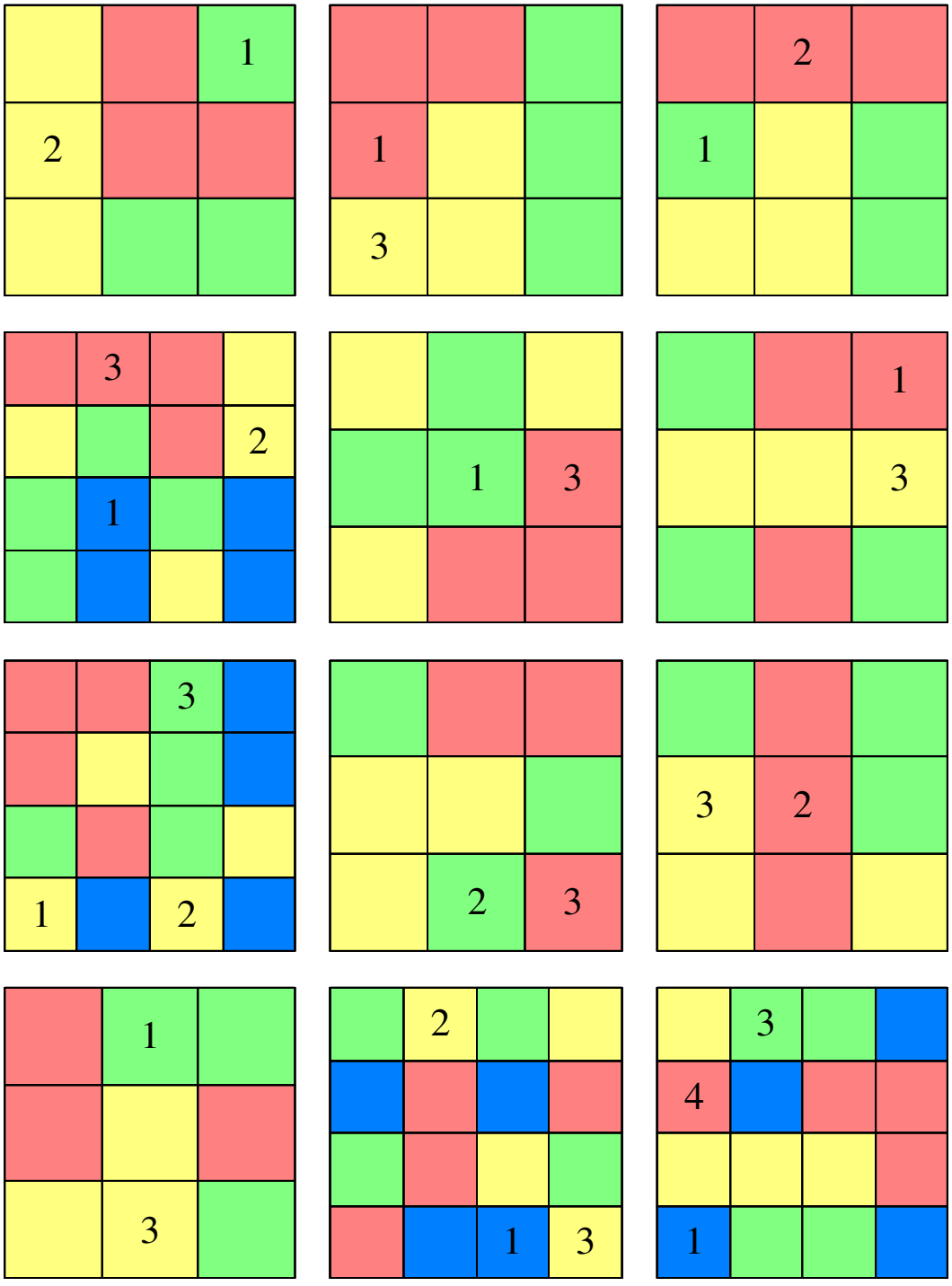
