x))((z)_i = (x)_i) \wedge \\ &\quad (\forall i < lh(y))((z)_{lh(x)+i} = (y)_i)) \\ &= \ \prod_{i<lh(x)} p_i^{(x)_i} \cdot \prod_{i<lh(y)} p_{lh(x)+i}^{(y)_i}. \end{aligned}$$

Iz toga slijedi da je funkcija konkatenacije primitivno rekurzivna.

12.2.2 Kodiranje RAM–stroja

Dokazali smo da je svaka parcijalno rekurzivna funkcija RAM-izračunljiva. Sada ćemo dokazati obrat. U tu svrhu ćemo "kodirati" RAM–stroj. Prvo svakoj instrukciji RAM-programa pridružujemo kod na sljedeći način:

$$\begin{aligned} \text{INC } \mathcal{R}_i &\mapsto \langle 0, i\rangle \\ \text{DEC } \mathcal{R}_i, m &\mapsto \langle 1, i, m\rangle \\ \text{GO TO } m &\mapsto \langle 2, m\rangle \\ \text{STOP} &\mapsto \langle 3\rangle \end{aligned}$$

Ako je P program za RAM–stroj koji sadrži n instrukcija čiji su pripadni kodovi redom $y_1, \ldots, y_n$ tada po definiciji programu P pridružujemo kod $\langle y_1, \ldots, y_n\rangle$. Obično ćemo kod programa P kratko označavati sa e. Označimo sa **Ins** jednomjesnu unarnu relaciju na $\mathbb{N}$ koja je definirana sa:

$Ins(x)$ ako i samo ako "*x je kod neke instrukcije za RAM-stroj*".

Lako je vidjeti da vrijedi:

$$Ins(x) \text{ ako i samo ako } x = \langle 0, (x)_1\rangle \vee x = \langle 1, (x)_1, (x)_2\rangle \vee x = \langle 2, (x)_1\rangle \vee x = \langle 3\rangle.$$

Iz toga slijedi da je relacija Ins primitivno rekurzivna. Označimo sa **Prog** jednomjesnu unarnu relaciju na $\mathbb{N}$ koja je definirana sa:

$Prog(x)$ ako i samo ako "*x je kod nekog programa za RAM-stroj*".

Lako je vidjeti da vrijedi:

$$\begin{aligned} Prog(x) \text{ ako i samo ako } & Seq(x) \wedge (\forall i < lh(x))[Ins((x)_i) \wedge \\ & (((x)_i)_0 = 1 \to ((x)_i)_2 < lh(x)) \wedge \\ & (((x)_i)_0 = 2 \to ((x)_i)_1 < lh(x))]. \end{aligned}$$

Iz posljednje ekvivalencije slijedi da je relacija $Prog$ primitivno rekurzivna. Objasnimo ukratko prethodno navedenu ekvivalenciju. Uvjet $Seq(x)$ znači da je x kod nekog konačnog niza prirodnih brojeva. Za svaki i, koji su strogo manji od duljine niza čiji je kod x, vrijedi redom sljedeće:

a) broj $(x)_i$ je kod neke instrukcije za RAM-program;

b) ako je $((x)_i)_0 = 1$, tj. $(x)_i$ je kod neke instrukcije oblika DEC $\mathcal{R}_i, n$, tada n mora biti strogo manji od duljine niza $x_0, \ldots, x_{k-1}$ (tj. u programu mora postojati instrukcija s rednim brojem n);

c) analogno kao b), ali za instrukciju oblika GO TO m.

