ISSN 2303-5366
http://bmc.pfb.ues.rs.ba/

БМЧ
Бијељински методички часопис
Вол. 7 (2020), 15-29
UDK
510.6:004.42
164.1
Оригинални научни чланак

METOD REZOLUCIJE I METOD TABLOA U ISKAZNOJ LOGICI

Dragica Milinković[1]
Slađana Mitrović
Univerzitet u Istočnom Sarajevu, Pedagoški fakultet u Bijeljini

Sažetak: *U ovom radu prikazaćemo značaj metoda rezolucije i metoda tabloa u iskaznoj logici. Metodom rezolucije ispitujemo da li je zadati skup klauza zadovoljiv ili nije, a metod tabloa koristimo za pobijanje iskazne formule, tj. za utvrđivanje njene nezadovoljivosti.*

Ključne reči*: metod rezolucije, metod tabloa, iskazna logika.*

Uvod

Ponekad je teško pronaći dokazni niz u iskaznom računu *L* za neku formulu, jer dokazni niz se ne vidi iz same formule koju treba dokazati, nego se često treba nečega dosetiti. Prema tome, nije lako izvršiti na računaru traganje za dokazom u Hilbertovom deduktivnom sistemu *L*. Metod rezolucije je postupak za dokazivanje da li je neka iskazna formula zadovoljiva, a koji se može lakše implementirati na računaru. Kako je formula tautologija, ako i samo ako njena negacija nije zadovoljiva, metod rezolucije se može koristiti za dokazivanje da li je neka formula tautologija. Formulisao ga je Alan Robinson 1965. godine (u radu pod naslovom "*A machine oriented logic based on the resolution principle*"), oslanjajući se na mnogobrojne rezultate koji su prethodili tom otkriću. On se primenjuje za formule koje su u konjunktivnoj normalnoj formi (KNF), tj. za formule koje su konjunkcije nekih disjunkcija (*Janičić, 2008).* Metod semantičkih tabloa prvi je opisao Evert 1955. godine, a dalje ga je razvio Rejmond Smaljan 1971. godine. On se često naziva i metodom analitičkih tabloa. Za razliku od metode rezolucije, metoda analitičkih tabloa se primenjuje u većini logika koje se koriste u računarstvu. U postupku izvođenja se koriste samo podformule polaznih formula, odakle potiče pridev analitički u imenu metode (*Janičić, 2008*).

[1] dragica.milinkovic@pfb.ues.rs.ba

Metod rezolucije

Za svaku iskaznu formulu postoji njoj ekvivalentna formula koja je u konjunktivnoj normalnoj formi. Algoritam za dobijanje konjunktivne normalne forme neke iskazne formule ima sledeće korake:

- eliminisati veznike $\Rightarrow$ i $\Leftrightarrow$ korišćenjem relacija:

$A \Rightarrow B \equiv \neg A \vee B$ i $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$

- relacije $\neg(A \wedge B) \equiv (\neg A \vee \neg B)$ i $\neg(A \vee B) \equiv (\neg A \wedge \neg B)$koristiti s leva na desno korišćenjem De Morganovih zakona
- eliminisati duple negacije $\neg\neg A \equiv A$
- koristiti zakon distributivnosti $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

Definicija 1: neka je S skup iskaznih slova. Literal je svako iskazno slovo ili negacija iskaznog slova, tj. svako p odnosno $\neg p$, gde $p \in S$.
Definicija 2: literali $L1$ i $L2$ su *komplementarni*, ako je jedan od njih atomička formula, a drugi njena negacija.
Definicija 3: klauza je konačan skup literala koji ne sadrži suprotne literale. Neprazna klauza C predstavlja formulu koja se dobija disjunkcijom literala iz C. Jedinična klauza je klauza koja sadrži tačno jedan literal.

Prazna klauza, u oznaci $\emptyset$, je klauza koja ne sadrži ni jedan literal i predstavlja proizvoljnu kontradikciju. Neka je $\mathrm{S} = \mathrm{C}_1, \mathrm{C}_2, \dots, \mathrm{C}_\mathrm{n}$ skup klauza, i neka klauze $\mathrm{C}_{1,}\mathrm{C}_2, \dots, \mathrm{C}_\mathrm{n}$ predstavljaju redom formule $\mathrm{A}_1, \mathrm{A}_2, \dots, \mathrm{A}_\mathrm{n}$. Tada skup klauza S predstavlja formulu $\mathrm{A}_1 \wedge \mathrm{A}_2 \wedge \dots \wedge \mathrm{A}_\mathrm{n}$ i obrnuto, kažemo da je S klauzalna forma formule $\mathrm{A}_1 \wedge \mathrm{A}_2 \wedge \dots \wedge \mathrm{A}_\mathrm{n}$.
Pošto je klauza disjunkcija literala, neka interpretacija I zadovoljava klauzu C ako zadovoljava bar jedan njen literal, pa prazna klauza nikada nije zadovoljena. Skup klauza je zadovoljen pri interpretaciji I ako su pri I sve klauze tačne. Prazan skup klauza je uvek tačan. Dva skupa klauza S_1 i S_2 su ekvivalentna, ako ih zadovoljavaju iste interpretacije. Ponekad će biti pogodno klauzu shvatiti kao skup literala, tako da se umesto $\mathrm{p} \vee \neg\mathrm{q}$ piše samo $\{\mathrm{p}, \neg\mathrm{q}\}$. Pri tome se podrazumeva da to nije nikakav nov pojam, već samo zgodniji zapis, kao i da je skup literala koji čine klauzu zadovoljiv ako je zadovoljiv bar jedan njegov član.
Primećujemo da se ovde razmatraju samo formule specijalnog oblika, tj klauze. Takođe, u ovom sistemu nema aksioma, već se izvođenje svodi na primenu jednog jedinog pravila.

