Matematička logika u računarstvu (MLUR)

predavanja: Nebojša Ikodinović
vežbe: Stefan Mitrović

2026/27

https://faculty.matf.bg.ac.rs/p/nebojsa.ikodinovic/

Obaveza 1– pismeni ispit [poeni: 0–40]
Pismeni je položen ako se postigne najmanje 21 poen; položen pismeni važi do prvog izlaska na usmeni (u bilo kom ispitnom roku).

Obaveza 2– provera temeljnih znanja [poeni: 0/30]
Provera temeljnih znanja neophodnih za ostvarivanje ciljeva kursa; obaveza 2 se ocenjuje sa 0 ili 30 poena; poeni mogu da se ostvare na jedan od sledeća dva načina:
∗ student dobija 30 poena ako tokom semestra osvoji više od 75% poena na svakom od dva kolokvijuma koja će biti organizovana u kolokvijumskim nedeljama u terminu predavanja ili vežbi; na kolokvijumu se proverava znanje o temeljnim konceptima kroz jednostavna pitanja i elementarne zadatke;
∗ student može da nadoknadi ovih 30 poena nakon položenog pismenog dela, u terminu usmenog, tako što potpuno tačno (100%) odgovori na sva zadata pitanja i zadatke o fundamentalnim konceptima kursa.

Obaveza 3– usmeni ispit [poeni: 0/5/10/15/20/25/30]
Student koji je položio pismeni deo ispita i proveru temeljnih znanja, stiče pravo da polaže i usmeni deo ispita. Usmeni ispit čini usmeno odgovaranje na tri slučajno izabrana teorijska pitanja i odgovor na svako pitanje se ocenjuje sa 0, 5 ili 10 poena, u zavisnosti od kvaliteta i širine odgovora.

Konačna ocena se formira u terminu usmenog ispita na osnovu ukupnog zbira osvojenih poena: 6 [51,60], 7 [61,70], 8 [71,80], 9 [81,90], 10 [91,100]. Ukoliko student tokom semestra osvoji 30 poena i položi pismeni, ocena može biti formirana bez polaganja usmenog ispita.

Student mora da prijavi ispit u ispitnom roku u kom će izaći na pismeni ispit, i u ispitnom roku u kom će izaći na usmeni ispit.

Literatura:

- S. Abramsky, Dov M. Gabbay, T. S. E. Maibaum (eds.) *Handbook of Logic in Computer Science*, Oxford University Press (vol. 1. 1992, vol. 2. 1993, vol. 3. 1995, vol. 4. 1995, vol. 5. 2000)
- M. Ben-Ari, *Mathematical Logic for Computer Science*, 3rd edition, Springer, 2012
- A. R. Bradley, Z. Manna, *The Calculus of Computation, Decision Procedures with Applications to Verification*, Springer, 2007
- J. Harrison, *Handbook of Practical Logic and Automated Reasoning*, Cambridge University Press, 2009
- S. Hedman, *A First Course in Logic*, Oxford University Press, 2004
- M. Huth, M. Ryan, *Modelling and Reasoning about Systems*, Cambridge University Press, 2004
- R. David, K. Nour, C. Raffali, *Introduction à la logique – Théorie de la démonstration*, Dunod, 2003

...

Formalni računi (gramatike, teorije...)

Alfabet Σ je bilo koji neprazan konačan skup čije elemente nazivamo *simbolima*:

- $\Sigma_{\mathrm{U}} = \{1\}$ – *unarni alfabet*, koji sadrži samo jedan simbol;
- $\Sigma_2 = \{0, 1\}$ – *binarni alfabet*;
- $\Sigma_{\mathrm{lat}} = \{a, b, c, \ldots, z\}$ – mala slova latinice;
- $\Sigma_{\mathrm{keyboard}} = \{\mathtt{A}, \mathtt{a}, \mathtt{B}, \mathtt{b}, \ldots, \mathtt{Z}, \mathtt{z}, \sqcup, >, <, (,), \ldots, !\}$ – alfabet svih simbola tastature (pri čemu $\sqcup$ označava blanko znak); itd.

Reč je konačan niz simbola Σ koji zapisujemo $a_1 \cdots a_m$, umesto $a_1, \ldots, a_m$, odnosno $(a_1, \cdots, a_m)$.

Za $m \geqslant 2$, Σ^m je skup svih reči dužine m.

Skup svih reči nad alfabetom Σ označavamo Σ^*, a skup svih nepraznih reči Σ^+:

$$\Sigma^+ = \bigcup_{m \geqslant 1} \Sigma^m, \ \Sigma^* = \bigcup_{m \geqslant 0} \Sigma^m = \Sigma^+ \cup \{\varepsilon\}.$$

Ako $w \in \Sigma^*$, tada jedinstveni m takav da $w \in \Sigma^m$ nazivamo dužinom reči w i pišemo $|w| = m$. Primetimo da je $|\Sigma^m| = |\Sigma|^m$.

Lema $(\Sigma^*, \hat{}, \varepsilon)$ je monoid. ($\hat{}$ je operacija dopisivanja, a ε prazna reč).

Napomena: Za $w \in \Sigma^*$ i $n \in \mathbb{N}$ definišemo w^n sa: $w^0 = \varepsilon, \ w^{n+1} = w^n w$.

Jezik nad alfabetom Σ je bilo koji podskup od Σ^*.

PRIMER Trivijalni jezici nad bilo kojim alfabetom Σ jesu: $\emptyset$ (prazan jezik), $\{\varepsilon\}$ (jezik koji sadrži samo praznu reč) i Σ^* (skup svih reči nad Σ). Nekoliko netrivijalnih jezika nad $\{a, b\}$:

$L_1 = \{\varepsilon, 01, 101\}$;

$L_2 = \{1^p \mid p \text{ je prost broj}\} = \{11, 111, 11111, 1111111, 11111111111 \ldots\}$;

$L_3 = \{0^i 1^{i+j} 0^j \mid i, j \geqslant 1\}$;

$L_4 = \{w \in \{0,1\}^* \mid |w|_0 = |w|_1\}$, pri čemu, za $s \in \{0,1\}$, $|w|_s$ označava broj pojavljivanja simbola s u reči w.

PROBLEMI: Problem pripadanja jeziku. (Membership problem)

KLASIFIKACIJA PROBLEMA: Odlučivi i neodlučivi problemi

KARDINALNI ARGUMENT: Kardinalnost skupa svih jezika nad alfabetom? Kardinalnost svih odlučivih jezika?

$L_{\text{PYTHON}} = \{w \in \Sigma_{\text{keyboard}} \mid w \text{ je sintaksno korektan PYTHON program}\}$ je odlučiv

```
# SyntaxError
print("Hallo world
```

$\notin L_{\text{PYTHON}}$

```
# Program AI
x = input("Unesi vrednost: ")
print("Odlican izbor je ", x)
```

$\in L_{\text{PYTHON}}$

```
# Program Sizif
while True:
    x = input("Unesi nešto: ")
    print("Samo tako nastavi.")
```

$\in L_{\text{PYTHON}}$

$L_{\text{HALT}} = \{(w, u) \mid$ Ako $w \in L_{\text{PYTHON}}$, onda se izvršavanje programa w za ulaz u zaustavlja$\}$

$\overline{L}_{\text{HALT}} = \{w \in \Sigma_{\text{keyboard}} \mid$ Ako $w \in L_{\text{PYTHON}}$, onda se izvršavanje programa w za ulaz w zaustavlja$\}$

$\notin \overline{L}_{\text{HALT}}$ $\in \overline{L}_{\text{HALT}}$ $\in \overline{L}_{\text{HALT}}$

Hipotetički program **Halt**
Input:
    Python program P

Output:
    TRUE if P halts on P # P terminates on P
    FALSE otherwise # P runs forever on P

Halt (AI, AI) = TRUE
Halt (Sizif, Sizif) = FALSE

Hipotetički program **SELF**
Input:
    Python program P

if Halt(P, P) == FALSE then
    return TRUE
else
loop forever

Halt (**SELF**, **SELF**)?

Sistemi prepisivanja/Semi-Tuov proces

Svaki par reči u i w, nad nekim alfabetom Σ, određuje *pravilo* koje se označava $u \to w$. Pravilo $u \to w$ možemo primeniti na svaku reč v koja sadrži u kao podreč, tj. na reč v oblika v_1uv_2, za neke (možda prazne) reči v_1 i v_2. Primena pravila podrazumeva da podreč u zamenimo rečju w; pišemo:

$$v_1uv_2 \to v_1wv_2.$$

Dozvoljena su i pravila sledećeg oblika:

- $u \to \varepsilon$, tj. brisanje podreči u;
- $\varepsilon \to v$, tj. umetanje reči v na bilo kom mestu.

Kada je zadat neki konačan skup pravila, kažemo da se iz v **izvodi** reč v', ako postoji konačan niz reči

$$v = v_0 \to v_1 \to v_2 \to \cdots \to v'$$

takav da je svaka reč, počevši od druge, dobijena primenom nekog od pravila na reč koja joj prethodi.

Neki skup pravila za transformaciju reči zadatog alfabeta naziva se *Semi-Tuov proces* ili *sistem prepisivanja* nad tim alfabetom.

Formalne gramatike

Formalna gramatika je posebna vrsta sistema prepisivanja nad alfabetom u kome se razlikuju dve vrste simbola:

- pomoćni simboli – tzv. *promenljive*, među kojima je jedna promenljiva izdvojena kao *polazni simbol*;
- krajnji simboli – tzv. *terminali*.

Nad ovakvim alfabetom, **formalna gramatika** je određena konačnim spiskom pravila sa čije se leve strane nalazi bar jedna promenljiva. **Jezik određen formalnom gramatikom** jeste skup svih reči sastavljene od terminala koje se mogu izvesti iz polaznog simbola.

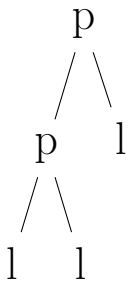
KANONSKI PRIMER Navodimo pojednostavljenu gramatiku za opisivanje punih binarnih satabla u LaTeX-u, upotrebom paketa `forest`.

