4.
Uporaba matematičke logike u računalima

1. Osnovne postavke logike

U našim dosadašnjim razmatranjima algoritama često smo koristili pojam nekog uvjeta i ispunjenja ili neispunjenja tog uvjeta. Ostvarenja programskih grananja i programskih petlji zasnivaju se na ispitivanju uvjeta i donošenja odluke koja ovisi o tome da li je neki uvjet ispunjen ili nije ispunjen. Općenito, ispitivanje različitih uvjeta, koji mogu biti i vrlo složeni, omogućuje konstruiranje vrlo korisnih programa.

Razumijevanje i stvaranje takvih programa znatno olakšava poznavanje logike[1].

Logika je grana filozofije koja se bavi utvrđivanjem razložnosti rasuđivanja. U običnom životu se kaže da je nešto logično, ako se to može obrazložiti nekim dokazima. Osnovni pojam u logici je *logički sud*. Pod tim nazivom razumijeva se svaka tvrdnja koja se ocjenjuje *samo s gledišta istinitosti ili lažnosti*. Neka druga svojstva sudova (primjerice: dobro, loše, lijepo, ružno) se u logici ne razmatraju. Logički sudovi moraju biti *valjani*. Valjan sud mora biti tako oblikovan da je ili *istinit* ili *lažan*. Sudovi koji mogu biti istovremeno i istiniti i lažni nisu valjani i u logici se ne razmatraju.

Primjer 4.1. Utvrditi istinitost ili lažnost sljedećih sudova: $5 > 2$, $3 > 6$, $4 = 4$, $7 \leqslant 6$.

Temeljem svojstava prirodnih brojeva zaključujemo:

sud "$5 > 2$" je istinit,

[1] dolazi od grčkog *logos*— riječ, razum, razlog, misao, zakon

sud "$3 > 6$" je lažan (neistinit),
sud "$4 = 4$" je istinit,
sud "$7 \leqslant 6$" je lažan (neistinit).

□

Logički sudovi mogu se tvoriti o različitim činjenicama i primjenjivati za rasuđivanje u vrlo raznolikim granama ljudske djelatnosti.

Pri zasnivanju algoritama i pri pisanju programa često se koriste sudovi oblikovani kao sudovi u primjeru 4.1. Pritom koristimo neke simbole koji se mogu nazvati *relacijskim simbolima ili operatorima.* Oni opisuju neke odnose između vrijednosti napisanih s njihove lijeve i desne strane. Relacijski simboli i njihova značenja prikazani su tablicom 4.1.

Tablica 4.1. *Relacijski operatori i njihovi simboli*

simbol	značenje
$<$	"je manje od" ili "prethodi" ili "dolazi ispred"
$>$	"je veće od" ili "slijedi" ili "dolazi iza"
$=$	"je jednako"
$\leqslant$ ili $<=$	"je manje ili jednako"
$\geqslant$ ili $>=$	"je veće ili jednako"
$\neq$ ili $<>$	"je različito" ("nije jednako")

Relacijski operatori mogu se primijeniti na skup prirodnih brojeva ili na bilo koji skup u kojem su elementi uređeni, tj. imaju utvrđeni poredak.

Znak za relacijski operator $=$ ne smije se zamijeniti sa znakom pridruživanja $:=$. Operator pridruživanja pridružuje varijabli koja je zapisana s njegove lijeve strane neku vrijednost, dok relacijski operator služi za uspoređivanje dviju vrijednosti.

Primjer 4.2. Za dane u tjednu možemo oblikovati sljedeće sudove i zaključiti da li su lažni ili istiniti:

sud "ponedjeljak $<$ srijeda" (ponedjeljak prethodi srijedi) je istinit
sud "petak $<$ utorak" (petak prethodi utorku) je lažan
sud "srijeda $>$ četvrtak" (srijeda slijedi četvrtak) je lažan.

□

Primjer 4.3. Za slova abecede možemo oblikovati sljedeće sudove i utvrditi njihovu istinitost ili lažnost:

sud "a $>$ b" (slovo a dolazi iza slova b) je lažan,
sud "l $>$ k" (slovo l dolazi iza slova k) je istinit,
sud "l $<$ m" (slovo l dolazi ispred slova m) je istinit.

Iz istinitosti zadnjih dvaju sudova možemo ustanoviti da je novi sud "slovo l dolazi između slova k i slova m" istinit. Drugim riječima, na temelju istinitosti ili lažnosti sudova možemo zaključiti istinitost ili lažnost nekih drugih sudova. □

Za istraživanje sudova i složenih sudova razvijena je posebna grana matematike — *matematička logika*. Osnove matematičke logike čini algebra sudova znana i pod nazivima *logička algebra* ili *Booleova*[2] *algebra*. U algebri sudova provode se *logičke* ili *Booleove operacije* nad sudovima.

Za označavanje sudova uvode se simboli. Pritom se govori da je taj simbol *logička varijabla* ili *Booleova varijabla*. Logička varijabla može poprimiti samo dvije vrijednosti i to: *lažnost* ili *istinitost*.

Te dvije vrijednosti označavaju se na različite načine:

- na temelju engleskih naziva[3] za istinito i lažno koriste se oznake: T i F;
- umjesto F neki koriste naopako slovo T (što bi trebalo značiti: obrnuto od istine), pa se koriste oznake: T i $\perp$;
- s obzirom da logička varijabla može poprimiti dvije vrijednosti (jednako kao i jedan bit), često se koriste i oznake: 0 i 1.

Mi ćemo koristiti zadnje spomenuti način označavanja vrijednosti logičkih varijabli, s tim da:

- istinitost označavamo s 1 i
- lažnost označavamo s 0.

Primjer 4.4. Za sudove iz primjera 4.3 uvedimo sljedeće simbole:

$$G = \text{"a} > \text{b"},$$
$$H = \text{"l} > \text{k"},$$
$$M = \text{"l} < \text{m"}.$$

Varijable G, H i M imaju ovdje sljedeće vrijednosti:

$$G = 0,$$
$$H = 1,$$
$$M = 1.$$

□

[2]George Boole (1815–1864), britanski matematičar i logičar, je u svojim djelima *Matematička analiza logike* (1847) i *Zakoni uma* (1854) prvi uveo matematičku simboliku za izražavanje logičkih procesa.

[3]engleski nazivi: *true* — istinito, *false* — lažno

Za *osnovne* ili *atomne*[4] *sudove* (kraće: *atome*) istinost ili lažnost mora se utvrditi neposrednim zaključivanjem o sudu, kao što je to i učinjeno u primjeru 4.4.