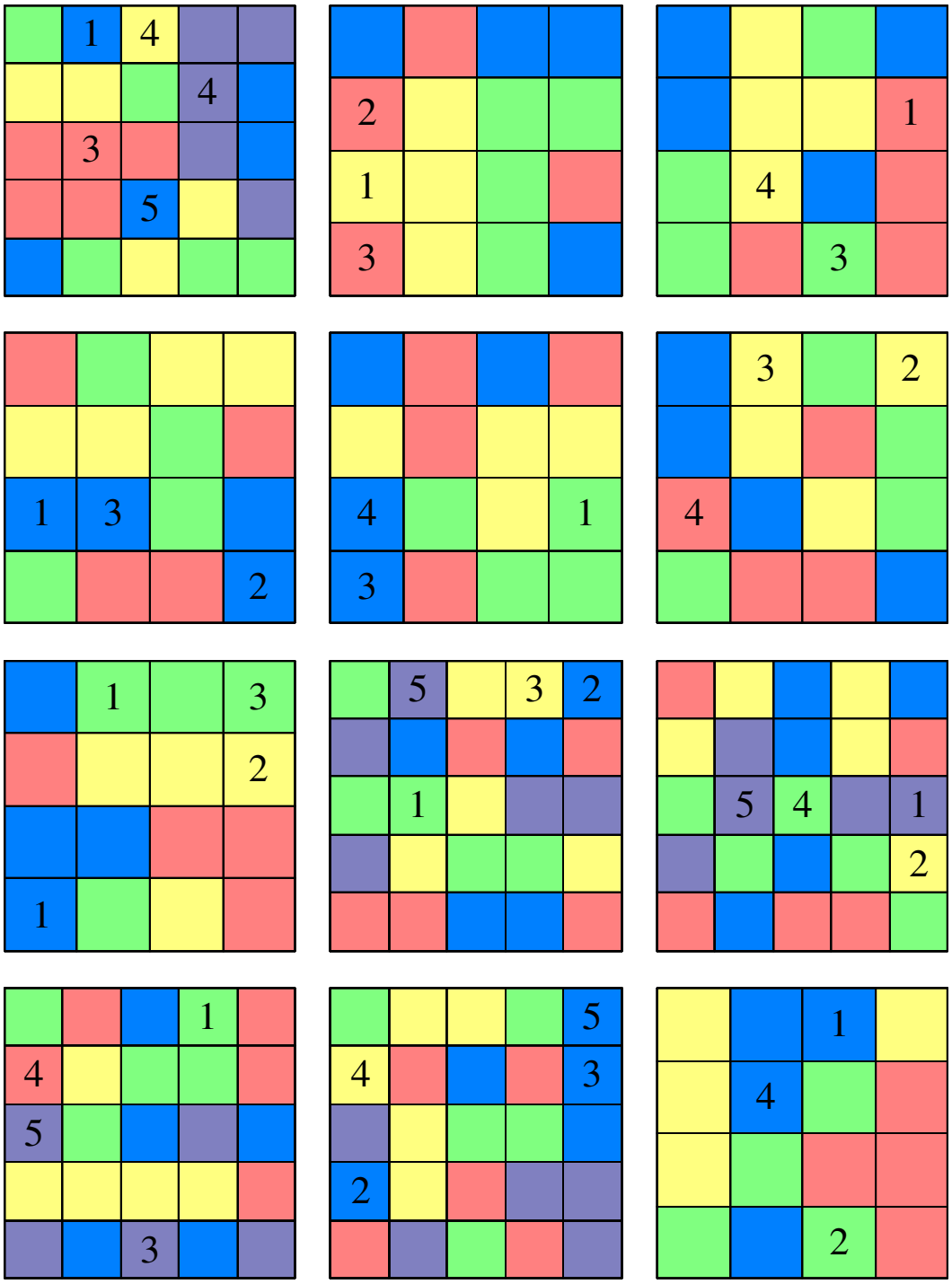
Barvni sudoku