Alfabet $\Sigma = \{T, p, l, [,]\}$, gde je T jedina promenljiva (pa samim tim i početni simbol), a p, l, [,] su terminali;

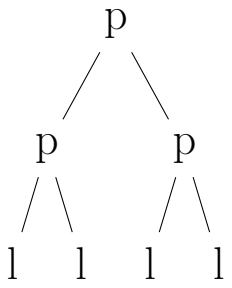
Pravila su: $\boxed{T \to [l] \quad T \to [pTT]}$

Pravilom $T \to [l]$ se grade najjednostavnija stabla koja sadrže samo jedan čvor l – list (leaf). Pravilom $T \to [pTT]$ se formiraju složenija stabla, u kojima je p unutrašnji čvor (parent), prvo T odredjuje levo podstablo, a drugo T desno podstablo.

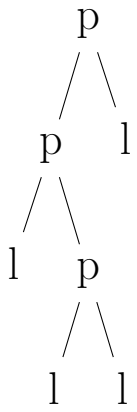
```
\begin{forest}
[p[p[l][l]][l]]
\end{forest}
```




```
\begin{forest}
[p[p[l][l]][p[l][l]]]
\end{forest}
```




```
\begin{forest}
[p[p[l][p[l][l]]][l]]
\end{forest}
```

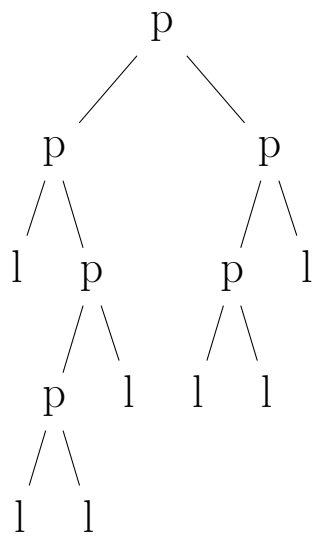


ZADATAK Navodimo pojednostavljenu gramatiku za opisivanje punih binarnih satabla u LaTeX-u, upotrebom paketa `forest`.

Alfabet $\Sigma = \{T, p, l, [,]\}$, gde je T jedina promenljiva (pa samim tim i početni simbol), a p, l, [,] su terminali;

Pravila su: $\boxed{T \to [l] \quad T \to [pTT]}$

```
\begin{forest}
?
\end{forest}
```




```
\begin{forest}
[p[l][p[l][p[p[p[l][l]][l]][l]]]]
\end{forest}
```

?

PRIMER Dikov jezik (Dyck language) generiše reči od dva simbola, leve i desne zagrade, u kojima su zagrade ispravno uparene. Na primer, (), ()(), (()()), (())(), ((())) i sl. ali ne)(, (()), ()(, (())), ...

Alfabet $\Sigma = \{D, (,)\}$, gde je D jedina promenljiva (pa samim tim i početni simbol), a (i) su terminali;

Pravila su: $\boxed{D \to \varepsilon \quad D \to (D) \quad D \to DD}$

NAPOMENA U teoriji formalnih jezika, Dikov jezik pravilno ugnježdenih zagrada povezan je sa tzv. stekovskim automatima (pushdown automata) i javlja se u sintaksnoj analizi programskih jezika.

ZADATAK. a) Uporediti broj:

- broj reči Dikovog jezika sa n parova zagrada;
- broj punih binarnih stabala sa n unutrašnjih čvorova (tj. p-čvorova),

za $n = 1, 2, 3, 4$.

b) Katalanovi brojevi su definisani jednakošću $C_n = \dfrac{1}{n+1}\binom{2n}{n}$. Izračunati C_0, C_1, C_2, C_3, C_4. Uporediti brojeve sa rezultatima dela pod a).

c) Definisati (*prirodnu*) bijekciju između reči Dikovog jezika i punih binarnih stabala, tako da broj parova zagrada u reči Dikovog jezika bude jednak broju unutrašnjih čvorova odgovarajućeg punog binarnog stabla.

d) Dokazati jednakost $C_n = \displaystyle\sum_{k=0}^{n-1} C_k C_{n-1-k}$.

PRIMER Palindrom je reč koja se jednako čita i sleva nadesno i zdesna nalevo. Naravno, i prazna reč je palindrom. Neka je P skup svih palindroma nad $\{0,1\}$, tj. reči koje se jednako čitaju sleva nadesno i zdesna nalevo: ε, 0, 1, 00, 11, 000, 010, 101, 111, 0000 itd.

$0101001101010001101001111110001010101000110010110001010101000111111001011000101011001010 \in P$?

[**Induktivna definicija**]

Skup P je najmanji, u smislu inkluzije, skup reči nad $\{0,1\}$ koji zadovoljava sledeće uslove:

- ε, 0 i 1 su palindromi ($\varepsilon, 0, 1 \in P$);
- ako je w palindrom, onda su to i $0w0$ i $1w1$ (ako $w \in P$, onda $0w0, 1w1 \in P$).

[**Formalna gramatika**]

$\Sigma = \{P, 0, 1\}$, gde je P jedina promenljiva (pa samim tim i početni simbol), a 0 i 1 su terminali;

$$P \to \varepsilon \qquad P \to 0 \quad P \to 1 \quad P \to 0P0 \quad P \to 1P1$$

Skraćeno,

$$P \to \varepsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$$

[**PYTHON program**]

```
rec = input("Unesi binarnu reč: ")
# Provera da li je reč binarna
if not all(c in "01" for c in rec):
    print("Greška: reč nije binarna.")
else:
    palindrom = True
    for i in range(len(rec) // 2):
        if rec[i] != rec[len(rec) - i - 1]:
            palindrom = False
            break
    if palindrom:
        print("Reč je palindrom.")
    else:
        print("Reč nije palindrom.")
```

Primer

[**Induktivna definicija**]

Skup aritmetičkih izraza I je najmanji skup koji zadovoljava sledeće uslove:

- promenljive x, y, z, i simboli 0 i 1 su izrazi ($x, y, z, 0, 1 \in I$);
- ako su α i β izrazi, onda je $(\alpha + \beta)$ izraz (tj. ako $\alpha, \beta \in I$, onda i $(\alpha + \beta) \in I$);
- ako su α i β izrazi, onda je $(\alpha * \beta)$ izraz (tj. ako $\alpha, \beta \in I$, onda i $(\alpha * \beta) \in I$).

[**Formalna gramatika**]

$\{I, +, *, x, y, z, 0, 1, (,)\}$, gde je I promenljiva, a ostali simboli su terminali;

$I \to x \mid y \mid z \mid 0 \mid 1 \mid (I + I) \mid (I * I)$

Zadatak Izvesti izraz $(0 + ((y * 1) + x))$

PRIMER Sastavimo gramatiku za opisivanje stabla izraza definisanih gramatikom

$$I \to x \mid y \mid z \mid 0 \mid 1 \mid (I + I) \mid (I * I)$$

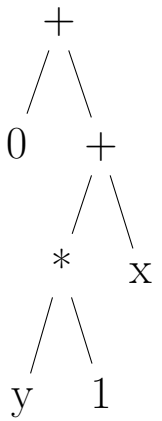
u LaTeX-u, upotrebom paketa `forest`.

Pravila su: $\boxed{T \to [x] \mid [y] \mid [z] \mid [0] \mid [1] \mid [+TT] \mid [*TT]}$

Npr.

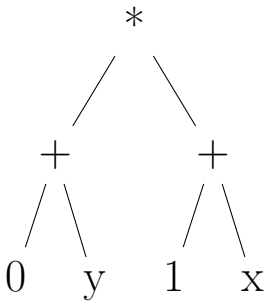
$(0 + ((y * 1) + x))$

```
\begin{forest}
[+[0][+[*[y][1]][x]]]
\end{forest}
```



$((0 + y) * (1 + x))$

```
\begin{forest}
[*[+[0][y]][+[1][x]]]
\end{forest}
```



Kanonski primer

[**Induktivna definicija**]

Skup aritmetičkih izraza I je najmanji skup koji zadovoljava sledeće uslove:

- simbole konstanti 0, 1, 2, ... i promenljive x_0, x_1, x_2, ...;
- ako su α i β izrazi, onda je $(\alpha + \beta)$ izraz (tj. ako $\alpha, \beta \in I$, onda i $(\alpha + \beta) \in I$);
- ako su α i β izrazi, onda je $(\alpha * \beta)$ izraz (tj. ako $\alpha, \beta \in I$, onda i $(\alpha * \beta) \in I$).

[**Formalna gramatika**]

$\{N, V, I, +, *, x, 0, ', (,)\}$, gde su N, V, I promenljive, I je početna promenljiva, a ostali simboli su terminali;

$N \to 0 \mid N'$

$V \to x \mid V'$

$I \to N \mid V \mid (I + I) \mid (I * I)$

N: 0, $0'$, $0''$, $0'''$, ... zapisujemo i 0, 1, 2, 3, ...

V: x, x', x'', x''', ... zapisujemo i x_0, x_1, x_2, x_3, ...

Npr. umesto $(((0'' \cdot x''') + (0''' \cdot 0'')) \cdot x')$ kraće pišemo $(((2 \cdot x_3) + (3 \cdot 2)) \cdot x_1)$

Napomena Često se promenljive označavaju nekom karakterističnom skraćenicom (ili čtiavom rečju) umesto jednim slovom, a umesto $\to$ se koristi ::= (ili := uglavnom uz konvenciju o *zloupotrebi* simbola)

$\langle \texttt{num} \rangle ::= 0 \mid \langle \texttt{num} \rangle'$

$\langle \texttt{var} \rangle ::= x \mid \langle \texttt{var} \rangle'$

$\langle \texttt{izraz} \rangle ::= \langle \texttt{num} \rangle \mid \langle \texttt{var} \rangle \mid (\langle \texttt{izraz} \rangle + \langle \texttt{izraz} \rangle) \mid (\langle \texttt{izraz} \rangle * \langle \texttt{izraz} \rangle)$

$\langle \texttt{num} \rangle ::= 0 \mid \langle \texttt{num} \rangle'$
$\langle \texttt{var} \rangle ::= x \mid \langle \texttt{var} \rangle'$
$\langle \texttt{izraz} \rangle ::= \langle \texttt{num} \rangle \mid \langle \texttt{var} \rangle \mid (\langle \texttt{izraz} \rangle + \langle \texttt{izraz} \rangle) \mid (\langle \texttt{izraz} \rangle * \langle \texttt{izraz} \rangle) \mid -\langle \texttt{izraz} \rangle$

$\langle \texttt{bool} \rangle ::= \text{True} \mid \text{False} \mid \langle \texttt{izraz} \rangle == \langle \texttt{izraz} \rangle \mid !\langle \texttt{bool} \rangle \mid \langle \texttt{bool} \rangle \,\&\, \langle \texttt{bool} \rangle$

$\langle \texttt{prog} \rangle ::= \textbf{Skip} \mid \langle \texttt{var} \rangle := \langle \texttt{izraz} \rangle \mid \langle \texttt{prog} \rangle; \langle \texttt{prog} \rangle \mid \textbf{if } \langle \texttt{bool} \rangle \textbf{ then } \langle \texttt{prog} \rangle \textbf{ else } \langle \texttt{prog} \rangle \textbf{ fi} \mid$
$\textbf{while } \langle \texttt{bool} \rangle \textbf{ do } \langle \texttt{prog} \rangle \textbf{ od}$

Mere olakšavanja primene "administrativnih" zabrana

Konvencije o brisanju zagrada: dobro-formirani izraz $(((2 \cdot x_3) + (3 \cdot 2)) \cdot x_1)$ zapisujemo $(2x_3 + 3 \cdot 2) \cdot x_1$

Supstitucija Uopšte, ideja promenljive je suštinski povezana sa supstitucijom, tj. zamenom bez izuzetka svih pojavljivanja te promenljive jednim istim izrazom.