Istinitost ili lažnost složenijih sudova može se utvrditi na temelju *formula*. Formule se tvore od:

- atoma,
- logičkih operatora i
- zagrada.

Uobičajeno se koristi pet logičkih operatora koji određuju pet logičkih operacija. To su:

simbol operatora	naziv operacije
$\neg$	negacija
$\wedge$	konjunkcija
$\vee$	disjunkcija
$\rightarrow$	implikacija
$\leftrightarrow$	dvostrana implikacija

Za osnovna razmatranja o programima nama će biti dovoljne prve tri osnovne operacije. Zbog cjelovitosti izlaganja razmotrit će se i operacije implikacije i dvostrane implikacije, ali tek na kraju ovog poglavlja i to samo kao neobvezatno gradivo za one koji su za to posebno zainteresirani.

Negacija, konjunkcija i disjunkcija

Operator *negacije* djeluje na jedan atom, nazovimo ga G, i piše se: $(\neg G)$. Negacija se može izgovoriti na više načina, primjerice: "nije G", "negacija od G", "ne G". Djelovanje negacije prikazano je tablicom 4.2.

Tablica 4.2. *Tablica definicije negacije*

G	$(\neg G)$
0	1
1	0

Negacija vrijednosti 0 je vrijednost 1 i, obrnuto, negacija vrijednosti 1 je vrijednost 0.

[4] Atom i atomni dolazi od grčkog *atomos* — nedjeljiv, nerazreziv. Isto porijeklo ima i naziv za osnovnu česticu nekog kemijskog elementa, jer se prvobitno smatralo da je ona nedjeljiva. Danas znamo da je sud "atom je nedjeljiva čestica materije" lažan.

Novi izvedeni sud označimo simbolom M i možemo pisati:

$$M \equiv \neg G,$$

gdje simbol $\equiv$ označava da je lijeva strana *jednakovrijedna* ili *ekvivalentna*[5] desnoj strani, te je to *simbol ekvivalencije*.

U odjeljku 3.4, koji obrađuje načine zapisivanja cijelih brojeva, ustanovili smo da se međusobna zamjena brojki 0 i 1 zove komplementiranje. Stoga se po uzoru na to, negacija katkada naziva i *komplementiranjem*.

Primjer 4.5. Razmotriti sudove iz primjera 4.1 i opisati relacijskim operatorima njihove negacije.

Sudove iz zadatka 4.1 označit ćemo s G, a njihove negacije s M i zatim zaključiti što predstavlja sud M. Prikažimo to tablično:

sud G	vrijednost G	vrijednost $\neg G$	sud M
"$5 > 2$"	1	0	"$5 \leqslant 2$"
"$3 > 6$"	0	1	"$3 \leqslant 6$"
"$4 = 4$"	1	0	"$4 \neq 4$"
"$7 \leqslant 6$"	0	1	"$7 > 6$"

□

Iz primjera 4.5 se može zaključiti da se odgovarajućom zamjenom relacijskog operatora u nekom sudu može dobiti negacija tog suda. Nazovimo takav zamjenski operator suprotnim operatorom. Parovi međusobno suprotnih relacijskih operatora prikazani su tablicom 4.3.

Tablica 4.3. *Međusobno suprotni relacijski operatori*

operator	suprotni operator
$<$	$\geqslant$ ili $>=$
$>$	$\leqslant$ ili $<=$
$=$	$\neq$ ili $<>$
$\leqslant$ ili $<=$	$>$
$\geqslant$ ili $>=$	$<$
$\neq$ ili $<>$	$=$

[5]ekvivalencija dolazi od latinskog *aequus* — jednak i *valere* — vrijediti

Operator *konjunkcije*[6] daje formulu $(G \wedge H)$. Formula se izražava riječima: "konjunkcija od G i H", "G i H". Djelovanje konjunkcije prikazuje tablica 4.4.

Tablica 4.4. *Tablica definicije konjunkcije*

G	H	$(G \wedge H)$
0	0	0
0	1	0
1	0	0
1	1	1

Formula $(G \wedge H)$ je istinita samo ako su oba atoma istinita. Možemo reći da formula ima vrijednost 1 onda i samo onda kada i jedan i drugi atom imaju vrijednost 1. Konjunkcija se stoga zove još i *I operacijom* odnosno, u skladu s engleskim, *AND operacijom*.

Konjunkcija daje novi sud za koji možemo pisati

$$M \equiv (G \wedge H).$$

Primjer 4.6. Temeljem primjera 4.3 i 4.4 ustanovimo da su sudovi $G =$ "l > k" i $H =$ "l < m" istiniti. Stoga je i sud $M \equiv (G \wedge H)$ istinit. Kako bismo riječima opisali taj novi sud?

Sud glasi "l > k" i "l < m", tj. "slovo l nalazi se u abecedi iza slova k i ispred slova m" ili drugim riječima sud znači "slovo l nalazi se između slova k i m". □

Operatorom *disjunkcije*[7] dobiva se formula $(G \vee H)$, ili riječima: "disjunkcija od G i H", "G ili H". Djelovanje disjunkcije opisano je tablicom 4.5.

Tablica 4.5. *Tablica definicije disjunkcije*

G	H	$(G \vee H)$
0	0	0
0	1	1
1	0	1
1	1	1

[6]konjunkcija dolazi od latinskog *conjungere* — svezati, sjediniti, združiti

[7]disjunkcija dolazi od latinskog *disjungere* — razvezati, rastaviti, razdružiti

Formula $(G \vee H)$ je istinita ako je jedan od atoma istinit ili ako su oba atoma istinita. Disjunkcija se stoga zove još i *ILI operacijom*, odnosno, u skladu s engleskim, *OR operacijom*. Disjunkcija je, prema tome lažna samo ako su oba atoma lažna. Ona je istinita ako je barem jedan od atoma istinit.

Disjunkcija daje novi sud za koji možemo pisati

$$M \equiv (G \vee H).$$

2. Logičke funkcije i njihovo izražavanje s pomoću formula

U opisima operacija negacije, konjunkcije i disjunkcije ustanovili smo da novi sud, dakle, nova logička varijabla M ovisi o vrijednostima logičkih varijabli G i H. Može se reći da su logičke varijable G i H nezavisne varijable, a da je vrijednost varijable M neka funkcija — nazovimo je f — varijabli G i H. Uvrštavanje jedne od mogućih kombinacija vrijednosti varijabli u neku formulu nazivamo jednom interpretacijom[8] formule. U općem slučaju za dva atoma — dvije varijable — moglo bi se pisati:

$$M \equiv f(G, H),$$

Funkciju f možemo definirati tablično i to na sličan način kao što su definirane operacije negacije, konjunkcije i disjunkcije. Pritom za sve moguće vrijednosti ulaznih varijabli treba odrediti vrijednost funkcije. Takve tablice nazivaju se *tablicama istinitosti*.