V n × n kvadratkov moraš vpisati začetna naravna števila od 1 do n tako, da bo v vsaki vrstici, v vsakem stolpcu in v kvadratkih iste barve nastopalo vseh n števil.

1.

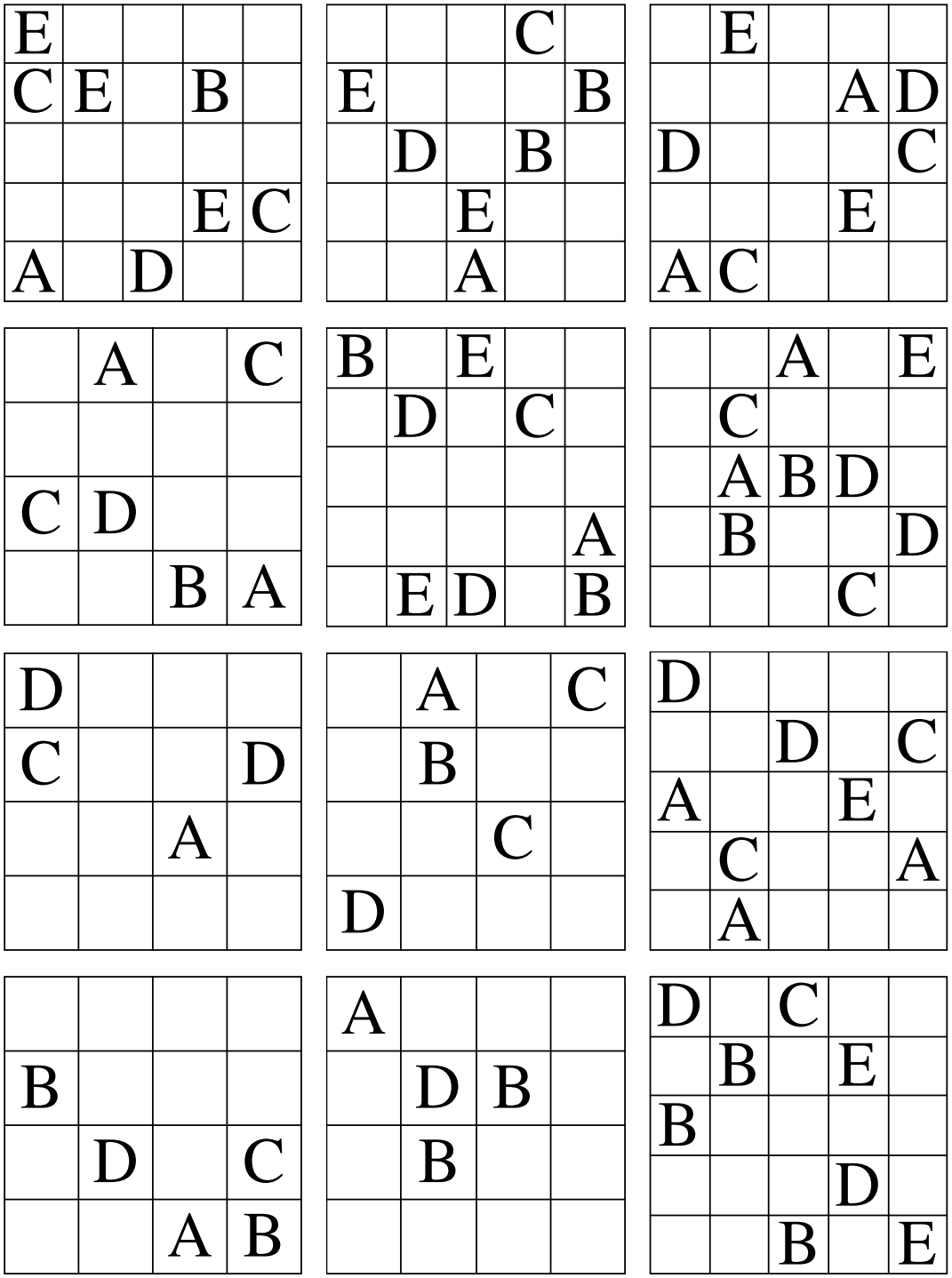


2.