Pravilo rezolucije

Metod rezolucije se sastoji od niza primena pravila rezolucije na skup klauzula. Pravilo rezolucije čuva zadovoljivost, tj. ako je skup klauzula S bio zadovoljiv, onda će biti zadovoljiv i skup klauzula koji se od S dobija pravilom rezolucije. Prema tome, ako se metodom rezolucije dobije skup klauzula koji sadrži praznu klauzulu (koja je nezadovoljiva), onda mora biti nezadovoljiv i polazni skup klauzula. Pravilo rezolucije dato je sledećom definicijom.

Definicija 4: neka su C_1 i C_2 klauze i neka su L_1 i L_2 komplementarni literali, takvi da se L_1 nalazi u C_1, a L_2 u C_2. Rezolventa klauza C_1 i C_2 po literalima L_1 i L_2 je klauza $\mathrm{Res}(C_1, C_2, L_1, L_2) = (C_1 \backslash \{L_1\}) \cup (C_2 \backslash \{L_2\})$ dobijena tako što je iz C_1 izbrisan L_1, a iz C_2 izbrisan L_2, i preostali literali povezani disjunkcijom. Klauze C_1 i C_2 su roditelji, a rezolventa njihov potomak.
Primer 1:
Rezolucijom klauza $(p \vee \neg q)$ i $(q \vee r)$ dobija se klauza $(p \vee r)$.
Rezolucijom klauza $(\neg p \vee q \vee r)$ i $(\neg q \vee s)$ dobija se klauza $(\neg p \vee r \vee s)$.
Rezolucija se ne može primeniti na klauze $(\mathrm{p} \vee \neg \mathrm{q})$ i $(\neg \mathrm{q} \vee \mathrm{r})$.

U sledećoj teoremi ćemo pokazati da, kada skup klauza proširimo nekom rezolventom klauza iz skupa, dobijeni skup ne sadrži nove informacije u odnosu na polazni skup, zapravo da su ta dva skupa međusobno ekvivalentna. Odatle je niz primena rezolucije logički ispravan postupak zaključivanja.

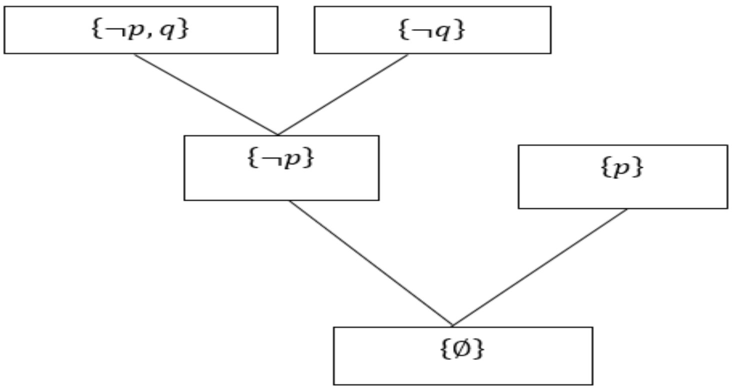
Teorema 1: Neka su C_1 i C_2 dve klauze i R njihova rezolventa. Tada je:
$\{C_1, C_2\} \simeq \{C_1, C_2, R\}$.
Dokaz: Neka je I interpretacija za koju je $I \vDash \{C_1, C_2, R\}$. Tada je trivijalno i $I \vDash \{C_1, C_2\}$.
Obrnuto, neka je I interpetacija za koju je $I \vDash \{C_1, C_2\}$, tj. $I \vDash C_1$ i $I \vDash C_2$. Neka su L i $\neg L$ literali po kojima se vrši rezolucija klauza C_1 i C_2, takvi da je: $L \in C_1$ i $\neg L \in C_2$. Ako $I \vDash L$, onda $I \vDash C_2 \backslash \{\neg L\}$, pa $I \vDash R$. Slično je i za $I \vDash \neg L$. Dakle, $I \vDash \{C_1, C_2, R\}$.