Npr. uz podrazumevane konvencije, nakon supstitucije $(2x_3 + 3 \cdot 2) \cdot x_1\ [x_3 := 1 + x_1]$ dobijamo novi izraz $(2(1 + x_1) + 3 \cdot 2) \cdot x_1$; nakon supstitucije $(2x_3 + 3 \cdot 2) \cdot x_1\ [x_2 := 1 + x_1]$ dobijamo isti izraz ...

Uopšteno, $I[x_i := E]$, gde su I i E neki izrazi i $i \geqslant 0$ definišemo **induktivno po složenosti** izraza I:

$\boxed{\langle \texttt{izraz} \rangle := \langle \texttt{num} \rangle \mid \langle \texttt{var} \rangle \mid (\langle \texttt{izraz} \rangle + \langle \texttt{izraz} \rangle) \mid (\langle \texttt{izraz} \rangle * \langle \texttt{izraz} \rangle)}$

I	$I[x_i := E]$
n (tipa $\langle \texttt{num} \rangle$)	n
x_j, $j \neq i$	x_j
x_i	E
$(P + Q)$	$(P[x_i := E] + Q[x_i := E])$
$(P * Q)$	$(P[x_i := E] * Q[x_i := E])$

Spec. $[x_i := x_j]$ nazivamo preimenovanje promenljive.

Upotreba metapromenljivih. Slova x, y, z, ... se često koriste da označe "proizvoljne" promenljive x_0, x_1, x_2, x_3, ...

ZADATAK. Dokazati da su sledeći izrazi istovetni $I[x := E][y := F]$ i $I[y := F][x := E[y := F]]$.

UPUTSTVO. Indukcijom "po" I.

Simultana supstitucija $[x := E, y := F]$ nije nadovezana primena običnih supstitucija $[x := E][y := F]$ ili $[y := F][x := E]$.

Npr.

$$\begin{array}{llll}
(2y+3)\cdot x\ [y := 1+x][x := y] & \to & (2(1+x)+3)\cdot x\ [x := y] & \to\ (2(1+y)+3)\cdot y \\
(2y+3)\cdot x\ [x := y][y := 1+x] & \to & (2y+3)\cdot y\ [y := 1+x] & \to\ (2(1+x)+3)\cdot(1+x) \\
(2y+3)\cdot x\ [x := y, y := 1+x] & \to & & (2(1+x)+3)\cdot y
\end{array}$$

Ipak, $I[x := E, y := F]$ se može svesti na uzastupne supstitucije. Neka je z promenljiva koja se ne pojavljuje u I, E, F. (Da li takva uvek postoji?) Neka je $\overline{E}$ izraz $E[y := z]$. Tada je:

$I[x := E, y := F]$ izraz $I[x := \overline{E}][y := F][z := y]$.

NAPOMENA! Izrazi sa promenljivama jesu tipični načini definisanja funkcija. Npr. $x \mapsto (2y+3) \cdot x$ posmatramo kao funkciju po x, dok y posmatramo kao (slobodan) parametar. Uobičajenom terminologijom, $x \mapsto (2y+3) \cdot x$ predstavlja familiju funkcija $f_y(x) = (2y+3)x$.

Primenu neke funkcije f_y na argument 5, tj. $f_y(5)$ posmatramo kao "izračunavanje" $(2y+3) \cdot x[x := 5]$.

Slično prethodnom, $y \mapsto (2y+3) \cdot x$ posmatramo kao familiju funkcija $f_x(y) = (2y+3)x$, i tada je $f_x(5)$ zapravo $(2y+3) \cdot x[y := 5]$.

Svaki slobodan parametar se dalje može "argmentizovati". Npr.

$$y \mapsto (x \mapsto (2y+3) \cdot x)$$

posmatramo kao funkciju dva argumenta i skraćeno pišemo:

$$(y, x) \mapsto (2y+3) \cdot x$$

Osnovna jednakost

$$\boxed{(x \mapsto I)(E) = I[x := E]}$$

KANONSKI PRIMER. λ-izrazi

Alfabet: $\Sigma_\lambda = \{x, ', (,), \lambda\}$

$\langle \texttt{variable} \rangle ::= x \mid \langle \texttt{variable} \rangle '$

$\langle \texttt{term} \rangle ::= \langle \texttt{variable} \rangle \mid (\langle \texttt{term} \rangle \langle \texttt{term} \rangle) \mid (\lambda \langle \texttt{variable} \rangle \langle \texttt{term} \rangle)$

Primeri: $(\lambda x.x)$, $(\lambda x.y)$, $((\lambda x.x)(\lambda x.y))$ $((\lambda y.(\lambda x.y))z)$

Mere olakšavanja primene "administrativnih" zabrana

Asocijativnost sleva:

$FE_1E_2 \cdots E_k$ je kraći zapis za $(\cdots((FE_1)E_2)\cdots E_k)$

Združivanje zdesna:

$\lambda x_1 x_2 \cdots x_k.E$ je skraćenje za $(\lambda x_1.(\lambda x_2.(\cdots(\lambda x_k.E)\cdots))$

Izostavljanje spoljašnjih zagrada:

Npr. $((\lambda x.x)y)$ skraćujemo $(\lambda x.x)y$

ZADATAK. Koje λ-izraze predstavljaju sledeće skraénice: $\lambda x.xx$, $\lambda xy.x$, $\lambda xyz.x(yz)$, $\lambda xyz.xz(yz)$?

Formalna teorija - "mašina" za generisanje teorema

Formalnu teoriju čine:

- alfabet Σ;
- skup formula For $\subseteq \Sigma^*$ (koji je uglavnom određen nekom gramatikom);
- skupom aksioma Ax $\subseteq$ For (iz skupa svih formula izdvajamo neke formule koje se smatraju aksiomama);
- skupom pravila izvođenja P oblika:

$$(P): \ \frac{\alpha_1 \quad \cdots \quad \alpha_n}{\alpha},$$

  gde $\alpha_1, \ldots, \alpha_n, \alpha$ označavaju neke formule iz For; formule $\alpha_1, \ldots, \alpha_n$ nazivaju se *premise* (pretpostavke), a formula α *zaključak* pravila.

Prilikom zadavanja neke formalne teorije $\mathcal{F}$, navodimo sva četiri skupa koja određuju tu teoriju i pišemo $\mathcal{F} = (\Sigma, \text{For}, \text{Ax}, \mathcal{P})$.

DEFINICIJA. Neka je Γ podskup skupa formula neke formalne teorije $\mathcal{F}$. Konačan niz formula $\varphi_1, \ldots, \varphi_k$ je *izvođenje* iz hipoteza Γ u formalnoj teoriji $\mathcal{F}$ ako svaka formula φ_i, $i = 1, \ldots, n$, ispunjava jedan od uslova:

- φ_i je aksioma, ili
- φ_i je hipoteza, tj. formula iz Γ, ili
- φ_i se može dobiti iz nekih od prethodnih formula $\varphi_1, \ldots, \varphi_{i-1}$ primenom nekog od pravila izvođenja iz $\mathcal{P}$.

Formula φ je *posledica hipoteza* Γ, u oznaci $\Gamma \vdash_{\mathcal{F}} \varphi$ ako postoji izvođenje iz Γ čiji je poslednji član formula φ.

Ukoliko je $\Gamma = \emptyset$, pišemo $\vdash_{\mathcal{F}} \varphi$ i kažemo da je φ teorema formalne teorije $\mathcal{F}$.

LEMA 1) Ako je $\Gamma \vdash_{\mathcal{F}} \varphi$, onda $\Gamma, \theta \vdash_{\mathcal{F}} \varphi$, za bilo koju formulu θ.
2) Ako je $\Gamma \vdash_{\mathcal{F}} \theta$ i $\theta \vdash_{\mathcal{F}} \varphi$, onda $\Gamma \vdash_{\mathcal{F}} \varphi$.