Ako P–izračunavanje sa $\vec{x}$ stane nakon m koraka tada sa r_i označimo kod registara stroja nakon i–tog koraka, za $i = 0, \ldots, m$. Tada broj $r = \langle r_0, \ldots, r_m \rangle$ nazivamo **kod P–izračunavanja sa $\vec{x}$**. Ako P–izračunavanje sa $\vec{x}$ stane izlazni rezultat označimo sa $\mathbf{U(r)}$, tj. $U(r)$ je broj zapisan u registru $\mathcal{R}_0$ nakon m–tog koraka. Uočite da vrijedi $U(r) = \left((r)_{lh(r)\dot{-}1}\right)_0$. To znači da U možemo promatrati kao funkciju definiranu na skupu $\mathbb{N}$. Štoviše, iz prethodnog identiteta slijedi da je U primitivno rekurzivna funkcija.

Za potpuno kodiranje rada nekog RAM–stroja trebalo bi još definirati funkcije koje opisuju rad brojača i upisa u registre. To ovjde nećemo raditi. Detalje možete vidjeti u nastavnom materijalu za kolegij *Izračunljivost* koji sam niz godina predavao na PMF–u. Spomenuti nastavni materijal se nalazi na mrežnoj adresi:
https://www.math.pmf.unizg.hr/sites/default/files/pictures/izn-skripta-2009.pdf.

Za svaki $k \in \mathbb{N}$ sa $\mathbf{T}_k(e, \vec{x}, y)$ označavamo relaciju definiranu sa:

> "*e je kod programa P, $\vec{x}$ je uređena k–torka prirodnih brojeva a y je kod P–izračunavanja sa $\vec{x}$.*"

Može se pokazati da je za svaki $k \in \mathbb{N}$ relacija T_k primitivno rekurzivna.

Definicija 12.17. *Neka su $e, k \in \mathbb{N}$ i $\vec{x} \in \mathbb{N}^k$. Ako je e kod nekog programa P za RAM–stroj s k ulaznih podataka, te ako P–izračunavanje sa $\vec{x}$ kao ulaznim podacima stane, tada sa $\{e\}^k(\vec{x})$ označavamo izlazni rezultat.*

U svim ostalim slučajevima smatramo da izraz $\{e\}^k(\vec{x})$ nije definiran. Obično ćemo umjesto $\{e\}^k$ pisati samo $\{e\}$ kada neće biti nužno isticati mjesnost.

12.2.3 Zadaci

1. Odredite $\{128\}(127)$ i $\{63\}(64)$.
   Rješenje. Odredimo program čiji je kod 128. Budući da je $128 = 2^7 = 2^{6+1}$ tada se traženi program za RAM–stroj sastoji samo od jedne instrukcije. Kod te jedne instrukcije je 6. Budući da je $6 = 2^{1+0} \cdot 3^{1+0}$, tj. $6 = \langle 0, 0 \rangle$ tada je 6 kod instrukcije INC $\mathcal{R}_0$. Dakle, 128 je kod programa: 1. INC $\mathcal{R}_0$. Očito taj program izračunava funkciju f definiranu sa $f(x) = 1$. Dakle, 128 je indeks funkcije f. Iz toga zaključujemo da je $\{128\}(127) = 1$.

   Broj 63 nije kod niti jednog programa jer je neparan. To znači da funkcija $\{63\}$ nije definirana niti za jedna prirodan broj. Posebno, izraz $\{63\}(64)$ nije definiran.

2. Neka je $n = 2^{2700\cdot 125+1} \cdot 3^7 \cdot 5^{73} \cdot 7^{19}$. Odredite $\{n\}(125, 332, 46)$.
Rješenje. Uočimo: $n = \langle 2700 \cdot 125,\ 6,\ 72,\ 18\rangle$. Zatim, imamo:

$$2700 \cdot 125 = \ 27 \cdot 100 \cdot 125 \ = \ 3^3 \cdot 2 \cdot 5 \cdot 2 \cdot 5 \cdot 5^3 =$$

$$= \ 2^2 \cdot 3^3 \cdot 5^5 \ = \ 2^{1+1} \cdot 3^{2+1} \cdot 5^{4+1} \ = \ \langle 1, 2, 4\rangle$$

Budući da je $6 = 2 \cdot 3 = \langle 0, 0\rangle$, $\quad 72 = 2^3 \cdot 3^2 = \langle 2, 1\rangle$ i $18 = 2 \cdot 3^2 = \langle 0, 1\rangle$, tada imamo da je n kod sljedećeg RAM–programa:

1. DEC $\mathcal{R}_2, 4$
2. INC $\mathcal{R}_0$
3. GO TO 1
4. INC $\mathcal{R}_1$

Sada je lako vidjeti da vrijedi: $\{n\}(125, 332, 46) = 332$.