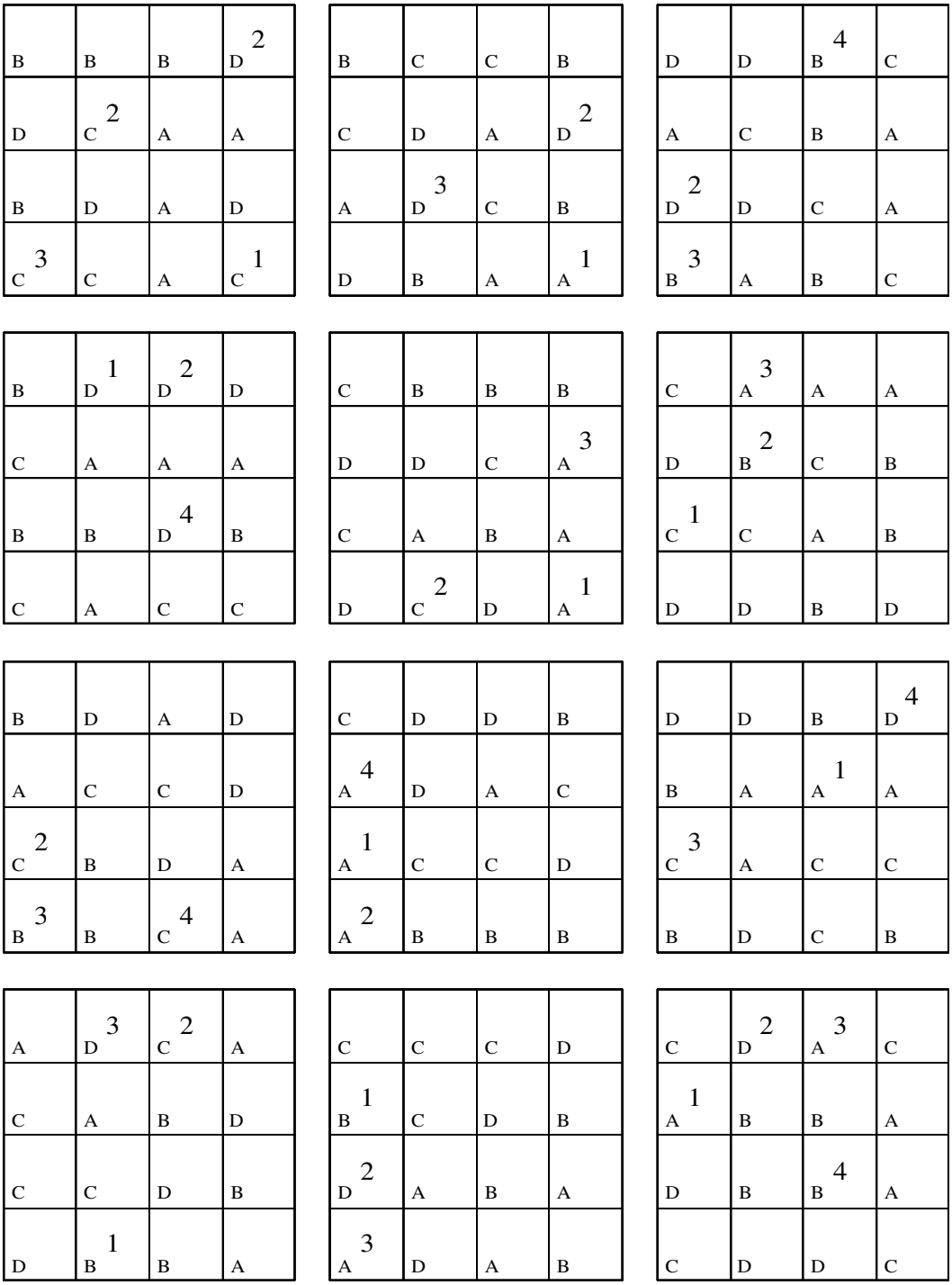
Latinski kvadrati

V n × n kvadratkov moraš vpisati začetne črke A, B, C, … tako, da bo v vsaki vrstici, v vsakem stolpcu nastopalo vseh n črk.



