True, False. . . FAIL!

Ana Virag

Uvod (uf, previše sam pjesnički raspoložena večeras)

Gledam kroz prozor kako se nebom navlače oblaci (opet!) i razmišljam dok iz druge sobe do mene dopiru zvuci vremenske prognoze. . . I pitam se lažemo li češće nego što mislimo. Evo, sad oni meni na TV-a govore da će sutra padati snijeg . Hm, baš zgodno, ali. . . Možemo li uopće znati istinitost takvih izjava, ili, matematički preciznije, iskaza? Koju istinosnu vrijednost ima iskaz

"Sutra će sjati sunce."

u ovom trenutku i možemo li je odrediti? To me nisu učili u školi. ☹ Brzopotezno preselih svoje težište u stolac i prstići su već zaplesali po tipkovnici u nekoj svojoj zamišljenoj utrci sa samima sobom. Prvo mi je na pamet pao, naravno - Internet ☺, a tek potom MAPLE. Pa da vidim što pametniji od mene kažu.

Što je Ana saznala jedne tmurne večeri?

U redu, dosta sada odugovlačenja. Bacam se na logiku, matematiku i *matematički software*. Samo mali podsjetnik za one su imali logiku, ali im je isparila iz glave, ili one koji se još uopće nisu imali sreće susresti s njom. Unutar logike postoji grana koju zovemo **Booleova algebra**. Ona se bavi proučavanjem istinitosti iskaza te njihovim međusobnim odnosima (laički rečeno). Svakom iskazu dodjeljujemo određenu istinosnu vrijednost, već prema tome je li on istinit ili lažan. Ali, kako sam nedavno spoznala, postoje i situacije u kojima je nemoguće odrediti istinitost iskaza, a i za to postoji treća istinosna vrijednost koja se u MAPLE-u zove − **FAIL**!

FAIL i što s njim

FAIL se koristi u MAPLE-u u sklopu Booleove algebre da bismo označili nepoznatu ili neodređenu istinosnu vrijednost. Upotrebljavamo li ga zajedno s true i false, tj. istina i laž, zagazili smo u područje trovalentne logike. Postoje i istinosne tablice koje uključuju FAIL: (označen je simbolom O)

A	B	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$\neg A$
T	T	T	T	T	N
T	N	N	T	N	N
T	O	O	T	O	N
N	T	N	T	T	T
N	N	N	N	T	T
N	O	N	O	T	T
O	T	O	T	T	O
O	N	N	O	O	O
O	O	O	O	O	O

Treba napomenuti da je algebra koja uključuje FAIL valjana, ali ograničena. Uočimo da sada više ne vrijede neke istovrijednosti, npr.:

```
if(izraz) then (x) else (y)
```

sada više nije isto što i

```
if (not(izraz)) then (y) else (x).
```

Pokušajmo riješiti par primjera u MAPLE-u (negacija je definirana naredbom not, konjukcija and, a disjunkcija naredbom or). Postoji funkcija (naredba) evalb (od engleskog **eval***uate as a* **B***oolean expression*) koja određuje istinosnu vrijednost izraza. Odredimo istinitost nekih tautologija. Kako je ispisivanje svih redaka tablice često dosadan i suhoparan posao, dat ćemo računalu. U ovom slučaju MAPLE-u. Neka provede račun i provjeri umjesto nas.

Primjer 1. Provjerimo vrijedi li poznati **de Morganov**[1] zakon

$$\neg(A \vee B) = (\neg A) \wedge (\neg B)$$

```
> evalb (not(A and B)=(notA or notB));
```

true

Primjer 2. Provjerimo zakon dvostruke negacije $\neg(\neg A) = A$.

```
> evalb (not((notA))=A);
```

true

Primjer 3. Je li moguće da je istovremeno $(3 < 3) \wedge (2 \geq 1)$?

```
> evalb((3<3) and (2>=1));
```

false

Istinitost možemo provjeravati i ne koristeći se naredbom evalb:

Primjer 4. Koliko je $\mathrm{T} \wedge \mathrm{O}$?

```
> true and fail;
```

fail

Kad smo bili u okvirima logike koja uključuje samo dvije istinosne vrijednosti, postojala je određena aritmetička veza: istinosnoj vrijednosti true pridružen je broj 1, a false 0 te smo logičke operacije prikazivali matematičkim, npr.
$A \wedge B$ bilo je ekvivalentno $A \cdot B$,
$A \vee B$ s $A + B$,
...