3. Neka je $n = 2^{181} \cdot 3^7$. Odredite domenu jednomjesne funkcije $\{n\}$.

4. Dokažite da postoji parcijalno rekurzivna funkcija f koja se ne može proširiti do rekurzivne funkcije.
Rješenje. Neka je funkcija f definirana sa $f(x) \simeq \{x\}(x) + 1$. Funkcija f je parcijalno rekurzivna. Pretpostavimo da postoji rekurzivna funkcija g koja je proširenje funkcije f. Neka je e indeks funkcije g. Iz $g(x) \simeq \{e\}(x)$ slijedi da je $\{e\}(e)$ definirano pa je f definirana u točki e te imamo $\{e\}(e) + 1 = f(e) = g(e) = \{e\}(e)$, što je kontradikcija.

5. Dokažite da za svaku rekurzivnu funkciju postoji definicija pomoću funkcija koje su sve totalne.

Poglavlje 13

Trinaesto predavanje – izračunljivost

13.1 Kleenijev teorem o normalnoj formi

Ako P–izračunavanje sa $\vec{x}$ stane tada vrijedi

$$U(r) = \{e\}(\vec{x}) \qquad \text{i} \qquad r = \mu y T_k(e, \vec{x}, y)$$

Ako pak P–izračunavanje sa $\vec{x}$ ne stane tada ne postoji $y \in \mathbb{N}$ tako da bi vrijedilo $T_k(e, \vec{x}, y)$, a to znači da izraz $U(\mu y T_k(e, \vec{x}, y))$ nije definiran.

Iz ovih razmatranja slijedi da za sve $e, k \in \mathbb{N}$ i $\vec{x} \in \mathbb{N}^k$ vrijedi

$$\{e\}(\vec{x}) \simeq U(\mu y T_k(e, \vec{x}, y)).$$

Time smo dokazali sljedeći važan teorem.

Teorem 13.1. (Kleenijev teorem o normalnoj formi)
Postoji primitivno rekurzivna funkcija U, i za svaki $k \geq 1$ postoji primitivno rekurzivna relacija T_k tako da za svaku k–mjesnu parcijalno rekurzivnu funkciju φ postoji $e \in \mathbb{N}$ tako da vrijede sljedeće tvrdnje:

a) $\varphi(\vec{x}) \downarrow$ *ako i samo ako postoji y tako da vrijedi* $T_k(e, \vec{x}, y)$;
b) $\varphi(\vec{x}) \simeq U(\mu y T_k(e, \vec{x}, y))$

Kleenijev teorem smo dokazali prethodnim razmatranjima. No, ipak ukratko ćemo ponoviti skicu tog dokaza. Neka je φ neka k-mjesna parcijalno rekurzivna funkcija. Tada je ta funkcija i RAM–izračunljiva. Neka je P neki program za RAM-stroj koji izračunava funkciju φ, tj. za sve $\vec{x} \in \mathbb{N}^k$ vrijedi:

> $\varphi(x) \downarrow$ ako i samo ako P–izračunavanje sa $\vec{x}$ stane, te je tada u registru $\mathcal{R}_0$ zapisan broj $\varphi(\vec{x})$.