S obzirom da nezavisne varijable mogu poprimiti samo dvije vrijednosti i da funkcija može poprimiti, također, samo dvije vrijednosti, tablice imaju točno utvrđenu veličinu i za dani broj varijabli postoji samo odgovarajući ograničeni broj različitih funkcija.

Funkcije jedne varijable

Jedna logička varijabla može poprimiti samo dvije vrijednosti pa za jednu nezavisnu logičku varijablu postoje samo dva retka u tablici istinitosti. Stupac u koji treba zapisati vrijednosti funkcije ima, dakle, dva mjesta.

Primjer 4.7. Tablicom 4.2 definirana je operacija negacije. Ta tablica je, ustvari, tablica istinitosti za funkciju

$$f(G) \equiv (\neg G).$$

[8] interpretacija dolazi od latinskog *interpretatio* — tumačenje, prijevod

Ponovimo tu tablicu još jedanput s tim da u oznaci ne pišemo $f(G)$ već samo f. U tablici 4.6, dakle, stupac označen s G određuje sve vrijednosti varijable G, a stupac f vrijednosti funkcije $f(G)$.

Tablica 4.6. *Tablica istinitosti funkcije* $f(G) \equiv (\neg G)$

G	f
0	1
1	0

□

Mi znamo da se dva mjesta u stupcu za zapisivanje vrijednosti funkcije mogu popuniti (kao i dvobitovni broj) na $2^2 = 4$ načina, pa postoje ukupno četiri moguće funkcije jedne varijable. Za svaku od njih bi mogli napisati tablicu istinitosti. Međutim, mi ćemo ih objediniti u jednu tablicu i to tako da su za nezavisnu varijablu G u tablici 4.7 prikazane sve te četiri funkcije.

Neke drugačije funkcije jedne varijable ne postoje, pa prema tome mogu postojati samo četiri funkcije jedne varijable.

Tablica 4.7. *Sve funkcije jedne varijable*

nezavisna varijabla	vrijednosti funkcija			
G	f_0	f_1	f_2	f_3
0	0	1	0	1
1	0	0	1	1

Četiri različite funkcije f_0, f_1, f_2 i f_3 imaju svaka svoj stupac. Indeksi su odabrani tako da budu jednaki brojevima koje dobijemo čitanjem stupca kao binarnog broja u kojem su bitovi zapisani redom odozdo prema gore. Mi, međutim, znamo da ovdje jedinice i nule znače istinitost odnosno lažnost.

Iz tablice vidimo da je:

- $f_0(G) \equiv 0$ — ta funkcija daje uvijek vrijednost 0 bez obzira na vrijednost od G;
- $f_3(G) \equiv 1$ — ta funkcija daje uvijek vrijednost 1 bez obzira na vrijednost od G;
- $f_1(G) \equiv (\neg G)$ — funkcija je negacija varijable G;
- $f_2(G) \equiv G$ — ta funkcija daje vrijednost jednaku vrijednosti G, pa se naziva funkcijom identiteta[9].

[9] identitet dolazi od latinskog *identitas* — istovjetnost, podudaranje, potpuna jednakost

Funkcije dviju varijabli

Tablica istinitosti za funkcije dviju nezavisnih varijabli ima četiri retka, jer te dvije varijable mogu poprimiti ukupno četiri različita para vrijednosti.

Primjer 4.8. Stupac u koji treba zapisati funkciju ima četiri mjesta i može se s dvije vrijednosti popuniti na $2^4 = 16$ načina (jednako kao i četverobitovni broj). Prema tome, možemo definirati ukupno 16 različitih logičkih funkcija dviju varijabli, koje su sve predstavljene tablicom 4.8.

Tablica 4.8. *Sve funkcije dviju varijabli*

nezavisne varijable		vrijednosti funkcija															
G	H	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Indeksi funkcija i ovdje su određeni tako da su jednaki broju koji se dobije ako se stupac funkcije čita odozdo prema gore kao binarni broj.

Promotrimo neke od funkcija:

- $f_0(G, H) \equiv 0$ — ta funkcija ima vrijednost 0 bez obzira na vrijednosti G i H;
- $f_8(G, H) \equiv (G \wedge H)$ — ta funkcija je konjunkcija varijabli G i H.
- $f_{14}(G, H) \equiv (G \vee H)$ — ta funkcija je disjunkcija varijabli G i H;
- $f_{15}(G, H) \equiv 1$ — ta funkcija ima vrijednost 1 bez obzira na vrijednosti G i H.

Treba uočiti da svaka funkcija ima svoj par u kojem su nule zamijenjene s jedinicama i, obrnuto, jedinice zamijenjene s nulama. Funkcije u tim parovima su, dakle, jedna drugoj negacije ili možemo reći da su *međusobno komplementarne*. Iz tablice 4.8 vidimo da su međusobno komplementarne one funkcije čiji je zbroj indeksa jednak 15.

Tako, vidimo da je:

- $f_0(G, H) \equiv \neg f_{15}(G, H)$;
- $f_8(G, H) \equiv \neg f_7(G, H)$ ili $f_7(G, H) \equiv \neg f_8(G, H)$;
- $f_{14}(G, H) \equiv \neg f_1(G, H)$ ili $f_1(G, H) \equiv \neg f_{14}(G, H)$.

S obzirom da funkcije $f_8(G, H)$ i $f_{14}(G, H)$ znamo već zapisati s pomoću formula, dobivamo:

- $f_7(G, H) \equiv (\neg(G \wedge H))$ i

- $f_1(G,H) \equiv (\neg(G \vee H))$.

Funkcija $f_7(G,H)$ ostvaruje se tako da se negira konjunkcija atoma G i H, tj. obavi najprije I operacija i zatim se rezultat negira. Kraće bi se to moglo reći "ne — I operacija" ili *NI operacija*, odnosno u skladu s engleskim, *NAND operacija*.

Funkcija $f_1(G,H)$ ostvaruje se tako da se negira disjunkcija atoma G i H, dakle obavlja se "ne — ILI operacija" ili kraće *NILI operacija*, odnosno *NOR operacija*. □

Primjer 4.9. Pogledajmo koje ćemo funkcije dobiti ako najprije varijable G i H negiramo i zatim načinimo njihovu konjunkciju odnosno disjunkciju, tj. odredimo vrijednost formula $((\neg G) \wedge (\neg H))$ odnosno $((\neg G) \vee (\neg H))$ za sve vrijednosti G i H. Načinit ćemo to s pomoću tablica istinitosti 4.9.