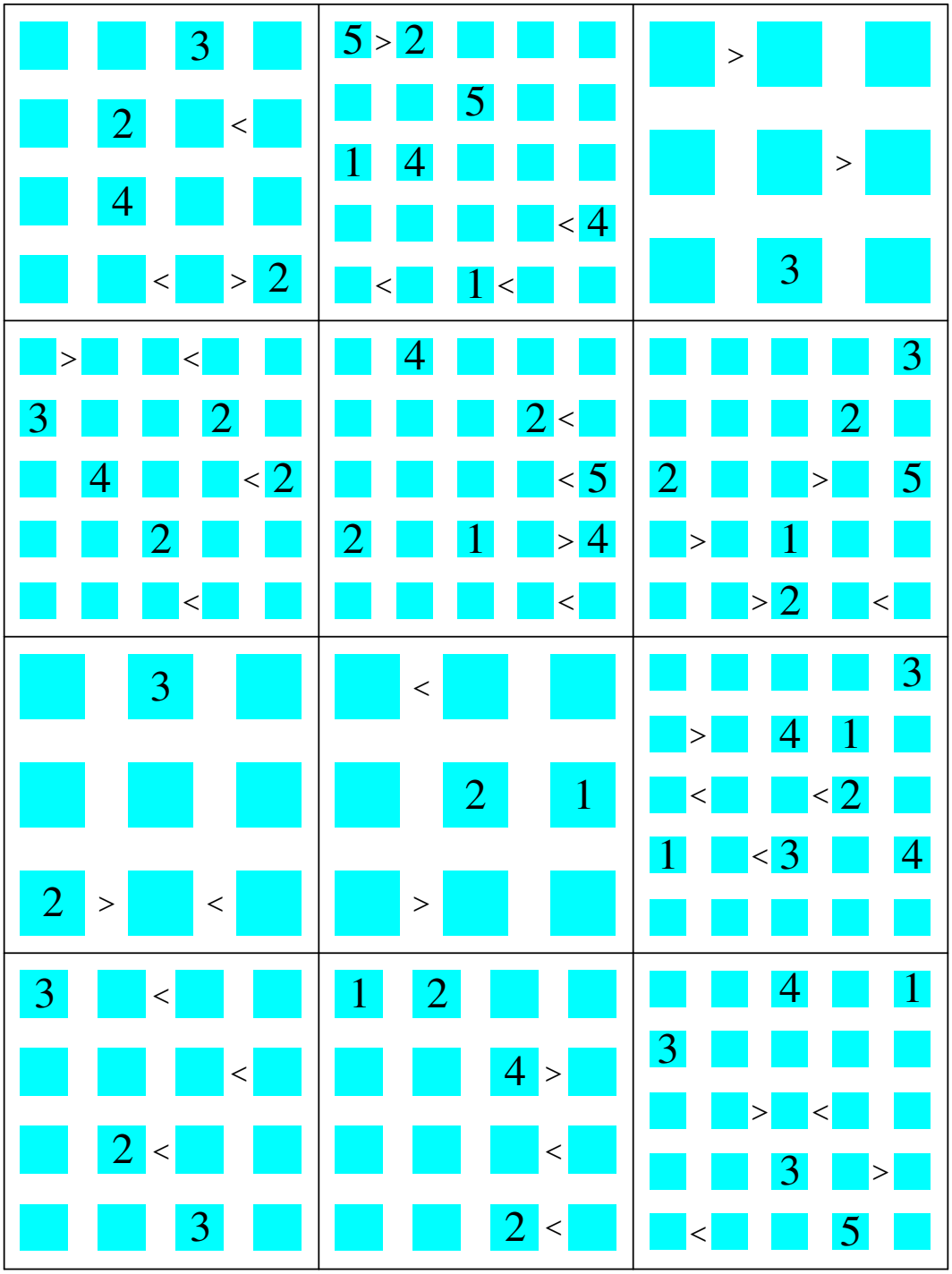
Sudoku s črkami

V n × n kvadratkov moraš vpisati začetna naravna števila od 1 do n tako, da bo v vsaki vrstici, v vsakem stolpcu in v kvadratkih z isto črko nastopalo vseh n števil.



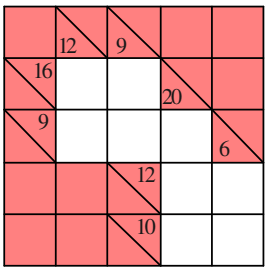
Futoshiki

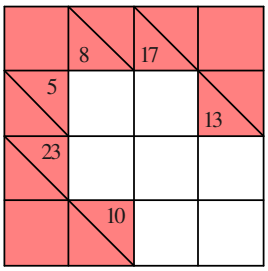
V n × n kvadratkov moraš vpisati začetna naravna števila od 1 do n tako, da bo v vsaki vrstici in v vsakem stolpcu nastopalo vseh n števil ter da bodo izpolnjene vse relacije.

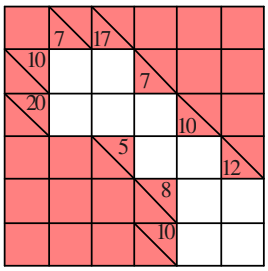


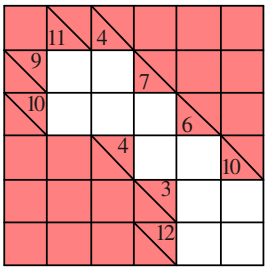
Križne vsote

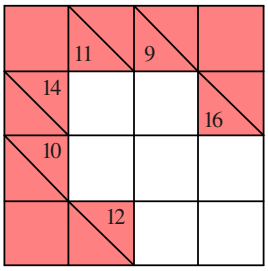
Naloga reševalca je, da izpolni bele kvadratke s števkami od 1 do 9 tako, da je vsota števk v zaporednih belih kvadratkih po vrsticah in stolpcih enaka številu, ki je zapisano v rdečem kvadratku na začetku vrstice (stolpca) nad (pod) diagonalo. Pri tem pa morajo biti vse števke v posamezni vrstici (stolpcu) različne.