Označimo sa e kod programa P. Ako stroj stane, označimo sa y_0 kod P–izračunavanja sa $\vec{x}$. Po definiciji relacije T_k tada imamo da vrijedi $T_k(e, \vec{x}, y_0)$. To znači da postoji $y \in \mathbb{N}$ tako da vrijedi $T_k(e, \vec{x}, y)$, odnosno funkcija U je definirana za $\mu y T_k(e, \vec{x}, y)$. No, sjetimo se da je $U(r)$ upravo broj zapisan u registru $\mathcal{R}_0$ na kraju izvršenja programa (ako stroj stane!). Dakle, ako stroj stane tada vrijedi $\varphi(\vec{x}) = U(\mu y T_k(e, \vec{x}, y))$. Lako je vidjeti da vrijedi $\varphi(\vec{x}) \uparrow$ ako i samo ako ne postoji y takav da vrijedi $T_k(e, \vec{x}, y)$.

13.1.1 Posljedice Kleenijevog teorema

Definicija 13.2. *Neka je $\varphi : S \subseteq \mathbb{N}^k \to \mathbb{N}$ neka funkcija. Kažemo da za funkciju φ postoji* **indeks** *ako postoji $e \in \mathbb{N}$ takav da za sve $\vec{x} \in \mathbb{N}^k$ vrijedi:*

$$\varphi(\vec{x}) \simeq \{e\}(\vec{x}).$$

Teorem 13.3. *Funkcija $\varphi : S \subseteq \mathbb{N}^k \to \mathbb{N}$ je parcijalno rekurzivna ako i samo ako postoji indeks za φ.*

Napomena 13.4. *Za sve prirodne brojeve e i k ima smisla napisati $\{e\}^k$, jer iz Kleenijevog teorema imamo da vrijedi sljedeća jednakost:*

$$\{e\}^k(\vec{x}) \simeq U(\mu y T_k(e, \vec{x}, y).$$

Dakle, za sve $e, k \in \mathbb{N}$ je definirana funkcija $\{e\}^k$. Ako e nije kod nekog programa za RAM–stroj tada za sve $k \in \mathbb{N}$ i $\vec{x} \in \mathbb{N}^k$ vrijedi $\{e\}^k(\vec{x}) \uparrow$. Posebno to znači da ako e nije kod nekog programa za RAM–stroj tada ne možemo govoriti o fiksnoj mjesnosti funkcije.

Primjerice, broj 5 nije kod niti jednog RAM–programa, pa na primjer sljedeći izrazi nisu definirani:

$$\{5\}^2(1,2), \qquad \{5\}^4(1,2,34,5) \quad \text{i} \quad \{5\}^6(6,78,9,567,0,1).$$

Uočite još da za svaki $e \in \mathbb{N}$ postoji $e' \in \mathbb{N}$ tako da je e' kod nekog programa za RAM–stroj, te vrijedi $\{e\} \simeq \{e'\}$.

Teorem 13.5. *Funkcija φ je RAM–izračunljiva ako i samo ako je parcijalno rekurzivna.*

Bili smo dokazali da je funkcija definirana po slučajevima pomoću rekurzivnih funkcija i rekurzivnih disjunktnih relacija, također rekurzivna. Sada ćemo primjenom indeksa navesti analogni rezultat za parcijalno rekurzivne funkcije.

Teorem 13.6. (Definicija funkcije po slučajevima – verzija 2) *Neka su* $R_1, \ldots,$ R_n *rekurzivne relacije koje imaju svojstvo da za svaki* $\vec{x} \in \mathbb{N}^k$ *postoji najviše jedan* $i \in \{1, \ldots, n\}$ *za kojeg vrijedi* $R_i(\vec{x})$*. Neka su* $F_1, \ldots, F_n$ *neke* k*–mjesne parcijalno rekurzivne funkcije. Funkciju* F *definiramo po slučajevima ovako:*

$$F(\vec{x}) \simeq \begin{cases} F_1(\vec{x}), & \text{ako vrijedi } R_1(\vec{x}); \\ \vdots & \\ F_n(\vec{x}), & \text{ako vrijedi } R_n(\vec{x}). \end{cases}$$