Tablica 4.9. *Tablice istinitosti za funkcije* $((\neg G) \wedge (\neg H))$ *i* $((\neg G) \vee (\neg H))$

G	H	$(\neg G)$	$(\neg H)$	$((\neg G) \wedge (\neg H))$	$((\neg G) \vee (\neg H))$
0	0	1	1	1	1
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	0

Vrijednosti funkcija određujemo redak po redak.

Najprije popunjavamo stupce $(\neg G)$ i $(\neg H)$ znajući da je: $\neg 0 \equiv 1$ i $\neg 1 \equiv 0$.

Nakon toga određujemo vrijednosti u stupcu $((\neg G) \wedge (\neg H))$ znajući da je: $1 \wedge 1 \equiv 1,\ 1 \wedge 0 \equiv 0,\ 0 \wedge 1 \equiv 0,\ 0 \wedge 0 \equiv 0$.

Isto tako određujemo vrijednosti u stupcu $((\neg G) \vee (\neg H))$ znajući da je: $1 \vee 1 \equiv 1,\ 1 \vee 0 \equiv 1,\ 0 \vee 1 \equiv 1,\ 0 \vee 0 \equiv 0$.

Usporedbom s tablicom 4.5, vidimo da je:

$$((\neg G) \wedge (\neg H)) \equiv f_1(G,H) \text{ i}$$
$$((\neg G) \vee (\neg H)) \equiv f_7(G,H).$$

Na takav način može se tablično odrediti vrijednost za bilo koju formulu, ili, drugim riječima, može se odrediti koju od mogućih šesnaest funkcija formula predstavlja. □

Ustanovili smo da simbol $\equiv$ označava da je lijeva strana jednakovrijedna ili ekvivalentna desnoj strani. Sada to možemo izreći na drugi način, tako da kažemo: *dvije su formule ekvivalentne ako predstavljaju istu funkciju.*

S obzirom da smo već ranije ustanovili da je $f_1(G,H) \equiv (\neg(G \vee H))$ i sada u primjeru 4.8 da je $f_1(G,H) \equiv ((\neg G) \wedge (\neg H))$, to je:

$$(\neg(G \vee H)) \equiv ((\neg G) \wedge (\neg H)).$$

Isto tako, kako je $f_7(G,H) \equiv (\neg(G \wedge H))$ i istovremeno $f_7(G,H) \equiv ((\neg G) \vee (\neg H))$, zaključujemo da vrijedi i ekvivalencija:

$$(\neg(G \wedge H)) \equiv ((\neg G) \vee (\neg H)).$$

U pisanju formula mogu se neke zagrade i izostaviti. U tom slučaju operatore se primjenjuje sa sljedećim redoslijedom prvenstva:

- negacija $\neg$,
- konjunkcija $\wedge$,
- disjunkcija $\vee$.

Zagrade, međutim, treba u formulama zadržati ako bi njihovim ispuštanjem nastale neke nejasnoće, ili ako želimo olakšati njihovo čitanje.

Gornje dvije ekvivalencije mogu se tako prepisati u jednostavniji oblik:

$$\neg(G \vee H) \equiv \neg G \wedge \neg H \text{ i}$$
$$\neg(G \wedge H) \equiv \neg G \vee \neg H.$$

Te su dvije ekvivalencije poznate kao de Morganovi[10] zakoni.

Postupak ostvarenja funkcija s pomoću negacije, konjunkcije i disjunkcije

Sve se logičke funkcije mogu ostvariti s pomoći negacije, konjunkcije i disjunkcije.

Da bismo se u to uvjerili pogledajmo najprije u tablici 4.8 one funkcije koje u svome stupcu imaju samo jednu jedinicu. To su funkcije:

- $K_0 \equiv f_1(G,H)$, koja ima vrijednost 1 samo u nultom retku;
- $K_1 \equiv f_2(G,H)$, koja ima vrijednost 1 samo u prvom retku;
- $K_2 \equiv f_4(G,H)$, koja ima vrijednost 1 samo u drugom retku i
- $K_3 \equiv f_8(G,H)$, koja ima vrijednost 1 samo u trećem retku.

Funkcija $f_8(G,H)$ je nama već poznata konjunkcija odnosno I operacija. Ona ima jedinicu u zadnjem retku u kojem obje varijable imaju vrijednost 1. Zaključujemo da bismo i preostale tri funkcije također mogli ostvariti konjunkcijama. Tako bi funkciju $f_1(G,H)$ mogli ostvariti tako da uzmemo negacije $\neg G$ i $\neg H$ i s njima načinimo konjukciju. Funkcije $f_2(G,H)$ i $f_4(G,H)$ ostvarit ćemo tako da prije primjene konjukcije negiramo samo varijablu G odnosno H. Te funkcije prozvat ćemo elementarnim konjunkcijama. Ostvarenje elementarnih konjunkcija prikazano je tablicom 4.10.

[10] Augustus de Morgan (1806–1871), britanski matematičar bio je suradnik Boolea u razradi simboličke logike

Tablica 4.10. *Ostvarenje elementarnih konjunkcija*

G	H	$\neg G$	$\neg H$	$K_0 \equiv \neg G \wedge \neg H$	$K_1 \equiv \neg G \wedge H$	$K_2 \equiv G \wedge \neg H$	$K_3 \equiv G \wedge H$
0	0	1	1	1	0	0	0
0	1	1	0	0	1	0	0
1	0	0	1	0	0	1	0
1	1	0	0	0	0	0	1

Nakon što smo ustanovili kako ćemo dobiti elementarne konjunkcije, podsjetimo se da operacija disjunkcije daje vrijednost 1 ako barem jedna od varijabli poprimi vrijednost 1. To znači da se *bilo koja funkcija* može dobiti tako da se:

- za danu funkciju napiše njezin stupac;
- za svaki redak u kojem se pojavljuje vrijednost 1 odabere pripadna elementarna konjunkcija;
- sve odabrane konjunkcije povežu disjunkcijom.

Primjer 4.10. Razmotrimo, primjerice, način ostvarenja funkcije $f_6(G, H)$. Ta funkcija ima vrijednosti 1 u prvom i drugom retku, a vrijednosti 0 u nultom i trećem retku. Stoga ćemo odabrati elementarne konjunkcije K_1 i K_2 i povezati ih disjunkcijom. Tablica 4.11 pokazuje da smo na taj način ostvarili funkciju $f_6(G, H)$.