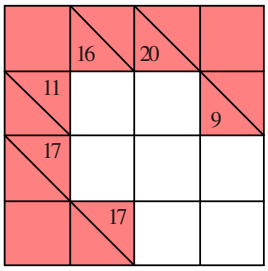


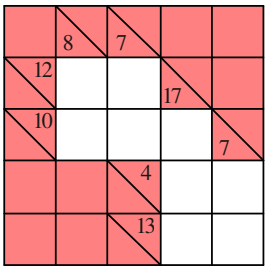


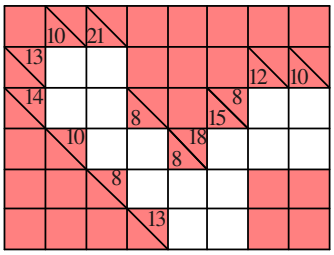


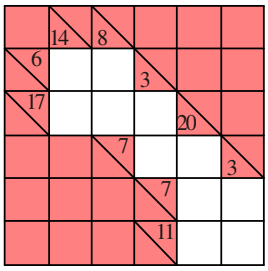


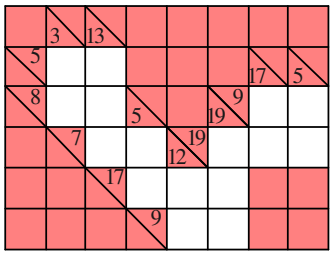


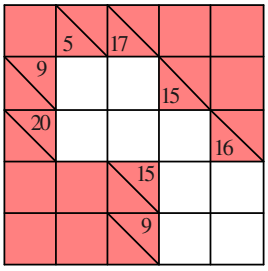


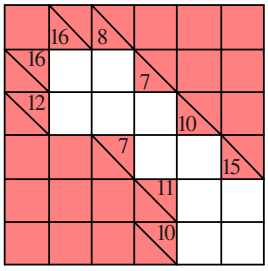






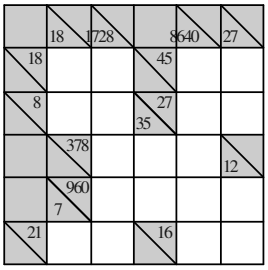


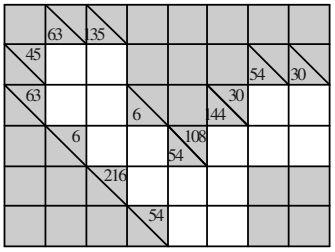


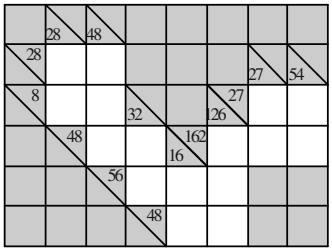


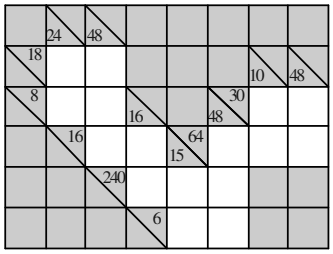
Križni produkti

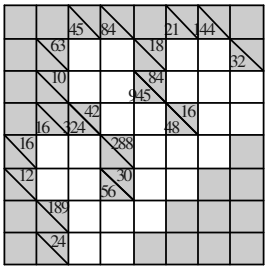
Naloga reševalca je, da izpolni bele kvadratke s števkami od 2 do 9 tako, da bo zmnožek števk v zaporednih belih kvadratkih po vrsticah in stolpcih enak številu, ki je zapisano v sivem kvadratku na začetku vrstice (stolpca) nad (pod) diagonalo. Pri tem pa morajo biti vse števke v posamezni vrstici (stolpcu) različne.