Tada je funkcija F *parcijalno rekurzivna.*

Teorem 13.7. *Za svaku parcijalno rekurzivnu funkciju* φ *postoji definicija u kojoj se* μ*-operator pojavljuje najviše jednom.*

13.1.2 Teorem rekurzije

Ako je $G : S \subseteq \mathbb{N}^{k+1} \to \mathbb{N}$ neka parcijalno rekurzivna funkcija, tada je svaka od sljedećih funkcija također parcijalno rekurzivna:

$$\begin{gathered} \vec{x} \mapsto G(0, \vec{x}) \\ \vec{x} \mapsto G(1, \vec{x}) \\ \vec{x} \mapsto G(2, \vec{x}) \\ \vdots \end{gathered}$$

Postavlja se pitanje postoji li $e \in \mathbb{N}$ tako da je e indeks funkcije $\vec{x} \mapsto G(e, \vec{x})$. Pozitivan odgovor na to pitanje daje sljedeći teorem.

Teorem 13.8. (Teorem rekurzije) *Neka je* G *neka* $(k+1)$*–mjesna parcijalno rekurzivna funkcija. Tada postoji* $e \in \mathbb{N}$ *tako da za sve* $\vec{x} \in \mathbb{N}^k$ *vrijedi:*

$$\{e\}^k(\vec{x}) \simeq G(e, \vec{x}).$$

Neposredna posljedica teorema rekurzije je sljedeći teorem.

Teorem 13.9. (Teorem o fiksnoj točki) *Za svaku unarnu parcijalnu rekurzivnu funkciju* F *postoji* $e \in \mathbb{N}$ *tako da vrijedi:*

$$\{e\} \simeq \{F(e)\}.$$

Zadaci. (1. primjena teorema rekurzije – domene)

1. Dokažite da postoji $e \in \mathbb{N}$ tako da je domena funkcije $\{e\}^1$ jednaka upravo skupu $\{e\}$.
   Rješenje. Definiramo prvo dvomjesnu relaciju R sa:

   $$xRy \text{ ako i samo ako } x = y.$$

   Očito je relacija R rekurzivna. Zatim definiramo dvomjesnu funkciju G ovako:

   $$G(x, y) \simeq \mu z[\overline{sg}(\chi_R(x, y)) + z \simeq 0].$$

   Očito je G parcijalno rekurzivna funkcija pa iz teorema rekurzije slijedi da postoji $e \in \mathbb{N}$ tako da vrijedi $\{e\}^1(y) \simeq G(e, y)$. Lako je vidjeti da je domena funkcije $\{e\}$ jednaka skupu $\{e\}$.

2. Dokažite da postoji $e \in \mathbb{N}$ takav da je domena funkcije $\{e\}$ jednaka $\{e, e + 1\}$.
   Rješenje. Prvo definiramo dvomjesnu relaciju R na sljedeć način:

   $$xRy \text{ ako i samo ako } x = y \ \vee x + 1 = y.$$

   Očito je relacija R rekurzivna. Zatim definiramo dvomjesnu funkciju G ovako:

   $$G(x, y) \simeq \mu z[\overline{sg}(\chi_R(x, y)) + z \simeq 0].$$

   Očito je G parcijalno rekurzivna funkcija, pa iz teorema rekurzije slijedi da postoji $e \in \mathbb{N}$ tako da za svaki $y \in \mathbb{N}$ vrijedi $\{e\}(y) \simeq G(e, y)$. Budući da je $Dom(G) = \{(y, y),\ (y, y + 1) : y \in \mathbb{N}\}$, tada je $Dom(\{e\}) = \{e, e + 1\}$.