Postoji slična ekvivalencija i za trovalentnu logiku, uz prethodni dogovor:

$$\text{fail} \to 0 \qquad \text{false} \to 1 \qquad \text{true} \to 2$$

Sada vrijedi

- $A \veebar B \to A \cdot B$ (isključna disjunkcija, tj. A i B moraju imati različitu istinosnu vrijednost da bi $A \veebar B$ bio istinit)
- $\neg A \to -A$
- $A \Leftrightarrow B \to -A \cdot B$

[1] **Augustus de Morgan**, 1806.−1871., škotski matematičar

- $A \wedge B \rightarrow (A \cdot B)^2 - (A + B) \cdot (A + B + 1)$
- $A \vee B \rightarrow -(A \cdot B)^2 + (A + B) \cdot (A + B - 1)$
- $A \Rightarrow B \rightarrow -(A \cdot B)^2 + (A - B) \cdot (A - B - 1)$

u što se vrlo lako možemo uvjeriti (probajte! ☺). Vidimo da su se izrazi malo zakomplicirali ☹ u odnosu na dvovalentnu logiku.

Još FAIL-a. . . i neke logičke funkcije

Ali prvo još malo logike. ☺ Postoji nešto što se zove zadovoljivost izraza (iskaza, suda. . .). To znači da postoji mogućnost da je iskaz istinit barem u jednom slučaju. Također postoji i valjanost (općevaljanost; tautologija): to je istina u svim kombinacijama, odnosno u svim mogućim slučajevima. MAPLE ima funkcije `coulditbe` i `is` koje relativno dobro opisuju te dvije definicije:

`is` je funkcija koja provjerava valjanost nekog iskaza i vraća `true` (ako je valjan), `false` (ako nije valjan) ili `fail` (ako se ne može odlučiti za `true` ili `false` ili nedostaje podataka).

`coulditbe` provjerava zadovoljivost, tj postoji li bar jedan slučaj u kojem je iskaz istinit i također vraća neku od triju istinosnih vrijednosti:

```
> coulditbe(a*b > 10);
```

true

```
> is(a*b < 10);
```

false

Iako nismo ništa rekli ni o a ni o b, možemo znati valjanost i zadovoljivost. Dodamo li još neke uvjete, funkcije postaju korisnije. Npr. pitamo se kada će neki izraz biti cjelobrojan ako je argument cjelobrojan:

```
> assume(i, integer);
```

(time smo odredili da je broj i cijeli broj) Sada pitamo MAPLE jesu li određeni izrazi cjelobrojni (`is`) i mogu li uopće ikada biti cjelobrojni (`coulditbe`):

```
> coulditbe(i^2-237, integer);
```

true

(odnosno, postoji neki cijeli broj i za koji je izraz $i^2 - 237$ također cijeli broj)

```
> is(i^2-237, integer);
```

true

```
> coulditbe(i/3, integer);
```

true

(postoji neki cijeli broj i za koji je izraz $i/3$ također cijeli broj)

```
> is(i/3, integer);
```

false

(nije za svaki cijeli broj i izraz $i/3$ cijeli broj; postoje brojevi i za koje $i/3$ nije cijeli broj)

```
> coulditbe(17*sqrt(i)-23, integer);
```

FAIL

(ne znamo postoji li i za koji je provjeravani izraz cjelobrojan, *kvaka* je u funkciji kvadratni korijen)

```
> is(17*sqrt(i)-23, integer);
```

