PRESEK
List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652
Letnik **12** (1984/1985)
Številka 1
Strani 29–31

Izidor Hafner:

METODA SEMANTIČNIH TABEL ZA REŠEVANJE LOGIČNIH NALOG

Ključne besede: matematika.

Elektronska verzija: http://www.presek.si/12/694-Hafner.pdf

METODA SEMANTIČNIH TABEL ZA REŠEVANJE LOGIČNIH NALOG

V tem sestavku se bomo seznanili s preprosto metodo reševanja logičnih nalog, kakršna je na primer naslednja uganka:

Andrej, Boris in Cene vsak dan kosijo v restavraciji, naročijo pa pečenko ali šunko.

1. Če je Andrej šunko, potem je Boris pečenko.
2. Šunko naroči Andrej ali Cene, vendar pa ne oba.
3. Oba, Boris in Cene, ne jesta hkrati pečenke.

Kdo je lahko včeraj jedel šunko, danes pa pečenko?

Preden se lotimo te uganke (bralec jo lahko za vajo reši sam), bomo uvedli osnovne izjavne povezave. Definirali jih bomo tako, da bomo povedali, kako je resničnost sestavljene izjave odvisna od resničnosti njenih delov:

ime povezave	simbolični zapis	resnica	neresnica
negacija	$\neg A$	če je A neresnična	če je A resnična
konjunkcija	$A \wedge B$	če sta A in B resnični	če je vsaj ena izmed izjav A in B neresnična
disjunkcija	$A \vee B$	če je vsaj ena izmed izjav A in B resnična	če sta obe izjavi A in B neresnični
implikacija	$A \Rightarrow B$	če je A neresnična ali pa B resnična	če je A resnična in B neresnična
ekvivalenca	$A \Leftrightarrow B$	če sta A in B obe resnični ali obe neresnični	če je A resnična in B neresnična ali: če je A neresnična in B resnična

Prvi del reševanja naloge je sestavljen iz prevajanja v logično simboliko, to je iz logične stavčne analize.

Označimo posamezne enostavne izjave takole:

z "A" izjavo "Andrej jé (danes) šunko."
z "B" izjavo "Boris jé (danes) šunko."
s "C" izjavo "Cene jé (danes) šunko."

Potem lahko izjavo

"Andrej jé (danes) pečenko." zaznamujemo "$\neg A$", ker jé Andrej šunko ali pečenko, vendar ne obojega isti dan. Podobno velja tudi za Borisa in Ceneta. Lahko bi seveda uvedli tri dodatne črke za izjave "... jé pečenko.", vendar se bomo držali naslednjega napotka: Uporabljaj čim manj črk.

Pogoj 1 prevedemo z

(α) $A \Rightarrow \neg B$

Ta pogoj namreč izključuje možnost, da bi Andrej jedel šunko (da je izjava A resnična) in da bi hkrati tudi Boris jedel šunko (in da je hkrati izjava B resnična).

Pogoj 2 pravi, da jé Andrej šunko natanko tedaj, kadar je Cene ne naroči, torej

(β) $A \Leftrightarrow \neg C$

Pogoj 3 pravi, da ni res, da oba (Cene in Boris) hkrati jesta pečenko. To pomeni, da je v danem dnevu vsaj eden ne naroči, torej jé vsaj eden šunko:

(γ) $B \vee C$

Seveda bi pogoj 3 lahko prevedli tudi kot

(γ') $\neg(\neg B \wedge \neg C)$

vendar bi s tem kršili naslednji napotek:

Prevodi naj bodo čim bolj enostavni.

Mimogrede opazimo, da velja De Morganov zakon

$$B \vee C \Leftrightarrow \neg(\neg B \wedge \neg C)$$

Sedaj smo pred naslednjo nalogo:

pri kakšnih vrednostih enostavnih izjav A, B in C so izpolnjeni (resnični) pogoji (α), (β) in (γ)?

Da bi bil izpolnjen pogoj (γ), mora biti resnična vsaj ena izmed izjav B oziroma C. Da bi bil izpolnjen pogoj (β), imamo dve možnosti: (1) da sta A in $\neg C$ obe resnični ali (2) da sta A in $\neg C$ obe neresnični in zato izjavi $\neg A$ in C resnični. Da bi bil pogoj (α) izpolnjen, sta dve možnosti: da A

ni resnična (da je $\neg A$ resnična) ali da je $\neg B$ resnična.