ZADATAK. Skup formula je određen gramatikom:

$\langle \texttt{var} \rangle ::= x \mid y \mid z \mid \langle \texttt{var} \rangle'$

$\langle \texttt{izraz} \rangle ::= \langle \texttt{var} \rangle \mid \min(\langle \texttt{izraz} \rangle, \langle \texttt{izraz} \rangle) \mid \max(\langle \texttt{izraz} \rangle, \langle \texttt{izraz} \rangle)$

$\langle \texttt{formula} \rangle ::= \langle \texttt{izraz} \rangle \leqslant \langle \texttt{izraz} \rangle$

Aksiome: $I \leqslant I$, za svaki izraz I

Pravila izvođenja:

$$\frac{A \leqslant C}{\min(A,B) \leqslant C}\,(\min_L) \qquad \frac{A \leqslant B \quad A \leqslant C}{A \leqslant \min(B,C)}\,(\min_D) \qquad \frac{\min(A,B) \leqslant C}{\min(B,A) \leqslant C}\,(\min_K)$$

$$\frac{A \leqslant C \quad B \leqslant C}{\max(A,B) \leqslant C}\,(\max_L) \qquad \frac{A \leqslant B}{A \leqslant \max(B,C)}\,(\max_D) \qquad \frac{A \leqslant \max(B,C)}{A \leqslant \max(C,B)}\,(\max_K)$$

$$\frac{A \leqslant B \quad B \leqslant C}{A \leqslant C}\,(T)$$

Dokazati:

(a) $x_1 \leqslant y_1, x_2 \leqslant y_2 \vdash \min(x_1, x_2) \leqslant \max(y_1, y_2)$

(b) $\max(x_1, x_2) \leqslant \min(y_1, y_2) \vdash x_2 \leqslant y_1$

(c) $\vdash x \leqslant \min(x, \max(x, y))$

(d) $\vdash \max(x, \min(x, y)) \leqslant x$

(e) $\vdash \max(x, \min(y, z)) \leqslant \min(\max(x, y), \max(x, z))$

(f) $\vdash \max(\min(x, y), \min(x, z)) \leqslant \min(x, \max(y, z))$

KANONSKI PRIMER. λ-račun

$\langle \texttt{variable} \rangle ::= x \mid \langle \texttt{variable} \rangle \prime$

$\langle \texttt{term} \rangle ::= \langle \texttt{variable} \rangle \mid (\langle \texttt{term} \rangle \langle \texttt{term} \rangle) \mid (\lambda \langle \texttt{variable} \rangle \langle \texttt{term} \rangle)$

$\langle \texttt{formula} \rangle ::= \langle \texttt{izraz} \rangle = \langle \texttt{izraz} \rangle$

Aksiome:

$E = E$

$(\lambda x.F)E = F[x := E]$

Pravila izvođenja:

$$\frac{E = F}{F = E} \qquad \frac{E = F \quad F = G}{E = G} \qquad \frac{E = F}{EG = FG} \qquad \frac{E = F}{GE = GF} \qquad \frac{E = F}{\lambda x.E = \lambda x.F}$$

ZADATAK. "Uprostiti" λ-izraze:
$(\lambda x.x)a$; $(\lambda x.xx)a$; $(\lambda xy.x)ab$; $(\lambda xyz.xz(yz))abc$; $(\lambda u.(\lambda v.(\lambda z.(zu)v)u)z)$

Šta je smisao λ-računa?

Semantika: interpetacija; (denotaciona) semantika ...

PRIMER Posmatrajmo jedan prilično jednostavan formalni račun.

Formalna teorija $\mathcal{D}$:

$\langle \texttt{bin} \rangle ::= \varepsilon \mid 0 \mid 1 \mid \langle \texttt{bin} \rangle \langle \texttt{bin} \rangle$

$\langle \texttt{formula} \rangle ::= \langle \texttt{bin} \rangle = \langle \texttt{bin} \rangle$

Aksiome:

$11111 = \varepsilon \qquad 00 = \varepsilon \qquad 11110 = 01$

Kakav je smisao ovog računa?

Pravila izvođenja:

$$\frac{}{E = E} \qquad \frac{E = F}{F = E} \qquad \frac{E = F \quad F = G}{E = G}$$

$$\frac{E = F \quad G = H}{EG = FH}$$

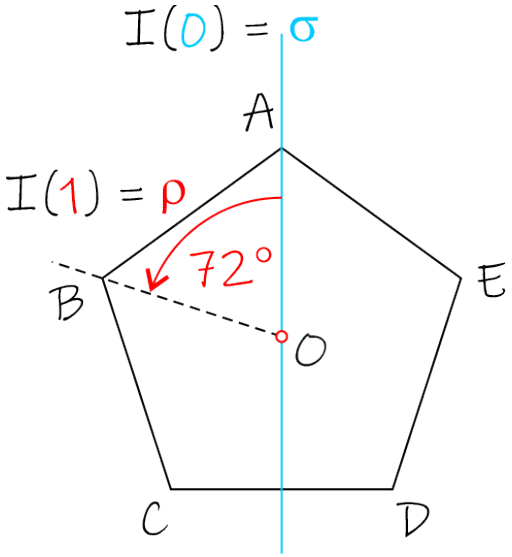
Npr. $\vdash_{\mathcal{D}} 111010100100011111 = 0010101011010$

Semantika: interpetacija; (denotaciona) semantika ...

PRIMER Posmatrajmo jezik $\{0,1\}^*$: $\langle \texttt{bin} \rangle ::= \varepsilon \mid 0 \mid 1 \mid \langle \texttt{bin} \rangle \langle \texttt{bin} \rangle$

Šta znače binarne reči?

Jedna interpretacija: binarne reči interpretiramo izometrijama ravni. Neka je $ABCDE$ pravilan petogao čiji je centar O, ι koincidencija; σ osna simetrija određena pravom AO; ρ rotacija oko tačke O za ugao $72°$, u smeru suprotnom kretanju kazaljke na satu i $\circ$ kompozicija izometrijskih transformacija. Definišimo preslikavanje $\mathcal{I} : \{0,1\}^* \to \mathrm{Iso}(\mathbb{R}^2)$:



$$\mathcal{I}(\varepsilon) = \iota,\ \mathcal{I}(0) = \sigma,\ \mathcal{I}(1) = \rho,\ \mathcal{I}(uv) = \mathcal{I}(u) \circ \mathcal{I}(v)$$

U skladu sa ovom interpretacijom, definišemo *semantičko izjednačavanje* reči $u, v \in \{0,1\}^*$:

$$u \equiv v \text{ akko } \lceil u \rceil = \lceil v \rceil.$$

Možemo li ovo izjednačavanje *sintaksno* okarakterisati nekom formalnom teorijom?

Sintaksa

Formalna teorija $\mathcal{D}$:

$\langle \texttt{bin} \rangle ::= \varepsilon \mid 0 \mid 1 \mid \langle \texttt{bin} \rangle \langle \texttt{bin} \rangle$

$\langle \texttt{formula} \rangle ::= \langle \texttt{bin} \rangle = \langle \texttt{bin} \rangle$

Aksiome:

$11111 = \varepsilon \qquad 00 = \varepsilon \qquad 11110 = 01$

Pravila izvođenja:

$$\frac{}{E = E} \qquad \frac{E = F}{F = E} \qquad \frac{E = F \quad F = G}{E = G}$$

$$\frac{E = F \quad G = H}{EG = FH}$$

Semantika

Interpretacija $\mathcal{I} : \{0,1\}^* \to \mathrm{Iso}(\mathbb{R}^2)$:

$\mathcal{I}(\varepsilon) = \iota,\ \mathcal{I}(0) = \sigma,\ \mathcal{I}(1) = \rho,\ \mathcal{I}(uv) = \mathcal{I}(u) \circ \mathcal{I}(v)$

Reči $u, v \in \{0,1\}^*$ su *semantički jednake*, u oznaci $\models_{\mathcal{I}} u = v$ (ili $u =_{\mathcal{I}} v$), ako je $\mathcal{I}(u) = \mathcal{I}(v)$.

Teorema: $\vdash_{\mathcal{D}} u = v$ akko $\models_{\mathcal{I}} u = v$.

Dokaz. ($\Rightarrow$) Tvrdjenje je tačno za aksiome? Pravila izvodjenja "čuvaju" tvrdjenje?

($\Leftarrow$) Lema 1. Za sve $u \in \{0,1\}^*$, $\mathcal{I}(u)$ je simetrija pravilnog petougla. Lema 2. Sve simetrije pravilnog petougla su ι, ρ, $\rho\rho$, $\rho\rho\rho$, $\rho\rho\rho\rho$, σ, $\rho\sigma$, $\rho\rho\sigma$, $\rho\rho\rho\sigma$, $\rho\rho\rho\rho\sigma$. Lema 3. Svaka binarna reč se u $\mathcal{D}$ može svesti na jedan od *kanonskih* oblika: ε, 1, 11, 111, 1111, 0, 10, 110, 1110, 11110. itd.

Iskazna logika

PRIMERI [zakoni zaključivanja]

1. Ako nekome kažemo *Ako si ti u pravu, onda sam ja lud ili glup.* Kada dodamo *Ja nisam lud i nisam glup*, sagovornik će znati da smo mu na jedan nobičan način saopštili: *Ti nisi u pravu.*

2. Znamo da (prema Pašovoj teoremi) *Ako prava s seče stranicu BC trougla ABC, onda s seče AB ili s seče AC.* Kada uočimo da *s ne seče AB i s ne seče AC*, zaključujemo da *prava s ne seče BC.*

3. Znamo da *Iz $ab = 0$ sledi da je $a = 0$ ili je $b = 0$.* Kada saznamo da je $a \neq 0$ *i* $b \neq 0$, zaključujemo da je $ab \neq 0$.

Ovakvih primera možemo napraviti neograničeno mnogo, postavljajući umesto čuvara mesta (tj. umesto slovo p, q, r) rečenice za koje mogu biti tačne ili netačne, menjajući pri tome iste čuvare mesta istim rečenicama:

Prva pretpostavka: *Ako p onda q ili r.*
Druga pretpostavka: *Nije q i nije r.*

Zaključak: *Nije p.*

Iskazna algebra/iskazna logika

$\mathbf{2} = (\{0,1\}, \neg, \wedge, \vee, \oplus, \Rightarrow, \Leftrightarrow)$

	$\neg$
0	1
1	0

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\oplus$	0	1
0	0	1
1	1	0

$\Rightarrow$	0	1
0	1	1
1	0	1

$\Leftrightarrow$	0	1
0	1	0
1	0	1

$\langle \texttt{log_var} \rangle ::= p \mid \langle \texttt{log_var} \rangle'$

$\langle \texttt{log_izraz} \rangle ::= 0 \mid 1 \mid \langle \texttt{log_var} \rangle \mid \neg \langle \texttt{log_izraz} \rangle \mid \langle \texttt{log_izraz} \rangle \wedge \langle \texttt{log_izraz} \rangle \mid$
$\langle \texttt{log_izraz} \rangle \vee \langle \texttt{log_izraz} \rangle \mid \langle \texttt{log_izraz} \rangle \oplus \langle \texttt{log_izraz} \rangle \mid \cdots$

PRIMER 1. [zakoni zaključivanja]

Prva pretpostavka: *Ako p onda q ili r.*
Druga pretpostavka: *Nije q i nije r.*
Zaključak: *Nije p.*

$$\frac{p \Rightarrow q \vee r \quad \neg q \wedge \neg r}{\neg p}$$

$$p \Rightarrow q \vee r, \neg q \wedge \neg r \models \neg p \qquad \models (p \Rightarrow q \vee r) \wedge (\neg q \wedge \neg r) \Rightarrow \neg p$$

Iskazna algebra

$\mathbf{2} = (\{0,1\}, \neg, \wedge, \vee, \oplus, \cdots)$

	$\neg$
0	1
1	0

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\oplus$	0	1
0	0	1
1	1	0

$\cdots$

$\mathtt{C} ::= 0 \mid 1$

$\mathtt{V} ::= p \mid \mathtt{V}'$

$\mathtt{E} ::= \mathtt{C} \mid \mathtt{V} \mid \neg\mathtt{E} \mid (\mathtt{E} \wedge \mathtt{E}) \mid \cdots$

Valuacija promenljivih $v : \{p, p', p'', \ldots\} \to \{0,1\}$

Nekoliko zadataka:

1. Izračunati vrednost formule

$$I(p,q) = (p \wedge 0) \oplus \neg q$$

za valuaciju $x = 0, y = 1$.