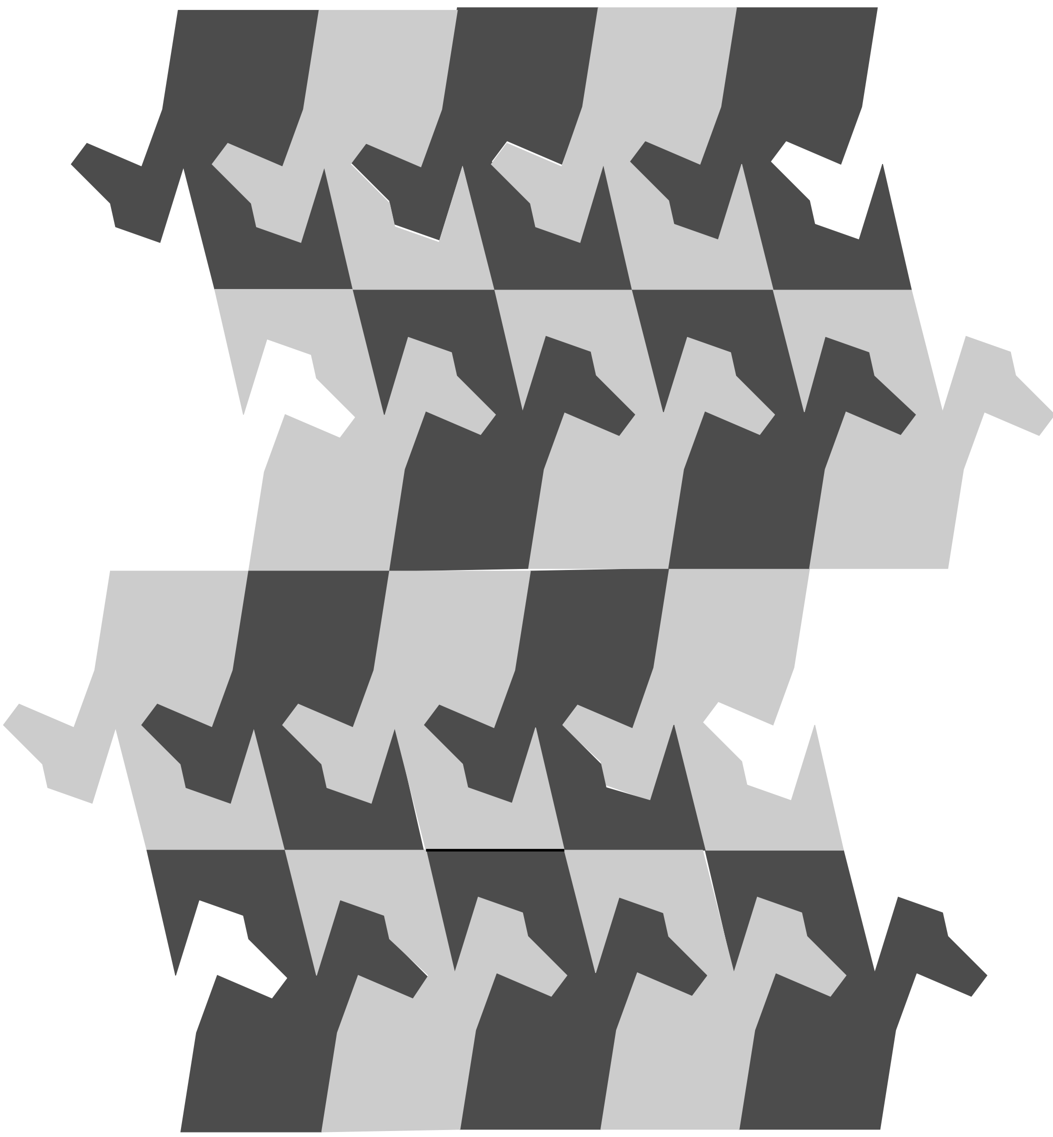
FAIL

(ma ovo nismo morali ni provjeravati jer je, kao što smo mogli zaključiti i nakon prethodnog primjera is jači uvjet, odnosno, ako je izraz true za funkciju is, sigurno je true i za coulditbe, ali obrat ne vrijedi).

Par napomena (i zadataka) za kraj...

Sad kad smo se tako lijepo zabavljali logikom i matematikom, nameće se pitanje čemu to uopće služi (osim za *neurobik* ☺, naravno). Pa, postoje još neki programski jezici koji se oslanjaju na trovalentnu logiku. Ako vas je zainteresirao članak, pokušajte otkriti koji su. A mogli biste i istražiti tko je uopće uveo treću istinosnu vrijednost u simboličku matematiku (i logiku), zar ne? A dotad... Do čitanja! I razgibavajte svoje sive stanice!

☺



Jedno popločavanje ravnine