3. Dokažite da postoji $e \in \mathbb{N}$ tako da je domena funkcije $\{e\}^1$ jednaka skupu $\mathbb{N}\backslash\{e\}$.
   Rješenje. Definiramo prvo dvomjesnu relaciju R ovako:

   $$xRy \text{ ako i samo ako } \neg(x = y).$$

   Očito je relacija R rekurzivna. Zatim definiramo dvomjesnu funkciju G ovako:

   $$G(x, y) \simeq \mu z[\overline{sg}(\chi_R(x, y)) + z \simeq 0].$$

   Očito je G parcijalno rekurzivna funkcija pa iz teorema rekurzije slijedi da postoji $e \in \mathbb{N}$ tako da vrijedi $\{e\}^1(y) \simeq G(e, y)$. Lako je vidjeti da je domena funkcije $\{e\}$ jednaka skupu $\mathbb{N} \setminus \{e\}$.

4. Dokažite da postoji $e \in \mathbb{N}$ takav da je domena funkcije $\{e\}$ jednaka:

   a) $\{(e, 0)\}$
   b) $\{(e, e)\}$
   c) $\{0, \ldots, e\}$
   d) $\{e \cdot k \ : \ k \in \mathbb{N}\}$
   e) $\{1 + e, 1 + e^2, 1 + e^3, \ldots\}$
   f) $\{(n, e) \mid n \in \mathbb{N}\}$
   g) $\{(x, y) \in \mathbb{N}^2 \mid x^2 + y^2 \leq e\}$.

   Upute. Za svaki od zadataka prvo definiramo relaciju R. Primjerice, za zadatak a) definiramo:

   $$(x_1, x_2, x_3) \in R \ \text{ ako i samo ako } \ x_1 = x_2 \ \wedge \ x_3 = 0$$

   Za zadatak c) definiramo: $(x_1, x_2) \in R$ ako i samo ako $x_2 \leq x_1$.
   Za zadatak e) definiramo: $x_1 R x_2$ ako i samo ako $(\exists m > 0) x_2 = 1 + x_1^m$.
   Svaka od navedenih relacija je očito rekurzivna. Zatim, definiramo parcijalno rekurzivnu funkciju G čija je mjesnost jednaka mjesnosti relacije R. Ako je R relacija mjesnosti k tada definiramo:

   $$G(x_1, \ldots, x_k) \simeq \mu z[\overline{sg}(\chi_R(x_1, \ldots, x_k)) + z \simeq 0].$$

   Očito je svaka takva funkcija G parcijalno rekurzivna. Primjenom teorema rekurzije slijedi egzistencija traženog $e \in \mathbb{N}$ za svaki pojedini zadatak.

Zadaci (2. primjena teorema rekurzije – slike)

1. Dokažite da postoji $e \in \mathbb{N}$ takav da je slika funkcije $\{e\}^1$ skup $\{x \in \mathbb{N} : x \geq e\}$.
   Uputa. Definiramo rekurzivnu relaciju R sa:

   $$(x_1, x_2) \in R \ \text{ ako i samo ako } \ x_2 \geq x_1.$$

   Zatim definiramo funkciju G sa:

   $$G(x_1, x_2) \simeq \mu z \left(\overline{sg}(\chi_R(x_1, x_2)) \ + \ z \simeq 0 \right) + x_2.$$

   Primjenom teorema rekurzije slijedi da postoji $e \in \mathbb{N}$ tako da vrijedi: $G(e, x_2) \simeq \{e\}(x_2)$, za svaki $x_2 \in \mathbb{N}$. Očito vrijedi $Dom(\{e\}) = Rng(\{e\}) = \{n \in \mathbb{N} : n \geq e\}$.

2. Dokažite da postoji $e \in \mathbb{N}$ takav da je slika funkcije $\{e\}^1$ jednaka $\{e, e + 1, e + 2, \ldots, e + 127\}$.