Definicija 5: neka je S konačan skup klauzula. Kažemo da je klauzula C rezolventni izvod iz skupa S ako postoji konačan niz klauzula $C_1, C_2, \dots, C_k$ gde je $C_k = C$, takav da svaka klauzula tog niza pripada skupu S ili se dobija pravilom rezolucije od ranijih formula u nizu. U tom slučaju takođe kažemo da je skup rezolventnih izvoda iz S dobijen metodom rezolucije iz skupa S.
Formalni sistem zasnovan na pravilu rezolucije počiva na sledećem tvrđenju:

Teorema 2: (*potpunost i saglasnost metoda rezolucije*) Skup klauza$S$ je kontradiktoran ako i samo ako je prazna klauza u skupu $Res(S)$.
Dokaz: prema teoremi 1 je $Res(F) \simeq F$.
Ako je $\emptyset \in Res(F)$, ovaj skup je kontradiktoran, pošto nijedna interpretacija ne zadovoljava praznu klauzu.
Ako je skup F nezadovoljiv, to je i skup Res(F). Na osnovu teoreme o kompaktnosti (skup formula T je zadovoljiv ako i samo ako je svaki njegov konačan podskup zadovoljiv.) iz skupa Res(F) se može izabrati konačan nezadovoljiv podskup $\mathrm{S} = \{\mathrm{C}_1, \mathrm{C}_2, \dots, \mathrm{C}_\mathrm{m}\}$. Označimo sa $|\mathrm{C}|$ broj literala u klauzi C.
Tvrđenje ćemo dokazati indukcijom po broju $\mathrm{n} = |\mathrm{C}_1| + \cdots + |\mathrm{C}_\mathrm{m}| - \mathrm{m}$. Ako je $n = 0$, moguće je da su ili sve klauze jednočlane, ili da je neka klauza prazna. U drugom slučaju tvrđenje je dokazano, dok u prvom, pošto je S nezadovoljiv, moraju postojati dve klauze čiji su literali komplementarni. Primenom pravila

rezolucije na ove dve klauze izvodi se prazna klauza, pa je $\emptyset \in Res^*(F)$. Neka indukcijska pretpostavka važi za svako $n < k$ i neka je $n = |C_1| + \cdots + |C_m| - m = k$. Sada bar jedna od klauza sadrži bar dva literala, neka je to C_i, ona je oblika $C_i^{'} \cup C_i^{''}$, gde su oni neprazni skupovi literala. Posmatrajmo skup klauza $\{C_1, \ldots, C_{i-1}, C_i^{'}, C_{i+1}, \ldots, C_m\}$. Kako je skup S nezadovoljiv, nezadovoljiv je i ovaj skup, ali je za njega $|C_1| + \cdots + |C_m| - m < k$, pa se iz ovog skupa klauza može izvesti prazna klauza. Ako u tom izvođenju nije učestvovala klauza $C_i^{'}$, isto važi za polazni skup *S*. Ako jeste učestvovala u izvođenju prazne klauze, razmotrimo izvođenje u kome se umesto nje koristiC_i. To bi bio dokaz za$C_i^{'}$. Ponovnom primenom indukcijske pretpostavke na skup $\{C_1, \ldots, C_{i-1}, C_i^{''}, C_{i+1}, C_m\}$ zaključujemo da se i iz ovog skupa izvodi prazna klauza. Odatle se i iz polaznog skupa klauza izvodi *S* prazna klauza.

Primer 2:
Neka je skup klauza $F = \{\{\neg p, q\}, \{\neg q\}, \{p\}\}$. Rezolventa prve dve klauze skupa je klauza $\neg p$. Rezolviranjem ove i poslednje klauze iz skupa *F* izvodi se prazna klauza. Prema Teoremi 2 skup klauza *F* je kontradiktoran.



Slika 1: Rezolucijsko izvođenje iz primera

Primer 3: Metodom rezolucije se iz skupa $\{\{\neg p\}, \{\neg q\}, \{r\}\}$ne može izvesti prazna klauza. Ovaj skup klauza je, dakle, zadovoljiv.
Rezolucijsko izvođenje se ponekad prikazuje u formi drveta. Izvođenje prazne klauze u primeru 2 je prikazano na Slici 1.

Teorema 3: (saglasnost pravila rezolucije) Neka je skup klauza S' dobijen od skupa klauza *S* primenom pravila rezolucije. Ako neka valuacija zadovoljava skup *S*, onda ona zadovoljava i skup S'.
Dokaz: pretpostavimo da valuacija *v* zadovoljava sve klauze iz skupa *S*. Dokažimo da ona zadovoljava i sve klauze iz skupa *S'*. Pretpostavimo da je pravilo rezolucije

primenjeno na klauze $\mathrm{C_a}$ i $\mathrm{C_b}$ i da je njihova rezolventa klauza $\mathrm{C_r}$. Jedina klauza koja pripada skupu $\mathrm{C}^{'}$, a moguće ne pripada skupu S je klauza $\mathrm{C_r}$. Dokažimo da valuacija v zadovoljava klauzu $\mathrm{C_r}$. Pretpostavimo da klauza $\mathrm{C_a}$ sadrži literal l, a klauza $\mathrm{C_b}$ literal $\bar{l}$. Ako je literal l tačan u valuaciji v, onda je literal $\bar{l}$ netačan, pa u klauzi mora da postoji neki literal (različit od $\bar{l}$) koji je tačan u valuaciji v. Taj literal je element klauze $\mathrm{C_r}$, pa je i ona zadovoljiva u valuaciji v. Ako je literal l netačan u valuaciji v, onda u klauzi mora da postoji neki literal (različit od l) koji je tačan u valuaciji v. Dakle, svaka klauza iz skupa $\mathrm{S}^{'}$ tačna u valuaciji v, odakle sledi tvrđenje teoreme.