Zapišimo vse kombinacije resničnih izjav:

(1) $A \Rightarrow \neg B$
(2) $A \Leftrightarrow \neg C$ } dani pogoji
(3) $B \vee C$

(4) B | (5) C — pogoj (3)

(6) A | (8) $\neg A$ | (10) A | (12) $\neg A$ } pogoj (2)
(7) $\neg C$ | (9) C | (11) $\neg C$ | (13) C
(14) $\neg A$ | (15) $\neg B$ | (16) $\neg A$ | (17) $\neg B$ | X | (18) $\neg A$ | (19) $\neg B$ — pogoj (1)
X X X

Veja (1), (2), (3), (5), (10), (11) vsebuje protislovje, namreč zahtevo po resničnosti obeh izjav C in $\neg$ C, zato smo jo označili z X (mrtva veja) in je nismo nadaljevali.

Ko smo upoštevali še tretji pogoj, smo dobili še nadaljnje mrtve veje. Preostale pa vsebujejo naslednje možnosti:

$B, \neg A, C$	veja (1), (2), (3), (4), (8), (9), (16)
$C, \neg A$	veja (1), (2), (3), (5), (12), (13), (18)
$C, \neg A, \neg B$	veja (1), (2), (3), (5), (12), (13), (19)

Vse pa vključujejo zahtevo po resničnosti izjav

C in $\neg A$

medtem ko sta za B obe možnosti odprti.

Torej:
Cene jé (vedno) šunko.
Andrej jé (vedno) pečenko.
Boris jé lahko eno ali drugo.

Analiza pogojev, kot smo jo naredili, se imenuje semantična tabela. Iz nje lahko razberemo vse možnosti, ki izpolnjujejo dane pogoje. Namesto "semantična tabela" bi lahko rekli tudi "semantično drevo."

Drugačen vrstni red upoštevanja pogojev nam bo dal drugačno semantično tabelo:

(1) $A \Rightarrow \neg B$
(2) $A \Leftrightarrow \neg C$
(3) $B \vee C$

(4) A | (6) $\neg A$ } pogoj (2)
(5) $\neg C$ | (7) C

(8) $\neg A$ | (9) $\neg B$ | (10) $\neg A$ | (11) $\neg B$ — pogoj (1)
X | (12) B | (13) C | (14) B | (15) C | (16) B | (17) C — pogoj (3)
X X X

Neprotislovne veje vselej vsebujejo $\neg A$ in C.

Algoritem semantičnih tabel ima za vhod neko množico izjav (za katere želimo, da so resnične), $A_1, A_2, \dots, A_n$.

Te izjave zapišemo eno za drugo v tabelo. Nato tabelo razširimo tako, da naredimo naslednje:

A

(a) Označimo za mrtve tiste veje, ki vsebujejo protislovne zahteve, to je, v njih se pojavlja neka izjava in njena negacija.

(b) Odkljukamo tiste točke, v katerih nastopajo le enostavne izjave ali njihove negacije.

(c) Končamo, če so vse veje mrtve.

B

(a) Poiščemo vejo, ki ni označena za mrtvo in vsebuje neodkljukano točko.

(b) Izberemo izjavo v neodkljukani točki.

(c) Razširimo vse žive veje, ki vsebujejo to točko in to točko odkljukamo.

(d) Končamo, če ni nobene žive veje z neodkljukano točko.

(e) Vrnemo se na točko A.

To, da smo določeno točko odkljukali, pomeni, da smo upoštevali pogoj, ki je zapisan v tej točki; razen seveda, ko pridemo do enostavnih izjav oziroma njihovih negacij, ki jih ne moremo naprej analizirati. Takšen pogoj pa moramo upoštevati na vseh živih vejah, ki to točko vsebujejo. Ko smo upoštevali vse pogoje, se lahko zgodi, da:

— končamo v A (c), to je tako, da so vse veje označene za mrtve. Tedaj je začetna množica pogojev protislovna.

— končamo v B (d). Veje, ki niso označene za mrtve, vsebujejo seznam enostavnih izjav ali njihovih negacij. Če so te izjave resnične, so resnične tudi dane izjave $A_1, \dots, A_n$.

Vsaka takšna veja nam da kakšno tako možnost.