Tablica 4.11. *Ostvarenje funkcije* $f_7(G, H)$

G	H	$\neg G$	$\neg H$	$f_6(G, H)$	$K_1 \equiv \neg G \wedge H$	$K_2 \equiv G \wedge \neg H$	$K_1 \vee K_2$
0	0	1	1	0	0	0	0
0	1	1	0	1	1	0	1
1	0	0	1	1	0	1	1
1	1	0	0	0	0	0	0

Postupamo na jednak način kao u primjeru 4.9.

Najprije redak po redak određujemo vrijednosti u stupcima $\neg G$ i $\neg H$ znajući da je: $\neg 0 \equiv 1$ i $\neg 1 \equiv 0$.

Nakon toga određujemo vrijednosti u stupcima K_1 i K_2 znajući da je: $1 \wedge 1 \equiv 1$, $1 \wedge 0 \equiv 0$, $0 \wedge 1 \equiv 0$, $0 \wedge 0 \equiv 0$.

Na kraju određujemo vrijednosti u stupcu $K_1 \vee K_2$ znajući da je: $1 \vee 0 \equiv 1$, $0 \vee 1 \equiv 1$, $0 \vee 0 \equiv 0$. Usporedbom tog stupca sa stupcem $f_6(G, H)$ ustanovljujemo jednakovrijednost, te je:

$$f_6(G, H) \equiv K_1 \vee K_2 \equiv (\neg G \wedge H) \vee (G \wedge \neg H).$$

□

Treba zapaziti da je funkcija $f_6(G,H)$ vrlo slična ILI operaciji. Ona se od nje razlikuje samo po tome što ima vrijednost 0 kada oba atoma imaju vrijednost 1. Drugim riječima ova funkcija daje vrijednost 1 ako je ili jedan ili drugi atom istinit, ali daje vrijednost 0 ako su oba atoma istinita. Zbog toga se kaže da formula $(\neg G \wedge H) \vee (G \wedge \neg H)$ definira *ISKLJUČIVO-ILI operaciju*, koju u skladu s engleskim zovemo *XOR operacijom*[11]. Za tu operaciju koristi se često i posebni operator — $\oplus$, tako da se može pisati:

$$G \oplus H \equiv (\neg G \wedge H) \vee (G \wedge \neg H).$$

Pogledajmo pažljivo tablicu 4.12 koja prikazuje rezultate funkcije $f_6(G,H)$ odnosno ISKLJUČIVO-ILI operaciju i usporedimo je s tablicom 4.5. Vidimo da razlika postoji samo u zadnjem retku.

Tablica 4.12. *Definicija operacije ISKLJUČIVO-ILI*

G	H	$(G \oplus H)$
0	0	0
0	1	1
1	0	1
1	1	0

Već smo ustanovili da se ista logička funkcija može izraziti različitim ekvivalentim formulama. Neke od formula mogu biti složenije (sadržavati više simbola atoma i operatora) a neke jednostavnije. Mi za neku danu funkciju možemo odabrati bilo koju ekvivalentnu formulu. Najčešće odabiremo najjednostavniju.

Primjer 4.11. Predpostavimo da želimo ostvariti funkciju $f_{14}(G,H)$ iz tablice 4.8. Vidimo da moramo odabrati elementarne konjunkcije K_1, K_2 i K_3 i povezati ih disjunkcijama. S obzirom da smo naučili da disjunkcija vrijedi za par varijabli načinimo to na sljedeći način: $((K_1 \vee K_2) \vee K_3)$. Zagrade možemo izostaviti pa dobivamo: $K_1 \vee K_2 \vee K_3$. Koristeći tablicu 4.10 možemo, dakle, pisati:

$$f_{14}(G,H) \equiv K_1 \vee K_2 \vee K_3 \equiv (\neg G \wedge H) \vee (G \wedge \neg H) \vee (G \wedge H).$$

S druge strane, znamo da je funkcija $f_{14}(G,H)$ odredena disjunkcijom, tj. da je

$$f_{14}(G,H) \equiv G \vee H,$$

pa je:

$$(\neg G \wedge H) \vee (G \wedge \neg H) \vee (G \wedge H) \equiv G \vee H.$$

[11] od engleskog *exclusive or* — isključivo ili

Vidimo da je razlika u složenosti formula velika i da je vrijedno pronalaziti jednostavnije ekvivalentne formule za izražavanje funkcija. □

Primjer 4.12. Izrazimo funkciju $f_{15}(G, H)$ iz tablice 4.8 kao disjunkciju sviju četiriju elementarnih konjunkcija, što daje:

$$f_{15}(G, H) \equiv K_0 \vee K_1 \vee K_2 \vee K_3 \equiv (\neg G \wedge \neg H) \vee (\neg G \wedge H) \vee (G \wedge \neg H) \vee (G \wedge H).$$

S druge strane znamo da je

$$f_{15}(G, H) \equiv 1,$$

pa je:

$$(\neg G \wedge \neg H) \vee (\neg G \wedge H) \vee (G \wedge \neg H) \vee (G \wedge H) \equiv 1.$$

Ova složena formula daje vrijednost 1 za sve moguće kombinacije svojih varijabli!

Ovakve formule, kod kojih su sve interpretacije istinite, u logici se zovu *tautologije*[12]. Formula koja je tautologija uvijek je istinita bez obzira na vrijednosti svojih varijabli. □

Primjer 4.13. Pogledajmo sve interpretacije formule $(\neg G \vee H) \wedge (G \wedge \neg H)$. Najjednostavnije je to načiniti tablično. Tablica 4.13 pokazuje da je formula za sve interpretacije neistinita.

Tablica 4.13. *Interpretacije formule* $(\neg G \vee H) \wedge (G \wedge \neg H)$

G	H	$\neg G$	$\neg H$	$\neg G \vee H$	$G \wedge \neg H$	$(\neg G \vee H) \wedge (G \wedge \neg H)$
0	0	1	1	1	0	0
0	1	1	0	1	0	0
1	0	0	1	0	1	0
1	1	0	0	1	0	0

Prema tome možemo pisati da je

$$(\neg G \vee H) \wedge (G \wedge \neg H) \equiv 0.$$

Ova formula daje vrijednost 0 za sve kombinacije vrijednosti svojih varijabli, tj. ona je uvijek lažna.