Na prvi pogled rezolucija nije pogodna za rad sa iskaznim formulama proizvoljnog oblika. Međutim, kao što je pokazano, svaka iskazna formula ima ekvivalentnu formulu u konjunktivnoj normalnoj formi, odnosno definicionu formu koja je zadovoljiva ako i samo ako je zadovoljiva i sama formula. Na osnovu prethodnog tvrđenja i činjenice da je formula A zadovoljiva pri nekoj interpretaciji I ako i samo ako ¬A nije zadovoljivo pri interpretaciji I, moguće je ispitivanje valjanosti, odnosno zadovoljivosti svih formula iskazne logike. Ako se želi ispitati valjanost formule A, ona se najpre negira, potom se tako dobijena formula prevede u konjuktivnu normalnu formu ili definicionu formu i potom primenom rezolucije pokušava izvođenje prazne klauze. Ako se u tome uspe, skup klauza koji odgovara KNF(¬A), odnosno definicionoj formi formule ¬A, je kontradiktoran, pa je kontradiktorna i formula ¬A, odnosno A je valjana. Da li je formula zadovoljiva ili kontradikcija ispituje se analognim postupkom u kome se polazi od KNF(A) ili definicione forme formule A. Ako se rezolucijom dobije prazna klauza, formula A je kontradikcija, a u suprotnom je zadovoljiva.

Primer 4: negiranjem formule $p \Rightarrow (q \Rightarrow p)$ i njenim prevođenjem u KNF, dobija se skup klauza $F = \{p, q, \neg p\}$. Primenom pravila rezolucije na prvu i treću klauzu skupa izvodi se prazna klauza, pa je formula $p \Rightarrow (q \Rightarrow p)$ valjana.

Zaustavljanje metoda rezolucije

Ako je skup klauza F konačan, pokazaćemo u narednoj teoremi, da će i skup $Res(F)$ biti takođe konačan, pa se može dobiti u konačno mnogo koraka izvođenja. Ovo je bitno, pošto garantuje da će se, ako računar ima dovoljno memorije, uvek dobiti odgovor na postavljeno pitanje (zadovoljivost, odnosno valjanost ulazne formule), što je još jedan dokaz odlučivosti tih problema za iskaznu logiku.

Teorema 4: (zaustavljanje metoda rezolucije) Metod rezolucije se zaustavlja.
Dokaz: U svakom koraku metoda rezolucije tekućem skupu se dodaje nova klauza. Klauze su skupovi literala, pa nije bitan poredak literala u njima, a višestruka pojavljivanja literala u klauzi nisu moguća. Nad skupom od n iskaznih slova ima $2n$ različitih literala (za svako iskazno slovo p postoje literali p i $\neg p$) i svaki od njih može da se pojavljuje ili ne pojavljuje u klauzi, te razlicitih klauza ima $2^{2n} = 4^n$. Dakle, metod rezolucije se zaustavlja u konačno mnogo koraka, jer se iz literala iz

skupa S može izvesti samo konačan broj (novih) klauza (taj broj međutim može biti ekponencijalno veliki (u funkciji broja literala u S) i primena procedure rezolucije može da se sastoji od veoma velikog broja koraka).

Programska realizacija metoda rezolucije

Jedna programska realizacija metode rezolucije za iskaznu logiku kojom se ispituje valjanost iskaznih formula sprovodi se sledećom procedurom:

```
procedure IskaznaRezolucija
begin
        Provesti negaciju ulazne formule A u KNF(¬A)
        ili definicionu formu D_1 ∧ D_2 ∧ ... ∧ D_n .
        F := {D_1, D_2, ...D_n}
        Res_0(F) := F
        Res_1(F) := R(Res_0(F))
        i := 1
        while((Res_i(F) ≠ Res_{i-1}(F)) and ∅ ∉ Res_i(F)) do
        begin
                i := i + 1
                Res_i(F) := R(Res_{i-1}(F))
        end
        if (∅ ∈ Res_i(F)) then
                Formula je valjana
        else
                Formula nije valjana
        end
end
```

gde je R funkcija koja kao argument dobija skup klauza E, a kao rezultat vraća skup E proširen klauzama koje se dobijaju rezolucijom dve klauze iz skupa E. Svaku klauzu možemo pretpostaviti kao listu literala, uređenih po nekom utvrđenom redosledu, a skup klauza kao listu klauza. U jednom pozivu precedure R rezolviraju se klauze koje počinju komplementarnim literalima (*Ognjanović i Krdžavac, 2004.*).

Metod tabloa

Pod pretpostavkom da imamo klasu modela kojom dajemo značenje formulama i neki formalni sistem koji karakteriše tu klasu modela, obično je pogodno koristiti aksiomatski sistem da bi se pokazalo da formula jeste valjana formula, dok je konstrukcija kontramodela, koja je semantički postupak, pogodnija da se pokaže da nešto nije valjana formula. U metodi analitičkih tabloa pokušava se objedinjavanje ova dva postupka, tj. formalno i sistematski se nastoji napraviti kontramodel za neku formulu, pa ako se u tome ne uspe, metoda garantuje da je formula valjana.