2. Rešiti jednačinu $(1 \wedge p) \oplus (p \vee 1) = 1$

3. Dokazati da vrednost izraza $((p \oplus q) \wedge p) \vee q$ ne zavisi od valuacije promenljivih.

4. Dokazati identitet

$$p \oplus (q \oplus r) = (p \oplus q) \oplus r.$$

Brojevna algebra

$(\mathbb{R}, -, +, \cdot, \sin, f, ...)$, pri čemu je neka f realna funkcija data sa $f(x,y,z) =_{\text{def}} \cdots$

Postupci za izračunavanje rezultata primene operacija na argumente.

$\mathtt{C} ::= 0 \mid 1 \mid 0.5 \mid 1.41 \mid 3.1415$

$\mathtt{V} ::= x \mid \mathtt{V}'$

$\mathtt{E} ::= \mathtt{C} \mid \mathtt{V} \mid -\mathtt{E} \mid (\mathtt{E}{+}\mathtt{E}) \mid (\mathtt{E}{\cdot}\mathtt{E}) \mid \sin(\mathtt{E}) \mid f(\mathtt{E},\mathtt{E},\mathtt{E}) \mid \cdots$

Valuacija promenljivih $v : \{x, x', x'', \ldots\} \to \mathbb{R}$

Nekoliko zadataka:

1. Izračunati vrednost formule

$$I(x,y) = x \cdot 0.5 + \sin y$$

za valuaciju $x = 0.17, y = 5$.

2. Rešiti jednačinu $0.5 \cdot x + x \cdot x = 3.14$

3. Dokazati da vrednost izraza $(a{+}b)^2 - a^2 - b^2 - 2ab$ ne zavisi od valuacije promenljivih.

4. Dokazati identitet

$$(a+b)^3 = a^3 + b^3 + 3ab(a+b).$$

Iskazna algebra

$\mathbf{2} = (\{0,1\}, \neg, \wedge, \vee, \oplus, \cdots)$

	$\neg$
0	1
1	0

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\oplus$	0	1
0	0	1
1	1	0

$\cdots$

$\mathtt{C} ::= 0 \mid 1$

$\mathtt{V} ::= p \mid \mathtt{V}'$

$\mathtt{E} ::= \mathtt{C} \mid \mathtt{V} \mid \neg\mathtt{E} \mid (\mathtt{E} \wedge \mathtt{E}) \mid \cdots$

Valuacija promenljivih $v : \{p, p', p'', \ldots\} \to \{0,1\}$

Nekoliko zadataka:

1. Izračunati vrednost formule

$$I(p,q) = (p \wedge 0) \oplus \neg q$$

za valuaciju $x = 0, y = 1$.

2. Rešiti jednačinu $(1 \wedge p) \oplus (p \vee 1) = 1$

3. Dokazati da vrednost izraza $((p \oplus q) \wedge p) \vee q$ ne zavisi od valuacije promenljivih.

4. Dokazati identitet

$$p \oplus (q \oplus r) = (p \oplus q) \oplus r.$$

Iskazna logika

Sintaksa

$\mathtt{C} ::= \top \mid \bot$ (logičke konstante $\top$, $\bot$)

$\mathtt{P} ::= p \mid \mathtt{P}'$ (iskazna slova: $\mathcal{P} = \{p_0, p_1, p_2, \ldots\}$)

$\mathtt{F} ::= \mathtt{C} \mid \mathtt{P} \mid \neg\mathtt{F} \mid \mathtt{F} \wedge \mathtt{F} \mid \cdots$ (skup iskaznih formula $\mathcal{F}$)

Zadatak. Induktivno definisati:
$\mathcal{P}_\varphi$ - skup svih slova koja se pojavljuju u φ;
$\mathrm{sl}(\varphi)$ - složenost formule φ;
$\mathcal{D}_\varphi$ - drvo formule φ;
$\varphi[p := \theta]$ - rezultat zamene svih pojavljivanja slova p u formuli φ formulom θ.

Semantika

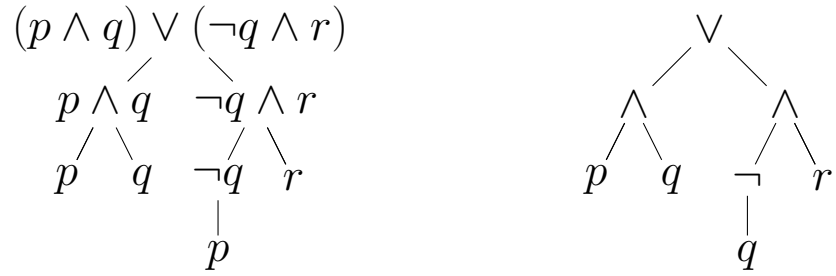
Valuacija promenljivih $v : \mathtt{P} \to \{0,1\}$ - model iskazne formule.

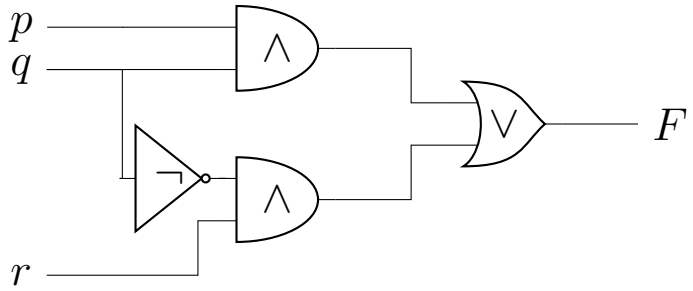
Zadatak. Induktivno definisati:
$\varphi[v]$ - vrednost formule φ za valuaciju v.

Izražajna moć iskaznog računa

Značaj algebre/logike se u velikoj meri određuje izražajna moć, tj. mogućnost da se izrazima modeluju različite situacije.

Primer 2. [Hardver i softver] Multiplekser (MUX) je kolo koje realizacije jednu od najprepoznatljivijih naredbi programskih jezika: `if` q `then` $F := p$ `else` $F := r$.



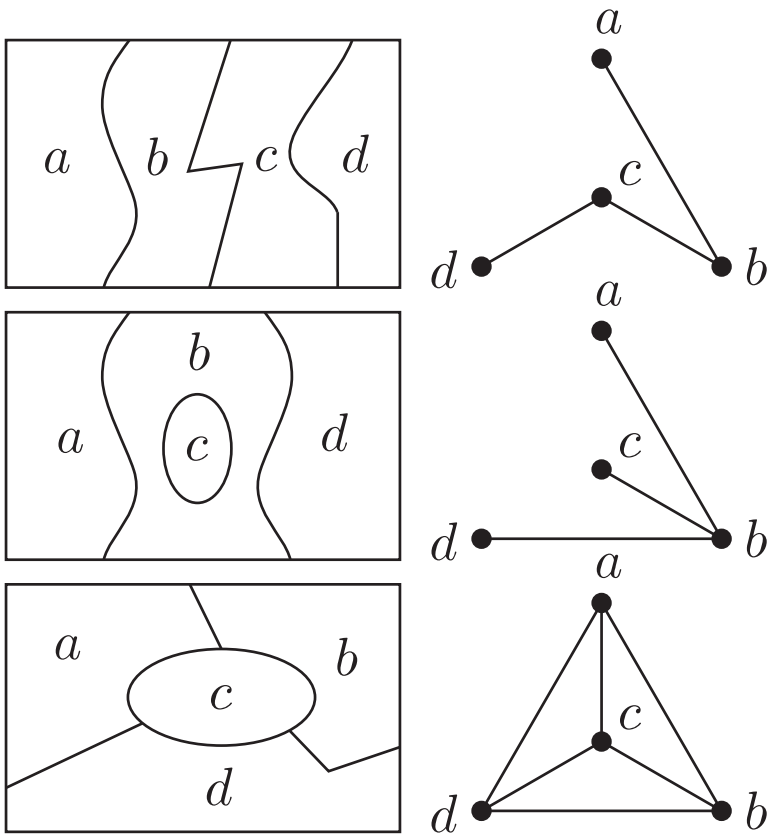


$(p$	$\wedge$	$q)$	$\vee$	$(\neg$	q	$\wedge$	$r)$
0		1			1		0

$(p$	$\wedge$	$q)$	$\vee$	$(\neg$	q	$\wedge$	$r)$
0	0	1	0	0	1	0	0

Zbog asocijativnosti, vrednost izraza $p \oplus q \oplus r$ ne zavisi od rasporeda zagrada, pa se one mogu izostaviti. Isto važi i za izraze $p_1 \oplus p_2 \oplus \cdots \oplus p_k$, za $k > 3$ promenljivih, koje kraće zapisujemo i $\bigoplus_{i=1}^{n} p_i$. Slično, $p_1 \wedge p_2 \wedge \cdots \wedge p_k$ i $p_1 \vee p_2 \vee \cdots \vee p_k$ redom zapisujemo $\bigwedge_{i=1}^{n} p_i$ i $\bigvee_{i=1}^{n} p_i$.