Sedaj želimo pokazato, da iz množice izjav
$\{ A \Rightarrow B, B \Rightarrow (C \vee D), \neg C, A \}$

sledi izjava D. To pomeni, da je nemogoče, da so vse izjave iz te množice resnične, da pa je hkrati D neresnična, to je, $\neg D$ resnična. Semantična tabela za množico

$$\{ A \Rightarrow B, B \Rightarrow (C \vee D), \neg C, A, \neg D \}$$

se mora zaključiti s samimi mrtvimi vejami:

(1) $A \Rightarrow B$
(2) $B \Rightarrow (C \vee D)$
(3) $\neg C$
(4) A
(5) $\neg D$

(6) $\neg A$ (7) B

X (8) $\neg B$ (9) $C \vee D$

X (10) C (11) D

X X

Druga metoda, s katero bi rešili problem, je metoda resničnih tabel. Za prvo nalogo imamo osem kombinacij za logične vrednosti enostavnih izjav. Nato izračunamo logične vrednosti pogojev

A	B	C	A ⇒ ¬B	A ⇔ ¬C	B ∨ C
1	1	1	0	0	1
1	1	0	0	1	1
1	0	1	1	0	1
1	0	0	1	1	0
0	1	1	1	1	1
0	1	0	1	0	1
0	0	1	1	1	1
0	0	0	1	0	0

Obkrožimo tiste nabore, za katere so vsi trije pogoji resnični (imajo vrednost 1). *A* je neresnična, *C* pa resnična izjava.

Tretja možnost je sklepanje:

Recimo, da je Andrej šunko. Potem je Boris pečenko. Zahteva 3 pravi, da je Cene šunko. Skratka – Andrej in Cene oba jesta šunko, to pa je v protislovju s pogojem 2.

Torej je Andrej pečenko. Iz drugega pogoja sledi, da Cene je šunko. Tretji pogoj je avtomatično izpolnjen.

Razlike med temi tremi metodami so naslednje: metodi semantičnih in resničnostnih tabel bosta vedno dali rezultat; sta torej mehanični metodi, ki ju lahko sprogramiramo na računalniku. Metoda resničnostnih tabel kratkomalo izčrpa vse možnosti za vrednosti enostavnih izjav.

Pri metodi semantičnih tabel lahko z dobrim vrstnim redom upoštevanja pogojev (ta izbira je kreativni del metode) nekatere veje hitro zaključimo in s tem izločimo marsikateri nabor logičnih vrednosti enostavnih izjav.

Metodi sta si inverzni. Pri resničnostnih tabelah izhajamo iz vrednosti enostavnih izjav in izračunavamo vrednosti danih pogojev. Pri semantičnih tabelah izhajamo iz resničnosti pogojev in iščemo vrednosti enostavnih izjav.

Pri sklepanju pa je večkrat potrebna kakšna "ideja".

NALOGE:

1. Zapiši še druge semantične tabele za našo logično nalogo in poišči tabelo, ki vsebuje najmanj izjav!

2. Pokaži, da so naslednje izjave resnične za vse vrednosti osnovnih izjav (semantična tabela za negacije takšnih izjav vsebuje le mrtve veje):

(a) $A \Rightarrow (B \Rightarrow A)$
(b) $(A \Rightarrow B) \Rightarrow ((A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \Rightarrow C))$
(c) $A \Rightarrow (B \Rightarrow (A \wedge B))$
(d) $(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \vee B) \Rightarrow C))$

3. Janez, Peter in Tomaž so osumljeni prestopka. Na sodišču so dali naslednje izjave:

Janez: Peter je kriv, Tomaž ni kriv.
Peter: Če je Janez kriv, potem je kriv tudi Tomaž.
Tomaž: Jaz nisem kriv, toda vsaj eden od drugih dveh je kriv.

(a) Ali so vse tri izjave lahko hkrati resnične?
(b) Izjava enega sledi iz izjave drugega. O katerih govorimo?
(c) Če so vsi trije nedolžni, kdo je lagal?
(d) Če so vse izjave resnične, kdo je kriv, kdo ne?
(e) Kdo je kriv, če krivi lažejo, nedolžni pa govore resnico?

4. Upoštevaj naslednje logične zakone:

(a) Če velja $A \Leftrightarrow D$, potem lahko povsod A nadomestimo z D.
(b) $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$
(c) $A \Rightarrow B \Leftrightarrow \neg A \vee B$

Iz pogojev naše začetne naloge odpravi A in nato reši nalogo na različne načine.

Izidor Hafner