Formulacija metoda tabloa

U bilo kojoj (fiksnoj) valuaciji *v* važi:

1.
a) ako je formula ¬A tačna, onda je *A* netačna;
b) ako je formula ¬A netačna, onda je *A* tačna;

2.
a) ako je formula A ∧ B tačna, onda su *A* i *B* tačne;
b) ako je formula A ∧ B netačna, onda je netačna *A* ili je netačna *B*;

3.
a) ako je formula A ∨ B tačna, onda je tačna formula *A* ili je tačna formula *B*;
b) ako je formula A ∨ B netačna, onda su A i B netačne;

4.
a) ako je formula A ⇒ B tačna, onda je ili A netačna ili B tačna;
b) ako je formula A ⇒ B netačna, onda je A tačna i B netačna.

Definicija 6: neka se iskazni jezik sastoji od prebrojivog skupa *Φ* iskaznih slova, logičkih veznika $\wedge$, $\vee, \neg$ i $\Rightarrow$, i zagrada. Neka su *T* i *F* novi formalni simboli koji se ne nalaze u iskaznom jeziku. Ako je *A* iskazna formula, onda su *TA* i *FA* označene formule. Konjugat označene formule *TA* je formula *FA* i obratno. Označene formule shvatamo na sledeći način. Za proizvoljnu interpretaciju *I*, $I(TA) = I(A)$ i $I(FA) = I(A)$. Zapravo znak *T* ispred formule se može shvatiti kao "tačno je", dok se znak *F* može shvatiti kao "netačno je". U Tabeli 1. prikazaćemo podele na te dve grupe formula i njihove komponente α_1, α_2, β_1 i $\beta_2$formule.

Tabela 1: Formule α i β

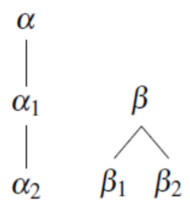
α	α_1	α_2
$T\ A \wedge B$	TA	TB
$F\ A \vee B$	FA	FB
$F\ A$⇒B	TA	FB
$T\ \neg A$	FA	FA
$F\ \neg A$	TA	TA

β	β_1	β_2
$F\ A \wedge B$	FA	FB
$TA \vee B$	TA	TB
$T\ A \Rightarrow B$	FA	TB

Očigledno je da se α formule ponašaju slično konjukciji, dok se β formule, ponašaju slično disjunkciji. Taj iskaz je dokazan sledećim tvrđenjem:

Teorema 5: za proizvoljnu interpretaciju I važi:
$I(\alpha) = T$ ako i samo ako $I(\alpha_1) = I(\alpha_2)$
$I(\beta) = T$ ako i samo ako $I(\beta_1) = I(\beta_2)$
Označena formula je tačna pri interpretaciji I ako i samo ako je njen konjugat netačan.

Uloga formulacije metode tabloa je da se mnogobrojni slučajevi složenih formula koje treba analizirati svode samo na dva, tj. na slučaj alfa i beta formula.



Slika 2: α i β pravilo

Tablo

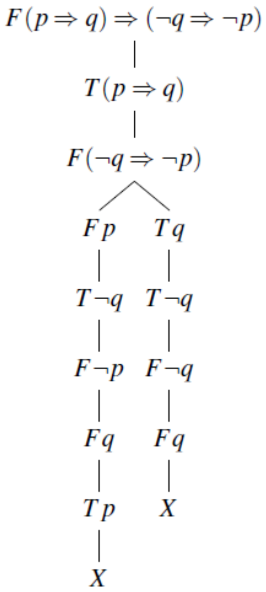
Definicija 7: analitički tablo za formulu A je uređeno stablo (tj. stablo u kome je skup sledbenika svakog čvora uređen) tako da je svakom čvoru pridružena neka označena formula. Ono se konstruiše iterativno na sledeći način:

- Korenu stabla se pridružuje označena formula *FA*.
- Ukoliko neka grana od korena do lista stabla sadrži označenu formulu koja još nije iskorišćena, odnosno, nije na nju primenjeno odgovarajuće pravilo, u zavisnosti od podtipa formule, nastavak konstrukcije je moguće izvesti na dva načina:
  - α-pravilo: ako imamo neku α neiskorišćenu formulu na grani, tada se grana produžava sa jednim čvorom, i pridružuje mu se formula α_1, odnosno sa dva nova čvora od kojih jedan sadrži α_1 formulu, a drugi formulu α_2.

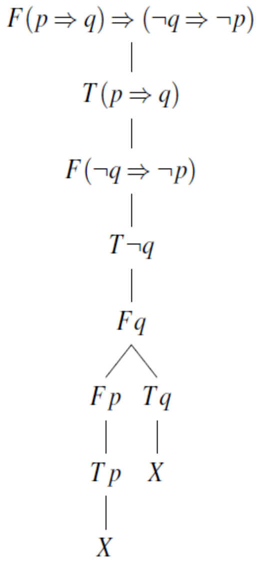
– β -pravilo: ako imamo neku β neiskorišćenu formula na grani, tada listu grane dodaju dva sledbenika, pri čemu levi sledbenik sadrži formulu β_1, a desni formulu β_2.