PRIMER 3. [Bojenje grafa] Za svaku od mapa navedenih na slikama levo, treba utvrditi da li oblasti a, b, c, d mogu obojiti u **dve** različite boje, tako da bilo koje dve susedne oblasti ne budu obojene istom bojom.



Pretpostavimo da čvorove grafa treba obojiti plavo ili crveno. Za svaki čvor v, uvedimo slova B_v i R_v koja redom označavaju *čvor v je plav* i *čvor v je crven*. Tada se bojenje čvorova u dve boje opisujemo konjunkcijom:

[svaki čvor je obojen tačno jednom bojom:]

$$(B_a \oplus R_a) \wedge (B_b \oplus R_b) \wedge (B_c \oplus R_c) \wedge (B_d \oplus R_d) \wedge$$

[susedni čvorovi nisu obojeni istom bojom:]

$$\begin{aligned} \wedge\ & (\neg B_a \vee \neg B_b) \wedge (\neg P_a \vee \neg P_b) \wedge \\ \wedge\ & (\neg B_b \vee \neg B_c) \wedge (\neg P_b \vee \neg P_c) \wedge \\ \wedge\ & (\neg B_b \vee \neg B_d) \wedge (\neg P_b \vee \neg P_d) \wedge \\ \wedge\ & (\neg B_c \vee \neg B_d) \wedge (\neg P_c \vee \neg P_d) \end{aligned}$$

PRIMER 4. [Raspored, npr. latinski kvadrati] Latinski kvadrat reda n je kvadratna tablica $n \times n$ u čija polja je upisano n različitih simbola, npr. brojevi $1, \ldots, n$, tako da se svaki simbol pojavljuje jednom u svakoj vrsti i jednom u svakoj koloni. Dodelimo poljima koordinate (i, j), $1 \leq i, j \leq n$, pri čemu su vrste numerisane odozgo nadole, a kolone sleva nadesno. Ako iskazna slova p_{ijk} znače *u polje* (i, j) *je upisan broj* k, $1 \leq i, j, k \leq n$, tada sledeće formule opisuju pravila formiranja latinskog kvadrata:

- U svako polje tablice $n \times n$ upisan je tačno jedan broj: $\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \left(\bigvee_{k=1}^{n} p_{ijk} \wedge \bigwedge_{1 \leq k < k' \leq n} \neg \left(p_{ijk} \wedge p_{ijk'} \right) \right)$
- u svakoj vrsti svaki broj se pojavljuje bar jednom: $\bigwedge_{i=1}^{n} \bigwedge_{k=1}^{n} \bigvee_{j=1}^{n} p_{ijk}$
- u svakoj koloni svaki broj se pojavljuje bar jednom: $\bigwedge_{j=1}^{n} \bigwedge_{k=1}^{n} \bigvee_{i=1}^{n} p_{ijk}$

Označimo konjunkciju svih navedenih pravila sa Λ_n.

Na primer, za $n = 2$ formula Λ_2 je (koristeći da je $q \oplus r \equiv (q \vee r) \wedge \neg(q \wedge r)$)[1]:

$$\begin{aligned} & (p_{111} \oplus p_{112}) \wedge (p_{121} \oplus p_{122}) \wedge (p_{211} \oplus p_{212}) \wedge (p_{221} \oplus p_{222}) \wedge \\ \wedge \quad & (p_{111} \vee p_{121}) \wedge (p_{112} \vee p_{122}) \wedge (p_{211} \vee p_{221}) \wedge (p_{212} \vee p_{222}) \wedge \\ \wedge \quad & (p_{111} \vee p_{211}) \wedge (p_{112} \vee p_{212}) \wedge (p_{121} \vee p_{221}) \wedge (p_{122} \vee p_{222}) \end{aligned}$$

ZADATAK. (a) Dopuniti započeti latinski kvadrat.

1	3		
		2	1
			4

(b) Rešiti jednačinu $p_{111} \wedge p_{123} \wedge p_{232} \wedge p_{241} \wedge p_{444} \wedge \Lambda_4 = 1$, po nepoznatim p_{ijk}, $1 \leq i, j, k \leq n$.

ZADATAK. Opisati iskaznom formulom pravila igre Sudoku.

[1]Formula $p \oplus q \oplus r$ znači "neparan broj istinitih slova (1 ili 3)", što nije ekvivalentno formuli $\neg(p \wedge q) \wedge \neg(q \wedge r) \wedge \neg(r \wedge p)$ koja znači "tačno jedno istinito slovo".

PRIMER 5. [Dirihleov princip/**P**igeon**h**ole **p**rinciple - PHP] Za prirodne brojeve $m, n \geq 1$ neka je:

$$(\mathrm{PHP}_{m,n}) \quad \text{ne postoji 1-1 funkcija iz } \{1, 2, \ldots, m\} \text{ u } \{1, 2, \ldots, n\},$$

ili ekvivalentno

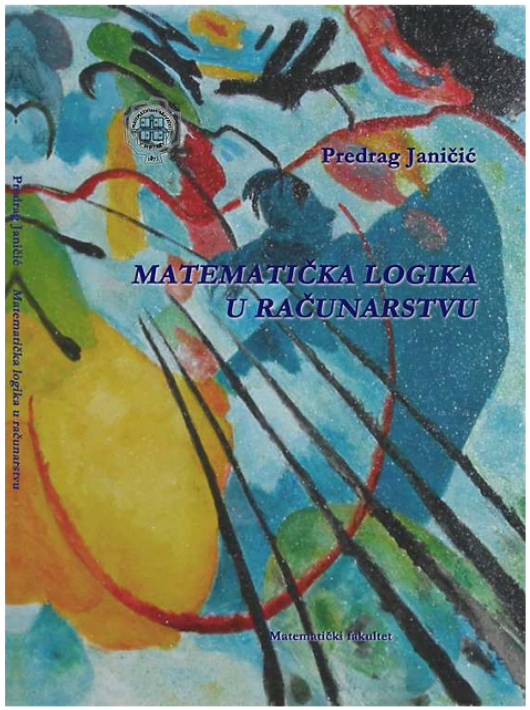
$$(\mathrm{PHP}_{m,n}) \quad \text{svaka funkcija iz } \{1, 2, \ldots, m\} \text{ u } \{1, 2, \ldots, n\} \text{ nije 1-1.}$$

Za koje vrednosti m, n je tvrdnja $\mathrm{PHP}_{m,n}$ tačna?

Za funkciju $f : \{1, 2, \ldots, m\} \to \{1, 2, \ldots, n\}$ uvodimo slova f_{ij}, $1 \leq i \leq m$, $1 \leq j \leq n$, sa značenjem $f(i) = j$. Tvrdnja $(\mathrm{PHP}_{m,n})$ se može iskazati sledećom formulom:

$$\bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{n} f_{ij} \wedge \bigwedge_{1 \leq j < j' \leq n} \neg(f_{ij} \wedge f_{ij'}) \right) \Rightarrow \bigvee_{1 \leq i < i' \leq m} \bigvee_{j=1}^{n} (f_{ij} \wedge f_{i'j})$$

ZADATAK. Za koje vrednosti m i n je formula $\mathrm{PHP}_{m,n}$ tautologija?



Sadržaj

1 Kratka istorija matematičke logike 1

2 Iskazna logika 7
2.1 Sintaksa iskazne logike 7
2.2 Semantika iskazne logike 10
2.2.1 Valuacija, interpretacija, model; zadovoljive, valjane,porecive i kontradiktorne formule 10
2.2.2 Istinitosne tablice 13
2.2.3 Logičke posledice, logički ekvivalentne formule,supstitucija 14
2.2.4 Potpuni skupovi veznika 21
2.2.5 Normalne forme 23
2.2.6 Dejvis–Patnam–Logman–Lovelandova procedura 28
2.2.7 Metod rezolucije 31
2.2.8 Metod tabloa 39
2.2.9 Teorema o kompaktnosti za iskaznu logiku 47
2.3 Sistemi za dedukciju u iskaznoj logici 49
2.3.1 Hilbertov sistem 52
2.3.2 Prirodna dedukcija 64
2.3.3 Račun sekvenata 70
2.4 Sažetak 80

3 Logika prvog reda 83
3.1 Sintaksa logike prvog reda 84
3.2 Semantika logike prvog reda 88
3.2.1 Valuacija, interpretacija, model; zadovoljive, valjane,porecive i kontradiktorne formule 88
3.2.2 Logičke posledice, logički ekvivalentne formule,supstitucija 93
3.2.3 Normalne forme 100
3.2.4 Erbranova teorema 107
3.2.5 Unifikacija 115
3.2.6 Metod rezolucije 119
3.2.7 Metod tabloa 135
3.3 Sistemi za dedukciju u logici prvog reda 145
3.3.1 Hilbertov sistem 146
3.3.2 Prirodna dedukcija 153
3.3.3 Račun sekvenata 155

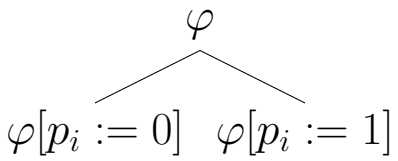
Autorsko izdanje

ZADATAK. Dokazati da je formula $(p \Rightarrow q) \wedge \neg q \Rightarrow \neg p$ tautologija.