Takve formule nazivaju se u logici *kontradikcijama*[13]. □

Neke korisne ekvivalencije

Pri pojednostavljenju formula koristimo se odgovarajućim ekvivalencijama. Neke ekvivalencije već smo objasnili u prethodnim točkama, a ostale se može provjeriti tablično.

[12] tautologija dolazi od grčkog *tauto* — isto i *logos* — riječ, govor

[13] kontradikcija dolazi od latinskog *contra* — protiv i *dictio* — kazivanje, govor

- Ekvivalencije s jednom varijablom i konstantama:

$\neg(\neg G)$	$\equiv G$	idempotentnost[14] negacije
$G \wedge G$	$\equiv G$	
$G \wedge 1$	$\equiv G$	
$G \wedge 0$	$\equiv 0$	
$G \wedge \neg G$	$\equiv 0$	
$G \vee G$	$\equiv G$	
$G \vee 1$	$\equiv 1$	
$G \vee 0$	$\equiv G$	
$G \vee \neg G$	$\equiv 1$	

- Ekvivalencije s dvije varijable

$G \wedge H$	$\equiv H \wedge G$	komutativnost[15] konjunkcije
$G \vee H$	$\equiv H \vee G$	komutativnost disjunkcije
$\neg(G \wedge H)$	$\equiv \neg G \vee \neg H$	de Morganov zakon
$\neg(G \vee H)$	$\equiv \neg G \wedge \neg H$	de Morganov zakon
$G \vee (\neg G \wedge H)$	$\equiv G \vee H$	apsorpcija[16]
$G \wedge (\neg G \vee H)$	$\equiv G \wedge H$	apsorpcija

- Ekvivalencije s tri varijable

$(G \wedge H) \wedge K$	$\equiv G \wedge (H \wedge K)$	asocijativnost[17] konjunkcije
$(G \vee H) \vee K$	$\equiv (G \vee H) \vee K$	asocijativnost disjunkcije
$G \wedge (H \vee K)$	$\equiv (G \wedge H) \vee (G \wedge K)$	distributivnost[18] konjunkcije
$(G \vee (H \wedge K)$	$\equiv (G \vee H) \wedge (G \vee K)$	distributivnost disjunkcije

Primjer 4.14. Provjeriti apsorpciju $G \vee (\neg G \wedge H) \equiv G \vee H$ tablično.

Tablica 4.14. *Provjera ekvivalencije* $G \vee (\neg G \wedge H) \equiv G \vee H$

G	H	$\neg G$	$\neg G \wedge H$	$G \vee (\neg G \wedge H)$	$G \vee H$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Najprije određujemo vrijednosti u stupcu $\neg G$ tako da invertiramo vrijednosti stupca G.

[14] idempotentnost dolazi od latinskog *idem* — isti, isto i *potens* — podoban, valjan
[15] komutativnost dolazi od latinskog *commutatio* — promjena
[16] apsorpcija dolazi od latinskog *absorbere* — srkati, proždirati
[17] asocijativnost dolazi od latinskog *associare* — udružiti, ujediniti
[18] distributivnost dolazi od latinskog *distribuere* — razdijeliti, podijeliti

Nakon toga dobivamo stupac $\neg G \wedge H$ odredujući redom vrijednosti: $1 \wedge 0 \equiv 0$, $1 \wedge 1 \equiv 1$, $0 \wedge 0 \equiv 0$ i $0 \wedge 1 \equiv 0$.

Iz stupaca G i $\neg G \wedge H$ dobivamo vrijednosti stupca $G \vee (\neg G \wedge H)$ računajući redom: $0 \vee 0 \equiv 0$, $0 \vee 1 \equiv 1$, $1 \vee 0 \equiv 1$ i $1 \vee 0 \equiv 1$.

Konačno, zadnji stupac je već poznati rezultat konjunkcije. Vidimo da su vrijednosti zadnjeg i predzadnjeg stupca međusobno jednake u svim redcima. Na temelju toga zaključujemo da su formule $G \vee (\neg G \wedge H)$ i $G \vee H$ ekvivalentne. □

3. Uporaba matematičke logike za pojednostavljenje programa

U poglavlju o algoritmima naučili smo da se mnogi algoritmi zasnivaju na odabiru alternativnih[19] programskih odsječaka ovisno o ispunjenju ili neispunjenju nekog uvjeta ili na ponavljanju programskih odsječaka do ispunjenja ili neispunjenja nekog uvjeta.

Sada znamo da su ti uvjeti ustvari logički sudovi koji mogu poprimiti jednu od dvije vrijednosti: istinitost ili lažnost. Uvjeti moraju biti tako oblikovani da imaju svojstva logičkih sudova, tj. da ne mogu istovremeno biti ispunjeni i neispunjeni.

Instrukciju odlučivanja možemo, dakle, napisati ovako:

```
ako je logički sud istinit
| onda
|   | grana 1
| inače
|   | grana 2;
```

Podsjetimo se da se grana 1 obavlja kada je logički sud istinit, a grana 2 kada je sud lažan. Uvjet se kraće zapisuje bez riječi "istinit". Umjesto logičkog suda možemo zapisati i simbol koji predstavlja logički sud, primjerice, simbol `G`. Tada ista instrukcija izgleda ovako:

```
ako je G
| onda
|   | grana 1
| inače
|   | grana 2;
```

[19] alternativni dolazi od latinskog *alter* — jedan od dvojice

Isti učinak postižemo ako umjesto G ispitujemo istinitost negacije od G, tj. ¬G, i pritom zamijenimo grane. Tako dobivamo:

```
ako je ¬G
|  onda
|  |  grana 2
|  inače
|  |  grana 1;
```

Naime, ako je G istinit onda je ¬G lažan i bit će obavljena grana koja je napisana iza riječi inače, a to je opet grana 1. Ako je G neistinit onda je ¬G istinit i bit će obavljena grana napisana iza riječi onda, a to je grana 2.

Primjer 4.15. Napisati funkciju za određivanje apsolutne vrijednosti na dva načina.