Možemo primetiti da u zavisnosti od toga kojim redosledom na neiskorišćene formule primenjujemo α i β pravila, za isto označenu formulu možemo konstruisati različita stabla. Jedan od načina za konstrukciju je *sistematična konstrukcija*, odnosno da pravila nikad ne primenjujemo na označenu formulu ukoliko nisu već iskorišćene sve formule koje su u istoj grani iznad nje. Drugi način je da damo prednost α- formulama, tj. da prvo primenjujemo α-pravila, na sve neiskorišćene formule, a tek nakon toga β -pravila. Sledeća dva primera ilustruju navedena pravila.

Primer 5: (*sistematična konstrukcija*) sistematično konstruisan tablo za formulu $(p \Rightarrow q) \Rightarrow (\neg q \Rightarrow \neg p)$ izgleda ovako:



Primer 6: (*prednost α-formula*) konstruišimo tablo formule $(p \Rightarrow q) \Rightarrow (\neg q \Rightarrow \neg p)$ dajući prednost α-formulama:



Posmatrajući prethodne primere, vidimo da je proces rastavljanja označene formule mnogo efikasniji ako damo prioritet α-formulama. Time različite grane tabloa ne sadrže iste delove. Dakle, ako imamo mogućnost izbora, biramo tu strategiju rastavljanja.

Definicija 8: grana tabloa je *zatvorena* ako se na njoj nalazi neka formula i njen konjugat, ili preciznije ako sadrži čvorove kojima su pridružene formule *TB* i *FB*. U suprotnom, kažemo da je grana tabloa *otvorena*. Zatvorenu granu tabloa obeležavamo sa *X*. Tablo je *zatvoren* ako je svaka njegova grana zatvorena.
Definicija 9: grana tabloa je *završena* ako su odgovarajuća pravila konstrukcije primenjena na sve čvorove sa te grane. Tablo je *završen* (kompletiran) ako mu je svaka grana završena.

Zaustavljanje metoda tabloa

Teorema 6: (*zaustavljanje metoda tabloa*) metod tabloa se zaustavlja.
Dokaz: Svaka označena formula u korenu stabla ima konačan broj simbola, a primenom pravila α i β , označene formule u novim čvorovima imaju bar jedan simbol manje od prethodne označene formule. Dakle, posle konačno mnogo koraka dobijamo tablo u kome nema neiskorišćenih označenih formula. Stoga, metod tabloa za svaku označenu formulu se uvek zaustavlja.

Korektnost i kompletnost metode tabloa

Definicija 10: grana tabloa je *zadovoljiva* ako postoji valuacija koja zadovoljava sve označene formule u toj grani. Tablo je *zadovoljiv* ako postoji valucija *v* takva da je bar jedna grana zadovoljiva pri toj valuaciji.
Definicija 11: za dva uređena tabloa T_1 i T_2 na čijim se čvorovima nalaze označene formule, kažemo da je T_2 direktno proširenje od T_1 , ako se T_2 dobija iz T_1 primenom jednog od α i β pravila.

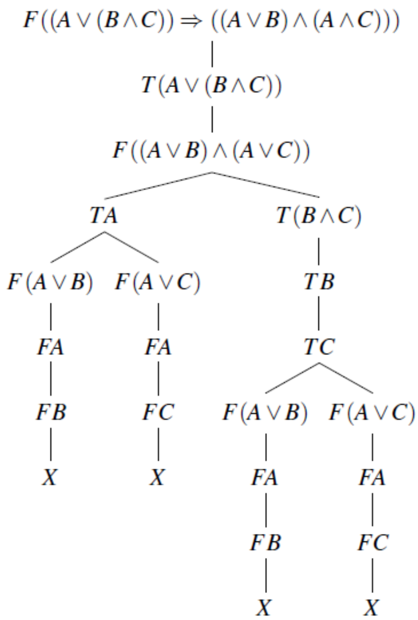
Dakle, tablo *T* je tablo za formulu *A* ako postoji konačan niz tabloa $T_1, T_2, \dots, T_n = T$, takvih da je T_1 tablo sa jednim čvorom (korenom) označen sa *FA*, a za svako$i \in \{1,2, \dots n-11\}$, T_{i+1} je direktno proširenje od T_i.

Teorema 7: ako je tablo T_1 zadovoljiv i T_2 njegovo direktno proširenje, onda je i tablo T_2 zadovoljiv.
*Dokaz:*neka je grana θ zadovoljiva u tablou T_1 za valuaciju *v*. Ako primena pravila α i β za dobijanje tabloa T_2 ne menja granu θ , tada T_2 ima granu θ, koja je zadovoljiva, pa na osnovu definicije sledi da je i tablo T_2 zadovoljiv. Sad ćemo razmotriti situaciju kad se primenom pravila menja grana θ. Razlikujemo dva slučaja:

- Ako je primenjeno pravilo tipa α na formulu α , onda su grani sukcesivno dodata dva nova čvora α_1 i α_2 ili je dodat čvor sa jednom formulom α_1 . Tada je formula α tačna u valuaciji *v* ako su tačne formule α_1 i α_2 u toj valuaciji. Analogno u slučaju sa jednim čvorom. Dakle, grana koja je dobijena od θ je takođe zadovoljiva, pa tablo T_2 ima zadovoljivu granu tj. T_2 je zadovoljiv.
- Ako je primenjeno pravilo tipa β na formulu β, onda su grani dodata dva nova čvora sa pridruženim označenim formulama β_1 i β_2. Znamo da za formule tipa β važi da je formula β tačna u valuaciji *v* ako je bar jedna od formula β_1 i β_2 tačna u toj valuaciji. Tada je bar jedan nastavak grane θ zadovoljiv, a time je zadovoljiv i tablo T_2.