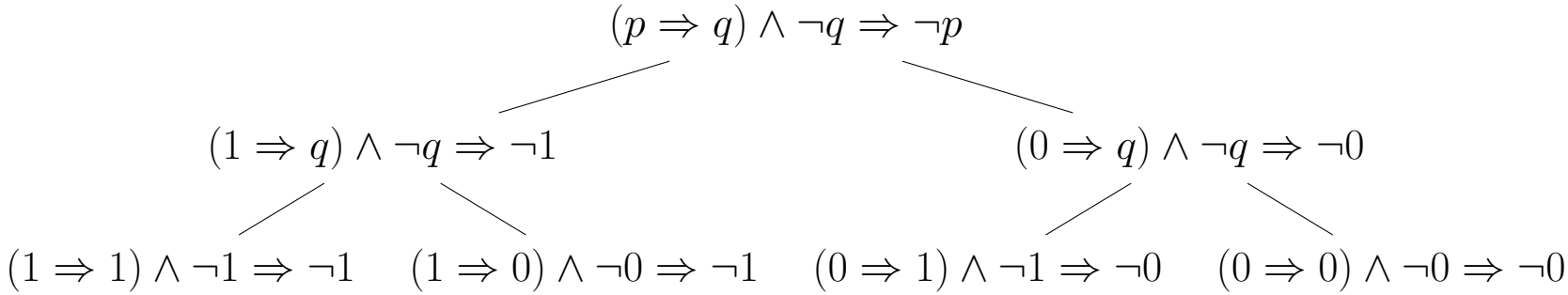
I NAČIN:

$(p$	$\Rightarrow$	$q)$	$\wedge$	$\neg$	q	$\Rightarrow$	$\neg$	p
0	1	0	1	1	0	**1**	1	0
0	1	1	0	0	1	**1**	1	0
1	0	0	0	1	0	**1**	0	1
1	1	1	0	0	1	**1**	0	1

Formiranje tablica se može prikazati i binarnim drvetom u čijem se korenu nalaza data formula, i primenjuje se "pravilo" dok god ima iskaznih slova



Na primer, istinitnosno drvo date formule je:



II NAČIN: Pretpostavimo da formula $(p \Rightarrow q) \wedge \neg q \Rightarrow \neg p$ nije tautologija, tj. da je za neku valuaciju (ili, sledeća jednačina ima rešenja):

$$(p \Rightarrow q) \wedge \neg q \Rightarrow \neg p = 0.$$

Tada je $(p \Rightarrow q) \wedge \neg q$ **tačno**, a $\neg p$ **netačno**, odn. p je **tačno**. Dalje zaključujemo da je i $p \Rightarrow q$ **tačno** i $\neg q$ **tačno**, odn. q je **netačno**. Najzad, iz tačnosti formule $p \Rightarrow q$ sledi da je p je **netačno** ili da je q je **netačno**, što je kontradikcija. Dakle, data formula jeste tautologija.

Ovaj način je zasnovan na primeni "pravila" koja su samo drugačija formulacija istinitosnih tablica logičkih veznika:

$(\neg\, \mathbf{1})$ Ako je $\neg\alpha$ **tačno**, onda je α **netačno**;

$(\neg\, \mathbf{0})$ Ako je $\neg\alpha$ **netačno**, onda je α **tačno**;

$(\wedge\, \mathbf{1})$ Ako je $\alpha \wedge \beta$ **tačno**, onda je α **tačno** i β je **tačno**;

$(\wedge\, \mathbf{0})$ Ako je $\alpha \wedge \beta$ **netačno**, onda je α **netačno** ili je β **netačno**;

$(\vee\, \mathbf{1})$ Ako je $\alpha \vee \beta$ **tačno**, onda je α **tačno** ili je β **tačno**;

$(\vee\, \mathbf{0})$ Ako je $\alpha \vee \beta$ **netačno**, onda je α **netačno** i β je **netačno**;

$(\Rightarrow\, \mathbf{1})$ Ako je $\alpha \Rightarrow \beta$ **tačno**, onda je α **netačno** ili je β **tačno**;

$(\Rightarrow\, \mathbf{0})$ Ako je $\alpha \Rightarrow \beta$ **netačno**, onda je α **tačno** i β je **netačno**.

$(\neg\,\mathbf{1})$ Ako je $\neg\alpha$ **tačno**, onda je α **netačno**;

$(\neg\,\mathbf{0})$ Ako je $\neg\alpha$ **netačno**, onda je α **tačno**;

$(\wedge\,\mathbf{1})$ Ako je $\alpha \wedge \beta$ **tačno**, onda je α **tačno** i β je **tačno**;

$(\wedge\,\mathbf{0})$ Ako je $\alpha \wedge \beta$ **netačno**, onda je α **netačno** ili je β **netačno**;

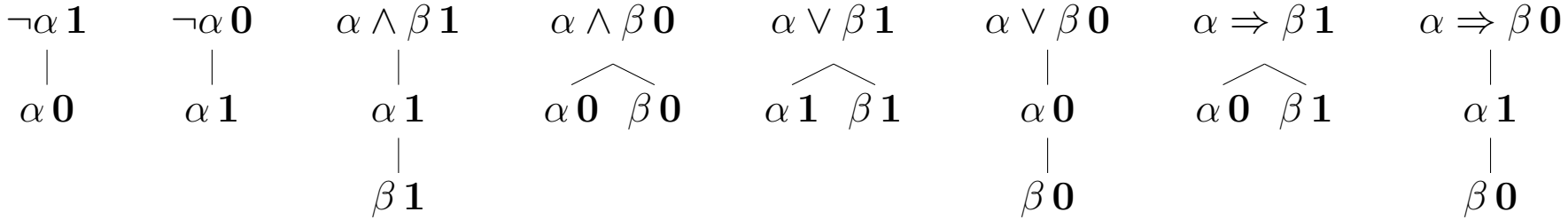
$(\vee\,\mathbf{1})$ Ako je $\alpha \vee \beta$ **tačno**, onda je α **tačno** ili je β **tačno**;

$(\vee\,\mathbf{0})$ Ako je $\alpha \vee \beta$ **netačno**, onda je α **netačno** i β je **netačno**;

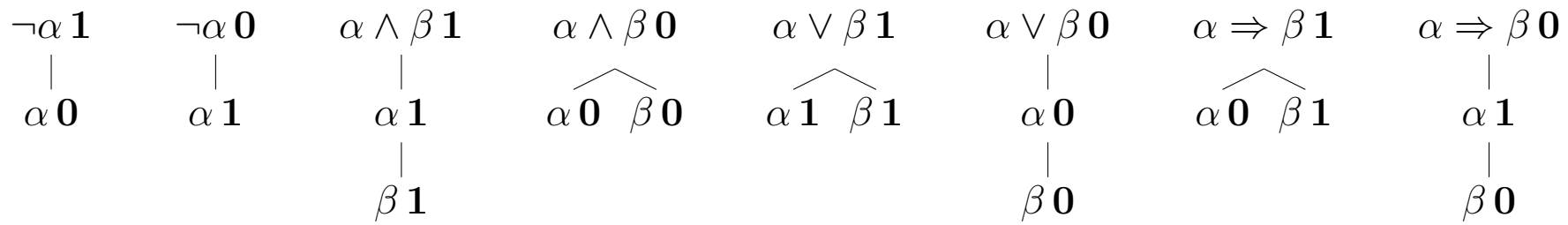
$(\Rightarrow\,\mathbf{1})$ Ako je $\alpha \Rightarrow \beta$ **tačno**, onda je α **netačno** ili je β **tačno**;

$(\Rightarrow\,\mathbf{0})$ Ako je $\alpha \Rightarrow \beta$ **netačno**, onda je α **tačno** i β je **netačno**.

Ako izjave '*formula* φ *je tačna*' i '*formula* φ *je netačna*' kraće označimo $\varphi\,\mathbf{1}$ i $\varphi\,\mathbf{0}$, gornja pravila možemo slikovitije prikazati na sledeći način:

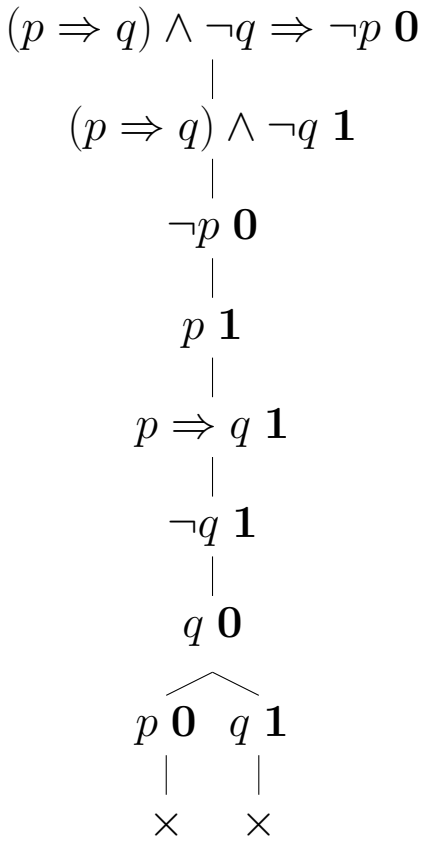


II NAČIN:



Pretpostavimo da formula $(p \Rightarrow q) \wedge \neg q \Rightarrow \neg p$ nije tautologija, tj. da postoji neka valuaciju za koju je $(p \Rightarrow q) \wedge \neg q \Rightarrow \neg p = 0$ **netačno**.

Tada je $(p \Rightarrow q) \wedge \neg q$ **tačno**, a $\neg p$ **netačno**, odn. p je **tačno**. Dalje zaključujemo da je i $p \Rightarrow q$ **tačno** i $\neg q$ **tačno**, odn. q je **netačno**. Najzad, iz tačnosti formule $p \Rightarrow q$ sledi da je p je **netačno** ili da je q je **netačno**, što je kontradikcija. Dakle, data formula jeste tautologija.