U poglavlju 1 napisana je funkcija za računanje apsolutne vrijednosti na sljedeći način:

```
abs (X)
|  ako je  X < 0
|  |  onda  Y := −X
|  |  inače  Y := X ;
|  vratiti (Y) ;
kraj .
```

Ona se može preurediti na sljedeći način:

```
abs (X)
|  ako je  ¬(X < 0)
|  |  onda  Y := X
|  |  inače  Y := −X ;
|  vratiti (Y) ;
kraj .
```

U tablici 4.3 navedeni su suprotni relacijski operatori. Tamo vidimo da vrijedi ekvivalencija:

$$\neg(X < 0) \equiv (X \geqslant 0),$$

pa se na kraju funkcija može prepisati u oblik:

```
abs (X)
  | ako je  X ⩾ 0
  |   | onda  Y := X
  |   | inače  Y := −X ;
  | vratiti (Y) ;
kraj .
```

□

U jezičnoj konstrukciji `ako je` ispituju se pojedinačni sudovi. Često se u programima pojavljuje potreba za ispitivanjem složenijih sudova. Struktura takvih programa se ponekad može znatno pojednostavniti ako se koriste saznanja matematičke logike. Pogledajmo to na nekoliko primjera.

Primjer 4.16. Pojednostaviti sudove: $(\neg(a \geqslant b))$, $(\neg(a \leqslant b))$, $(\neg(a \neq b))$, $(\neg((a > b) \vee (a < b)))$.

Koristeći se tablicom 4.3 dobivamo:

$$\begin{aligned}
&(\neg(a \geqslant b)) && \equiv (a < b)\\
&(\neg(a \leqslant b)) && \equiv (a > b)\\
&(\neg(a \neq b)) && \equiv (a = b)\\
&(\neg((a > b) \vee (a < b))) && \equiv (a = b)
\end{aligned}$$

□

Primjer 4.17. Pretpostavimo da varijabla `X` predstavlja jedan bajt i da će nakon instrukcije `ulaz (X)` poprimiti kodnu vrijednost jednog znaka. Napisati program u kojem će se ustanoviti da li je to kodna vrijednost jednog suglasnika prema tablici 3.4.

Možemo program zamisliti tako da dobavljenu ulaznu vrijednost za `X` usporeduje redom s kodnim vrijednostima za suglasnike, odnosno da ustanovi da li ulazna kodna vrijednost `X`, izražena dekadskim ekvivalentom, zadovoljava sljedeće sudove:

za velika slova:	za mala slova:
$X = 64$	$X = 96$
$66 \leqslant X \leqslant 68$	$98 \leqslant X \leqslant 100$
$70 \leqslant X \leqslant 72$	$102 \leqslant X \leqslant 104$
$74 \leqslant X \leqslant 78$	$106 \leqslant X \leqslant 110$
$80 \leqslant X \leqslant 84$	$112 \leqslant X \leqslant 116$
$86 \leqslant X \leqslant 94$	$118 \leqslant X \leqslant 126$

Program bi se bez daljnjeg promišljanja sveo na razmjerno veliki broj ispitivanja. Za svaki od gornjih intervala dobivamo dva suda (primjerice za interval $64 \leqslant$ `X` $\leqslant 68$ to su sudovi: `X` $\geqslant 64$ i `X` $\leqslant 68$) tako da bi trebalo ispitati ukupno 22 suda.

Program se može zasnovati na drugi način tako da se najprije ispita da li je znak samoglasnik, te ako on to nije tada se zaključuje da je suglasnik. Iz tablice 3.4 vidimo da samoglasnici imaju sljedeće dekadski izražene kodne vrijednosti:

slovo	kôd	slovo	kôd
A	65	a	97
E	69	e	101
I	73	i	105
O	79	o	111
U	85	u	117

Međutim, najprije bi trebalo ispitati da li je unešena vrijednost varijable uopće kodna vrijednost za slovo. Opet iz tablice 3.4 vidimo da sva slova imaju kodne vrijednosti u intervalima

$$64 \leqslant X \leqslant 94 \text{ i } 96 \leqslant X \leqslant 126.$$

Vidimo da bi se ta dva intervala mogla objediniti u jedan interval $64 \leqslant X \leqslant 126$ ako se izuzme vrijednost 95, tj. vrijedi ekvivalencija

$$((X \geqslant 64) \wedge (X \leqslant 94)) \vee ((X \geqslant 64) \wedge (X \leqslant 126)) \equiv (X \geqslant 96) \wedge (X \leqslant 126) \wedge (X \neq 95).$$

Nadalje, nas ne zanima je li slovo veliko ili malo. Drugim riječima, mi možemo broj ispitivanja raspoloviti ako "pretvorimo" sva slova u velika ili u mala. Razlika u kodnoj vrijednosti istog slova je 32. Odlučimo li se, recimo, sva slova pretvoriti u velika, onda moramo od kodne vrijednosti oduzeti 32 uvijek kada je ona veća ili jednaka 96.

Plan je programa, prema tome, sljedeći:

- najprije ispitati da li je vrijednost `X` u intervalima kodnih vrijednosti za slova;
- preračunati vrijednost `X` tako da predstavlja veliko slovo;
- ispitati da li je vrijednost `X` jedna od pet vrijednosti za samoglasnike.

Program se može zapisati na sljedeći način:

```
početak
    ulaz(X);
    ako je ((X ⩾ 64) ∧ (X ⩽ 126) ∧ (X ≠ 95))
        onda
            ako je (X ⩾ 96)
                onda X := X − 32;
            ako je ((X=65)∨(X=69)∨(X=73)∨(X=79)∨(X=85))
                onda izlaz ("X nije suglasnik")
                inače izlaz ("X je suglasnik");
        inače izlaz ("X nije kodna vrijednost
                      slova abecede");
kraj.
```

Program se može prepisati i na drugi način tako da se negira sud čija istinitost određuje da je `X` kodna vrijednost slova, tj. trebalo bi pisati: $\neg((\mathtt{X} \geqslant 64) \wedge (\mathtt{X} \leqslant 126) \wedge (\mathtt{X} \neq 95))$. Koristeći de Morganov zakon možemo napisati ekvivalenciju

$$\neg((X \geqslant 64) \wedge (X \leqslant 126) \wedge (X195)) \equiv \neg(X \geqslant 64) \vee \neg(X \leqslant 126) \vee \neg(X \neq 95).$$

Nadalje, na temelju tablice suprotnih relacijskih operatora možemo konačno pisati:

$$\neg((X \geqslant 64) \wedge (X \leqslant 126) \wedge (X \neq 95)) \equiv (X < 64) \vee (X > 126) \vee (X = 95).$$

Do ovog suda možemo doći i neposredno promatrajući kodne vrijednosti u tablici 3.4.