3. Dokažite da postoji $e \in \mathbb{N}$ tako da je $\{e\}^7$ konstantna funkcija čija je vrijednost e.
   Rješenje. Za rekurzivnu funkciju $G(x_0, x_1, \ldots, x_7) = x_0$ postoji $e \in \mathbb{N}$ tako da za sve $\vec{x} \in \mathbb{N}^7$ vrijedi $\{e\}(\vec{x}) \simeq G(e, \vec{x}) = e$ (primjena teorema rekurzije). Budući da je funkcija f totalna tada je i funkcija $\{e\}$ totalna.

13.1.3 Riceov teorem

Teorem 13.10. (Riceov teorem) *Neka je S rekurzivan podskup od $\mathbb{N}$ koji ima svojstvo da za sve $i, j \in \mathbb{N}$, takve da je $i \in S$ i $\{i\} \simeq \{j\}$ slijedi $j \in S$. Tada je $S = \emptyset$ ili $S = \mathbb{N}$.*

Zadaci (primjena Riceovog teorema)

1. Neka je $S = \{e \in \mathbb{N} : e \text{ je indeks funkcije } +\}$. Dokažite da skup S nije rekurzivan.
   Rješenje. Budući da je funkcija + parcijalno rekurzivna tada postoji indeks za nju. To znači da je $S \neq \emptyset$. Niti jedan neparan broj nije kod konačnog niza, pa nije ni indeks neke funkcije. Tada posebno primjerice $3 \notin S$. Dakle, $S \neq \mathbb{N}$. Očito vrijedi: ako $i \in S$, te $\{i\} \simeq \{j\}$, tada je $j \in S$. Iz Riceovog teorema slijedi da skup S nije rekurzivan.

2. Neka je $S = \{\langle a, b\rangle \mid \text{broj } a \text{ je element domene funkcije } \{b\}\ \}$. Dokažite da S nije rekurzivan skup.
   Rješenje. Označimo $T = \{b : \langle 0, b\rangle \in S\}$, tj. $T = \{b : \text{broj 0 je element domene funkcije } \{b\}\ \}$. Neka su $i, j \in \mathbb{N}$ takvi da je $i \in T$ te je $\{i\} \simeq \{j\}$. Tada je očito $j \in T$. Budući da, primjerice, $3 \notin T$, tada $T \neq \mathbb{N}$. S druge strane budući da je 128 kod programa 0. INC $\mathcal{R}_0$, te je očito $0 \in Dom_{\{128\}}$, tada je $T \neq \emptyset$. Iz Riceovog teorema slijedi da skup T nije rekurzivan. Iz definicije skupa T slijedi da ni skup S nije rekurzivan.

3. Neka je $S = \{\langle a, b\rangle \mid \text{funkcije } \{a \cdot b\} \text{ i } \{b\} \text{ imaju istu domenu}\}$. Dokažite da S nije rekurzivan skup.

4. Dokažite da skup $\{\langle a, b\rangle \mid \text{ funkcije } \{a\} \text{ i } \{b\} \text{ imaju istu sliku}\}$ nije rekurzivan.

13.2 Churchova teza

Bili smo već naglasili da smatramo da je svaka parcijalno rekurzivna funkcija izračunljiva (u intuitivnom smislu koji smo opisali na samom uvodu). Alonso Church je 1936. godine postavio tezu da vrijedi i obrat. Zbog važnosti sada je posebno ističemo.

Churchova teza. *Svaka izračunljiva funkcija je parcijalno rekurzivna.*

Budući da je pojam izračunljive funkcije intuitivan pojam, tj. nije strogo definiran, nemoguće je dati dokaz Churchove teze. Oboriti pak Churchovu tezu značilo bi odrediti funkciju za koju bi se svi složili da je izračunljiva, a istovremeno bi dokazali da nije parcijalno rekurzivna. No, to do sada nije učinjeno. Ovdje navodimo dvije važne činjenice zbog kojih Churchovu tezu smatramo istinitu.