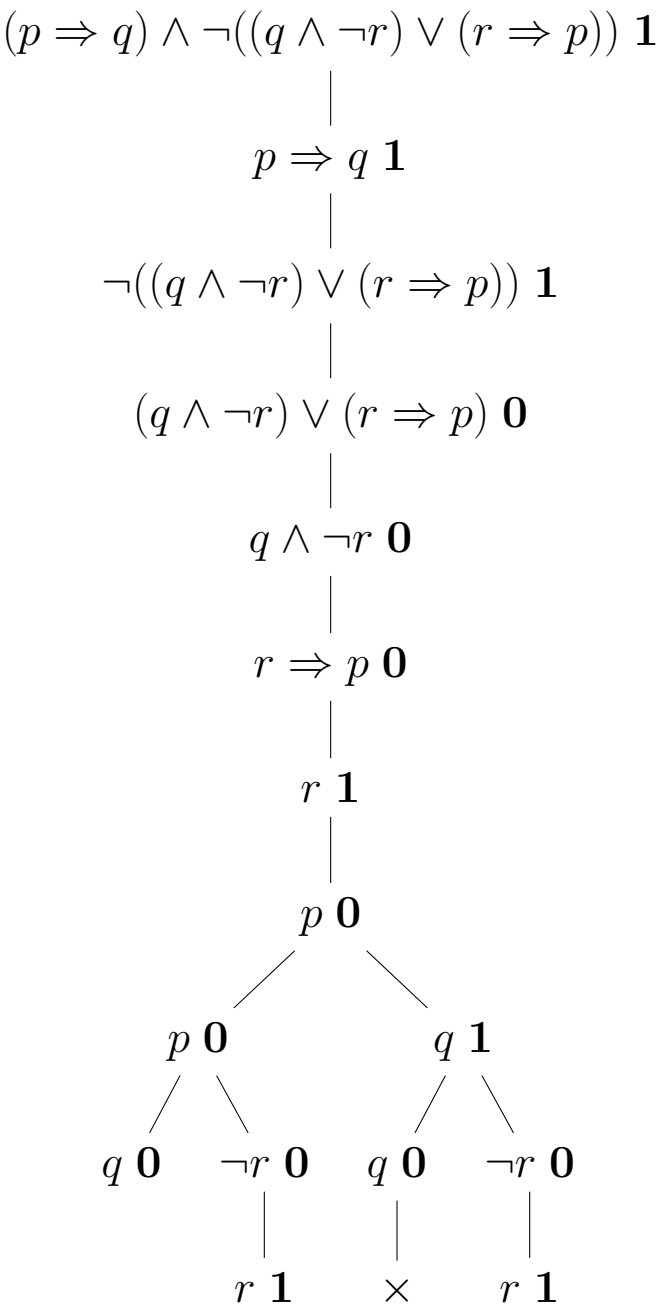
ZADATAK. Rešiti jednačinu

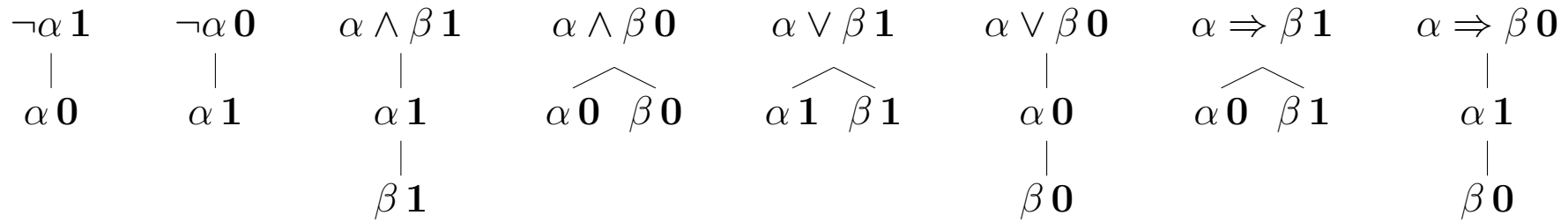
$$(p \Rightarrow q) \wedge \neg((q \wedge \neg r) \vee (r \Rightarrow p)) = 1.$$

Alternativno, ispitati da li je formula zadovoljiva

$$(p \Rightarrow q) \wedge \neg((q \wedge \neg r) \vee (r \Rightarrow p)).$$

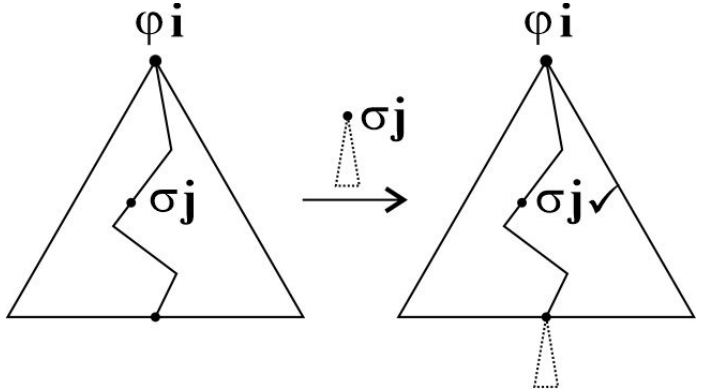
$($	p	$\Rightarrow$	q	$)$	$\wedge$	$\neg$	$(($	q	$\wedge$	$\neg$	r	$)$	$\vee$	$($	r	$\Rightarrow$	p	$))$
	0	1	0		0	0		0	0	1	0		1		0	1	0	
	0	1	0		1	1		0	0	0	1		0		1	0	0	
	0	1	1		0	0		1	1	1	0		1		0	1	0	
	0	1	1		1	1		1	0	0	1		0		1	0	0	
	1	0	0		0	0		0	0	1	0		1		0	1	1	
	1	0	0		0	0		0	0	0	1		1		1	1	1	
	1	1	1		0	0		1	1	1	0		1		0	1	1	
	1	1	1		0	0		1	0	0	1		1		1	1	1	





Prethodna pravila koristimo kao gradivne-blokove tzv. tablo-drveta koje u korenu ima neku "jednačinu" oblika $\varphi\ \mathbf{1}$ ili $\varphi\ \mathbf{0}$. Delimično izgrađeno tablo-drvo proširujemo tako što uočavamo čvor označen jednačinom $\sigma\,\mathbf{j}$ na koju ranije nije bilo primenjeno nijedno pravilo i σ nije iskazno slovo, a na list grane na kojoj se nalazi taj čvor nadovezujemo listove pravila u čijem se korenu nalazi $\sigma\,\mathbf{j}$. nakon tog koraka je korisno, pored jednačine $\sigma\,\mathbf{j}$ staviti zank ✓, da bi se naglasilo da je formula iskorišćena. Takođe, korisno je i ispred svake dopisane formule navesti njen redni broj, a iza navesti razlog zbog koga je dodata.

Izgradnja $\varphi\ \mathbf{i}$-drveta:



Kažemo da je grana *završena* ukoliko ne sadrži čvorove označene jednačinama na koje može primeniti neko od navedenih pravila, tj. kada kao neiskorišćene ostanu samo jednačine sa iskaznim slovima. Tablo-drvo je završeno kada su završene sve njegove grane.

Nakon primene svakog pravila, složenost formula koje se pojavljuju u dodatim jednačinama se smanjuje, odakle zaključujemo da posle konačno mnogo koraka izgradnja tablo-drveta, za bilo koji koren, završava.

Korektnost i potpunost metode tabloa.

Valuacija v zadovoljava jednačinu (jeste rešenje jednačine) φ **i** ako je $\varphi[v] = i$.

TEOREMA 1. Ako je v koja zadovoljava jednačinu φ **i**, onda u završenom tablo-drvetu čiji je koren φ **i** postoji grana takva da v zadovoljava sve jednačine na toj grani.

IDEJA DOKAZA. Za svako pravilo važi: ako valuacija v zadovoljava "roditelja", onda zadovoljava i sve potomke lančastih pravilima, odn. bar jednog potomka u pravilima grananja.

DEFINICIJA. Grana završenog tablo-drveta je zatvorena ukoliko na njoj postoje čvorovi označeni jednačinama p **0** i p **1**, za neko iskazno slovo p. U suprotnom, grana je otvorena.

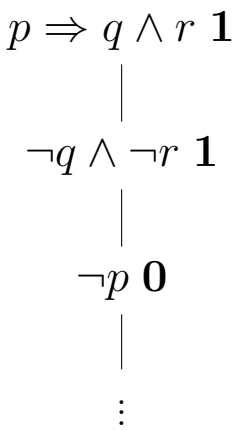
TEOREMA 2. Ukoliko tablo-drvo u čijem korenu je jednačina φ **i** sadrži otvorenu granu, onda postoji valuacija v koja zadovoljava jednačinu φ **i**.

IDEJA DOKAZA. Neka su p_{i_1} $\mathbf{i_1}$, ..., p_{i_k} $\mathbf{i_k}$ sve jednačine u kojima se pojavljuju iskazna slova i koje se pojavljuju na otvorenoj grani tablo-drveta. Definišimo valuaciju $v(p_{i_j}) = i_j$, $1 \leq j \leq k$, uzimajući da je $v(p)$ proizvoljno kada p nije među izdvojenim slovima. Tvrđenje sledi iz sledećih činjenica o pravilima:

- ako neka valuacija zadovoljava sve potomke lančastog pravila, onda ta valuacija zadovoljava i roditelja tog pravila;
- ako neka valuacija zadovoljava bar jednog potomka pravila grananja, onda ta valuacija zadovoljava i roditelja tog pravila.

Zadatak. Rešiti sistem (naći sva rešenja ili pokazati da nema rešenja):

$$p \Rightarrow q \wedge r = 1, \neg q \wedge \neg r = 1, \neg p = 0$$



(1) $p \Rightarrow q \wedge r$ **1**
(2) $\neg q \wedge \neg r$ **1**
(3) $\neg p$ **0**
(4) p **1** [$\neg$ **0** (3)]
(5) $\neg q$ **1** [$\wedge$ **1** (2)]
(6) $\neg r$ **1** [$\wedge$ **1** (2)]
(7) q **0** [$\neg$ **1** (5)]
(8) r **0** [$\neg$ **1** (6)]
(9) p **0** [$\Rightarrow$ **1** (1)] (10) $q \vee r$ **1** [$\Rightarrow$ **1** (1)]
(11) $\times$ [(4-9)] (12) q **1** [$\vee$ **1** (10)] (13) r **1** [$\vee$ **1** (10)]
(14) $\times$ [(7-12)] (15) $\times$ [(8-13)]

Sistem nema rešenja.

Zadatak. Rešiti sistem (naći sva rešenja ili pokazati da nema rešenja):

$$p \vee q \Rightarrow r = 0, r \Rightarrow s = 1, \neg q \wedge \neg s = 1, \neg p = 0$$

(1) $p \vee q \Rightarrow r$ **0**
(2) $r \Rightarrow s$ **1**
(3) $\neg q \wedge \neg s$ **1**
(4) $\neg p$ **0**
(5) p **1** [$\neg$ **0** (4)]
(6) $\neg q$ **1** [$\wedge$ **1** (3)]
(7) $\neg s$ **1** [$\wedge$ **1** (3)]
(8) q **0** [$\neg$ **1** (6)]
(9) r **0** [$\neg$ **1** (7)]
(10) $p \vee q$ **1** [$\Rightarrow$ **0** (1)]
(11) r **0** [$\Rightarrow$ **0** (1)]
(12) p **1** [$\wedge$ **1** (10)] (13) q **1** [$\wedge$ **1** (10)]
(14) $\times$ [8-13]

Sistem ima rešenja i jedino rešenje je $(p, q, r) = (1, 0, 0)$.