Prema tome, drugi oblik programa izgleda ovako:

```
početak
    ulaz (X);
    ako je ((X < 64) ∨ (X > 126) ∨ (X = 95))
        onda izlaz ("X nije kodna vrijednost
        slova abecede");
        inače
            ako je (X ⩾ 96)
                onda X := X − 32;
            ako je ((X=65)∨(X=69)∨(X=73)∨(X=79)∨(X=85))
                onda izlaz ("X nije suglasnik")
                inače izlaz ("X je suglasnik");
kraj.
```

□

4. Osnovna pravila logičkog zaključivanja

U ovom završnom odjeljku ovog poglavlja samo se zbog cjelovitosti razmatranja iznose osnovna svojstva operacija *implikacije* i *dvostrane implikacije*. Ovo gradivo se na razini prvog izučavanja algoritama i programa može potpuno preskočiti. Ipak, za izgradnju naprednijih programa, u kojima se na temelju istinitosti ili neistinitosti nekih činjenica donose zaključci, te dvije operacije imaju veliku važnost. U matematici se ove operacije logičkog zaključivanja koriste za dokazivanje istinitosti teorema.

Implikacija

Operatorom implikacije[20] dobiva se formula $(G \to H)$ definirana sljedećom tablicom:

[20] implikacija dolazi od latinskog *implicare* — zaplesti, pripojiti

Tablica 4.15. *Tablica definicije implikacije*

G	H	$(G \to H)$
0	0	1
0	1	1
1	0	0
1	1	1

Dakle, vidimo iz tablice da vrijedi: $0 \to 0 \equiv 1$, $0 \to 1 \equiv 0$, $1 \to 0 \equiv 1$ i $1 \to 1 \equiv 1$.

Implikacija $(G \to H)$ može djelovati malo zbunjujuće kada se opisuje riječima. Međutim, ona je vrlo važna formula pri zaključivanju. Zato, najprije pažljivo promotrimo zadnja dva retka tablice i zanemarimo postojanje prva dva retka:

- varijabla G ima u oba zadnja retka vrijednost istinitosti;
- u predzadnjem retku implikacija $(G \to H)$ je lažna i vrijednost varijable H je lažna;
- u zadnjem retku implikacija $(G \to H)$ je istinita i varijabla H je istinita.

Ove zaključke možemo izreći na različite načine (zapamtimo da gledamo samo zadnja dva retka tablice):

a) "Uz istinitost implikacije $(G \to H)$, istinitost od G dovoljna je za istinitost H" ili kraće: "*G je dovoljno za H*";

b) "Uz istinitost implikacije $(G \to H)$, istinitost od G dovoljni je uvjet za istinitost H", ili kraće: "*G je dovoljni uvjet za H*";

c) "Uz istinitost implikacije $(G \to H)$, istinitost od H je nužna za istinitost G", ili kraće: "*H je nužno za G*";

d) "Uz istinitost implikacije $(G \to H)$, vrijedi: ako je istinito G onda je istinito H", ili kraće: "Ako G onda H"

e) "Uz istinitost implikacije $(G \to H)$, G je istinito samo onda kada je istinito H", ili kraće: "G je samo onda kada je H"

Primjer 4.18. Pretpostavimo da G i H predstavljaju sljedeće sudove: $G =$ "$x = 2$" i $H =$ "$3x = 6$". Za koje je vrijednosti od $x \in \mathbf{N}$ zadovoljena implikacija $(G \to H)$? Za $x = 2$ dobivamo $G = 1$ i $H = 1$, te je u skladu s tablicom definicije implikacije $(G \to H) = 1$. U skladu s načinom čitanja d) dakle vrijedi: ako je "$x = 2$", onda "$3x = 6$". Međutim, pogledajmo što dobivamo za $x \neq 2$. Sud G je lažan, tj. $G = 0$, a i za sud H vrijedi $H = 0$. Prema tablici definicije implikacije dobivamo $(G \to H) = 1$. Drugim riječima, implikacija je istinita za $x = 2$ i za $x \neq 2$, tj. za sve x, pa nam naše zaključivanje — ako je "$x = 2$"onda "$2x = 6$"— prividno ne bi vrijedilo. Stoga uvijek moramo imati na umu da zaključivanje na temelju implikacije vrijedi samo ako je sud G istinit. Kada sud G nije istinit ne može se izreći nikakva tvrdnja o sudu H. □

Sada promotrimo *prvi i treći* redak tablice i zanemarimo ostala dva retka:

- varijabla H u oba ta retka ima vrijednost lažnosti;
- u prvom je retku implikacija $(G \to H)$ istinita i vrijednost varijable G je lažnost;
- u trećem je retku implikacija $(G \to H)$ lažna i vrijednost varijable G je istinitost.

Ove zaključke možemo također izreći na različite načine, no zadovoljimo se ovaj put samo s jednim i to :

f) "Uz istinitost implikacije $(G \to H)$ vrijedi: ako je neistinito H, onda je neistinito G", ili kraće: "ako nije H onda nije G".

Zbunjujuće može djelovati drugi redak tablice, ali on se pri donošenju zaključaka ni ne koristi.

Pravila zaključivanja

Implikacija se, dakle, može koristiti za donošenje zaključaka dvojako i to:

- na temelju izričaja d) dobiva se način koji se zove *modus ponens*[21]:
  "ako G i $(G \to H)$, onda H";
- na temelju izričaja f) dobiva se način koji se zove *modus tollens*[22]
  "ako $\neg H$ i $(G \to H)$ onda $\neg G$".

Primjer 4.19. Kako djeluje modus ponens može se prikazati tablično:

G	H	$G \to H$	$(G \wedge (G \to H))$	$((G \wedge (G \to H)) \to H)$
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	1	1	1	1

Rezultate predzadnjeg stupca dobivamo tako da primijenimo konjunkciju na vrijednosti stupca G i $(G \to H)$ i to redom:

$$0 \wedge 1 \equiv 0,\ 0 \wedge 0 \equiv 0,\ 1 \wedge 1 \equiv 1 \text{ i } 1 \wedge 1 \equiv 1.$$

Zadnji stupac dobijemo tako da određujemo (redom po redcima) implikacije za vrijednosti iz stupaca $(G \wedge (G \to H))$ i H pa dobivamo:

$$0 \to 0 \equiv 1,\ 0 \to 1 \equiv 1,\ 0 \to 0 \equiv 1 \text{ i } 1 \to 1 \equiv 1.$$

Formula $((G \wedge (G \to H)) \to H)$ je dakle tautologija — ona vrijedi za sve svoje interpretacije, tj. za bilo koje vrijednosti za G i H. Kažemo još da je ona logički istinita. Formula na algebarski način govori da modus ponens uvijek vrijedi. □

[21] dolazi od latinsko *modus* — način i *pono* — postaviti, tvrditi

[22] dolazi od latinskog *modus* — način i *tollo* — odstraniti, ukloniti