Teorema 8: (*korektnost metoda tabloa*). Ako se neka iskazna formula *A* može dokazati metodom tabloa (pobijanjem), onda je formula *A* tautologija.
Dokaz: početni tablo za formulu *A* sadrži samo koren kome je pridružena formula *FA*. Na osnovu prethodnog tvrđenja znamo da ako je formula zadovoljiva onda svaki tablo dobijen od početnog tabloa ima zadovoljivu granu. Prema tome, ako je tablo zatvoren, što znači da nema ni jednu zadovoljivu granu, sledi da ni početna označena formula *FA* nije zadovoljiva, tj. formula $\neg A$ je nezadovoljiva, pa je formula *A* tautologija.

Primer 7: koristeći metod tabloa pokažimo da je formula $(A \vee (B \wedge C)) \Rightarrow ((A \vee B) \wedge (A \vee C))$ tautologija.



Sad nam ostaje da ispitamo da li važi i obrat prethodne teoreme, tj. da li se svaka taugologija može dokazati metodom tabloa. Ili preciznije, da li za svaku tautologiju A postoji bar jedan zatvoren tablo u čijem se korenu nalazi označena formula FA.

Definicija 12: kažemo da je skup iskaznih formula S *Hintikin* ili nadole zasićen ako zadovoljava sledeće uslove:

- H_1: ne postoji iskazna promenljiva p takva da i $Tp \in S$ i $Fp \in S$
- H_2: ako za neku α-formulu važi $\alpha \in S$, tada su i $\alpha_1, \alpha_2 \in S$, odnosno, odgovarajuća formula $\alpha_1 \in S$
- H_3: ako za neku β -formulu $\beta \in S$, tada $\beta_1 \in S$ ili $\beta_2 \in S$.

Iz toga možemo da zaključimo da ako je tablo kompletiran (završen), onda je skup označenih formula koje pripadaju svakoj otvorenoj grani tog tabloa Hintikin.

Lema 1:(*Hintikina lema*) svaki Hintikin skup označenih formula je zadovoljiv.
Dokaz: neka je S proizvoljan Hintikin skup označenih formula. Tražimo valuaciju v u kojoj je svaki element skupa S tačan. Za svako iskazno slovo koje se pojavljuje u bar jednoj formuli skupa S, definišemo istinitosnu vrednost na sledeći način:

- ako je $Tp \in S$, onda $v(p) = 1$
- ako je $Fp \in S$, onda $v(p) = 0$

- ako ni Tp ni Fp ne pripadaju skupu S, tada p uzima proizvoljnu vrednost, ali zbog određenosti uzimamo $v(p) = 1$.

Navedena definicija je korektna, jer ni za jedno iskazno slovo ne može da važi$Tp, Fp \in S$. Neka je složenost označene formule jednaka složenosti odgovarajuće neoznačene formule. Indukcijom po složenosti označenih formula dokazaćemo da valuacija v zadovoljava sve označene formule skupa S.

Baza: svaka označena formula složenosti 0 tj. označena iskazna promenljiva koja se nalazi u skupu S tačna je u svakoj valuaciji (valuaciju v smo definisali tako da bi to važilo).

Indukcijska hipoteza: pretpostavimo da tvrđenje važi za sve elemente skupa S čija je složenost manja od $n, (n > 0)$ tj. da valuacija v zadovoljava sve označene formule složenosti manje od n.

Indukcijski korak: dokazujemo da je u valuaciji v tačna označna formula A složenosti n. Kako je složenost označene formule bar 1, razlikujemo dva slučaja: ona je tipa α ili tipa β.
Neka je označena formula tipa α. Iz uslova H_2sledi da $\alpha_1, \alpha_2 \in S$, odnosno, odgovarajuća formula$\alpha_1 \in S$.Kako je složenost formula α_1 i α_2 manja od n, za njih važi indukcijska hipoteza, te su one tačne u valuaciji v.Iz toga sledi da je i formula α tačna u valuaciji v.
Neka je označena formula tipa β. Iz uslova H_3 sledi da za odgovarajuće formule β_1 i β_2 važi $\beta_1 \in S$ ili $\beta_2 \in S$. Kako je složenost formula β_1 i β_2 manja od n, bilo koja od njih pripada skupu S, te na osnovu indukcijske hipoteze, ona mora da bude tačna. Dakle, formula β je tačna u valuaciji v.
Dakle, opisanom indukcijom smo dokazali da je svaki element iz skupa S tačan u valuaciji v, pa je Hintikin skup S zadovoljiv.