1. Razni načini definiranja novih funkcija pomoću već danih parcijalno rekurzivnih funkcija (npr. simultana rekurzija, povratna rekurzija, definicija funkcija po slučajevima, ...) daju ponovno parcijalno rekurzivne funkcije.

2. Sve do sada poznate definicije koje imaju za cilj opisati klasu izračunljivih funkcija (parcijalno rekurzivne funkcije, RAM–izračunljive funkcije, Turing izračunljive, ABAK–izračunljive, ...) definiraju istu klasu funkcija.

Važnost Churchove teze je vrlo velika. Ona se primjenjuje uvijek prilikom dokaza nepostojanja algoritma za rješavanje nekog problema (tj. nerješivosti problema). Sada dajemo dva primjera takvih problema.

Primjer 13.11. (Postoji funkcija koja nije izračunljiva)
U svrhu dokaza gornje tvrdnje definiramo funkciju F na sljedeći način:

$$F(x) \simeq \begin{cases} \{x\}(x)+1, & \text{ako je } \{x\}(x)\downarrow; \\ 0, & \text{inače.} \end{cases}$$

Lako je vidjeti da niti za jedan $e \in \mathbb{N}$ ne vrijedi $F \simeq \{e\}$. To znači da za funkciju F ne postoji indeks, a tada znamo da funkcija F nije parcijalno rekurzivna. Primjenom Churchove teze slijedi da funkcija F nije izračunljiva.

Primjer 13.12. (Halting problem nije rješiv)
Halting problem glasi:

> *odrediti algoritam koji će za proizvoljan program P za RAM–stroj i ulazne podatke $\vec{x}$ odrediti hoće li P–izračunavanje sa $\vec{x}$ stati.*

Dokazat ćemo da takav program ne postoji. Štoviše, definirat ćemo jedan program P za RAM–stroj i za njega pokazati da ne postoji algoritam koji bi za svaki $x \in \mathbb{N}$ ispitao hoće li P–izračunavanje sa x stati. Iz Churchove teze će tada slijediti da je halting problem nerješiv. U svrhu dokaza definiramo skup S ovako:

$$x \in S \quad \text{ako i samo ako} \quad \{x\}(x)\downarrow .$$

Napišimo definiciju funkcije F iz prethodnog primjera pomoću skupa S :

$$F(x) \simeq \begin{cases} \{x\}(x)+1, & \text{ako je } x \in S; \\ 0, & \text{inače.} \end{cases}$$

Ako bi skup S bio rekurzivan tada bi funkcija F bila definirana po slučajevima pomoću rekurzivnih uvjeta i parcijalno rekurzivnih funkcija. Iz teorema o rekurzivnosti funkcije definirane po slučajevima tada bi slijedilo da je funkcija F parcijalno rekurzivna. No, u prethodnom primjeru dokazali smo da funkcija F nije parcijalno rekurzivna. Dakle, skup S nije rekurzivan. Primjenom Churchove teze slijedi da ne postoji algoritam koji će za svaki $x \in \mathbb{N}$ ispitati je li izraz $\{x\}(x)$ definiran. Neka je funkcija G definirana sa $G(x) \simeq \{x\}(x)$. Iz Kleenijevog teorema o normalnoj formi slijedi da je funkcija G parcijalno rekurzivna. Neka je P program za RAM–stroj koji izračunava funkciju G. Očito vrijedi:

> *RAM–stroj će stati prilikom P–izračunavanja sa x ako i samo ako je izraz $\{x\}(x)$ definiran.*

No, prije smo bili dokazali da ne postoji algoritam koji bi za svaki $x \in \mathbb{N}$ ispitivao je li izraz $\{x\}(x)$ definiran. Dakle, za program P ne postoji algoritam koji bi za svaki $x \in \mathbb{N}$ ispitao hoće li P–izračunavanje sa x stati.