Teorema 10: (*kompletnost metoda tabloa)* ako je neka iskazna formula A valjana, onda se ona može dokazati metodom tabloa, tj. ako je formula A tautologija, svaki završeni tablo za A mora biti zatvoren.
Dokaz: dokazaćemo kontrapozicijom tj. ako postoji neki završen tablo za formulu FA koji je otvoren, onda formula A nije tautologija. Pretpostavimo da postoji neki završen tablo sa korenom FA koji je otvoren. To znači da on ima bar jednu otvorenu granu tj. granu koja ne sadrži par konjugovanih označenih formula TB i FB. Kako je skup označenih formula koje pripadaju toj grani Hintikin, na osnovu prethodne leme sledi da je taj skup zadovoljiv. Prema toma, zadovoljiva je i označena formula FA, pa formula A nije tautologija.

Programska realizacija metode tabloa

Metoda analitičkih tabloa se može isprogramirati na sledeći način: formula A se predstavlja u obliku binarnog drveta čiji čvorovi sadrže logičke veznike, a listovi iskazna slova formule. Tablo se konstruiše kao binarno drvo čiji čvorovi sadrže pokazivače na delove drveta formule i oznake T i F, zavisno od toga koji prefiks ima formula koja odgovara čvoru. Ako se radi sekvencijalno, konstrukcija tabloa se odvija grana po grana u skladu sa pravilima izvođenja. Posle primene svakog pravila izvođenja proverava se da li je grana zatvorena ili završena. Ako je grana zatvorena, prelazi se na sledeću granu ako takva grana postoji. Ako je grana završena, a nije zatvorena, polazna formula nije valjana i eventualno se može prikazati njen kontramodel. Ako su sve grane zatvorene, formula je valjana. Memorijska reprezentacija svake grane podseća na stek na koji upravljački program dodaje nove čvorove, odnosno sa koga oslobađa nepotrebne čvorove prilikom prelaska na novu granu. Pogodnost ovakvog postupka je malo memorijsko zauzeće pošto nema kopiranja podformula formule A. U paralelnom izvođenju je pogodno na svaki raspoloživi procesor poslati poseban deo formule koji odgovara jednoj ili više grana, pa se može uraditi prilikom primene β pravila kada se leva grana zadržava na aktuelnom procesoru, a desna šalje na neki slobodni procesor (*Ognjanović i Krdžavac, 2004.)*

Zaključak

U ovom radu, bavili smo se pregledom metoda u iskaznoj logici: metod rezolucije i metod tabloa, koje koristimo za dokazivanje zadovoljivosti nekih formula. Pravilo rezolucije, zapravo je uopštenje pravila modus ponens. Modus ponens se može predstaviti kao primena pravila rezolucije na klauze $\{\neg p \vee q\}$ i $\{p\}$, pri čemu se dobija klauza $\{q\}$. Ova metoda je po svojoj prirodi mehanička, orijentisana prema računarskom izvršavanju. To primećujemo kod primene na veće skupove formula kada se pojavljuje ogroman broj klauza. Ona se izučava, pored jednostavnog načina izvršavanja, i zbog toga što predstavlja osnov mehanizma izvođenja na kome je baziran programski jezik Prolog (*Ognjanović i Krdžavac, 2004.*). Rezolucija se može primeniti samo ako je moguće proizvoljne formule prevesti u oblik konjuktivne normalne forme, što nije slučaj u većini logika. Metod tabloa, za razliku od metoda rezolucije, može da se primenjuje i na iskazne formule koje nisu u konjuktivnoj normalnoj formi. On se definiše nad skupom veznika $(\vee, \wedge, \neg, \Rightarrow)$i predstavlja sistematičko traganje za nekom valuacijom koja zadovoljava datu formulu. Kod ovog metoda proces zapisujemo u obliku stabla, i rezultat čitamo na krajevima grana. Ako su ostale otvorene valuacija postoji, a ako su se sve grane zatvorile, došli smo do kontradikcije i valuacija ne postoji.

Oba metoda, mnogo jednostavnije se primenjuju na računaru, nego drugi, što znatno uvećava njihov značaj. Takođe, efikasnije je ispitati zadovoljivost (nezadovoljivost) formule pomoću njih, nego preko istinitosnih tablica.

Literatura

Janičić, P. (2008). *Matematička logika u računarstvu*. Beograd: Matematički fakultet

Madarász-Szilágyi, R. (2012). *Matematička logika*. Novi Sad: Prirodno-matematički fakultet.

Maksimović, P. (2008). *Jednočlani potpuni skupovi veznika za iskaznu logiku.* Beograd: Matematički fakultet.

Ognjanović, Z., & Krdžavac, N. (2004). *Uvod u teorijsko računarstvo*. Beograd - Kragujevac: Samostalno autorsko izdanje.

RESOLUTION METHOD AND METHOD OF TABLEAU IN PROPOSITIONAL LOGIC

Summary

In this paper we will show the importance of the resolution method and metod of tableau in the propositional logic. Resolution method is used when we want to examine whether a set of clauses is satisfiable or unsatisfiable, and the tableau method is used to disproof the statement formula, i.e. to determine its disatisfaction.

Keywords*: the resolution method, tableau method, propositional logic*

.