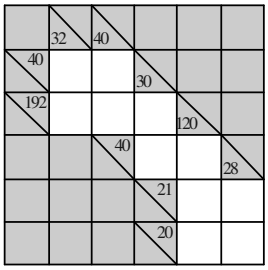


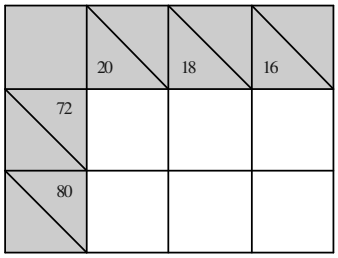


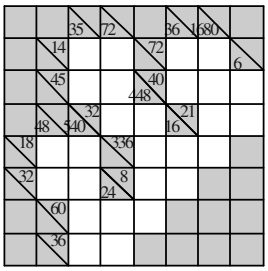


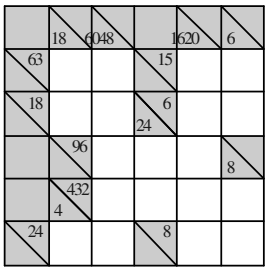


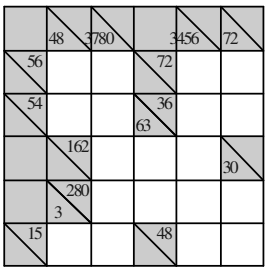


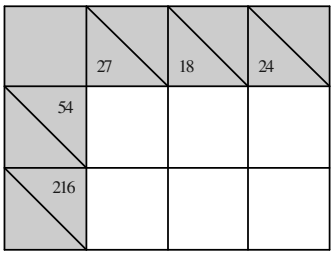


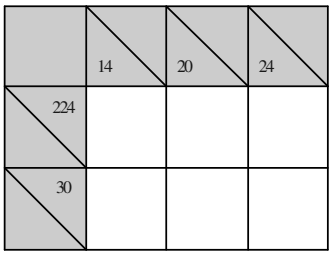






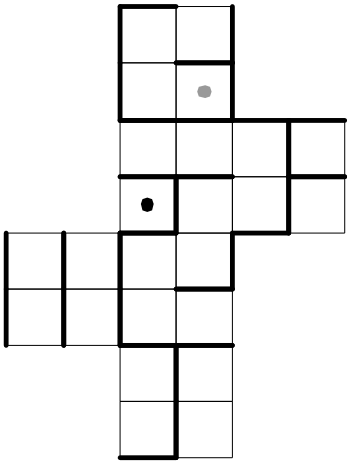


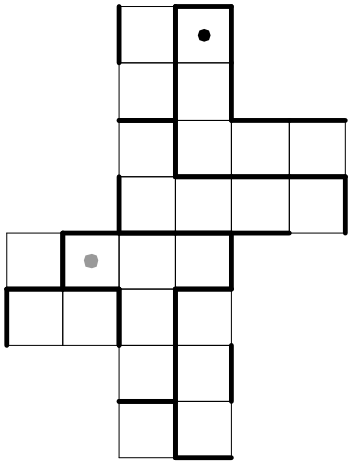


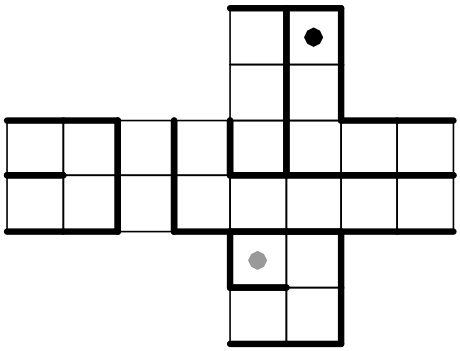


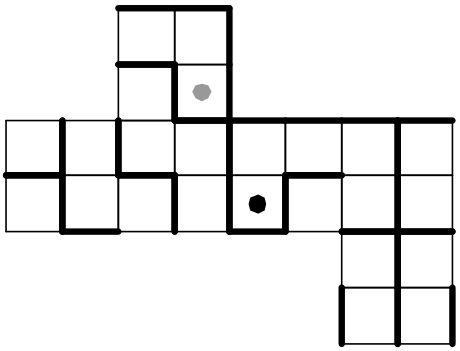
Labirint na kocki

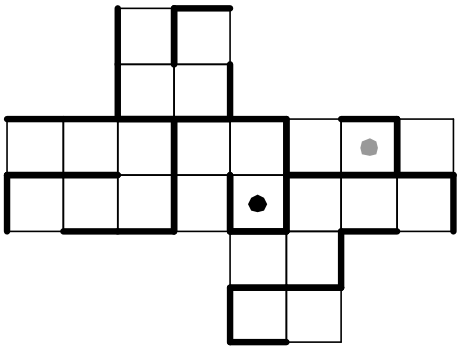
Poveži točki na kocki:

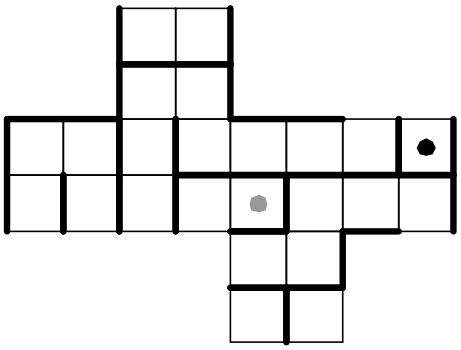






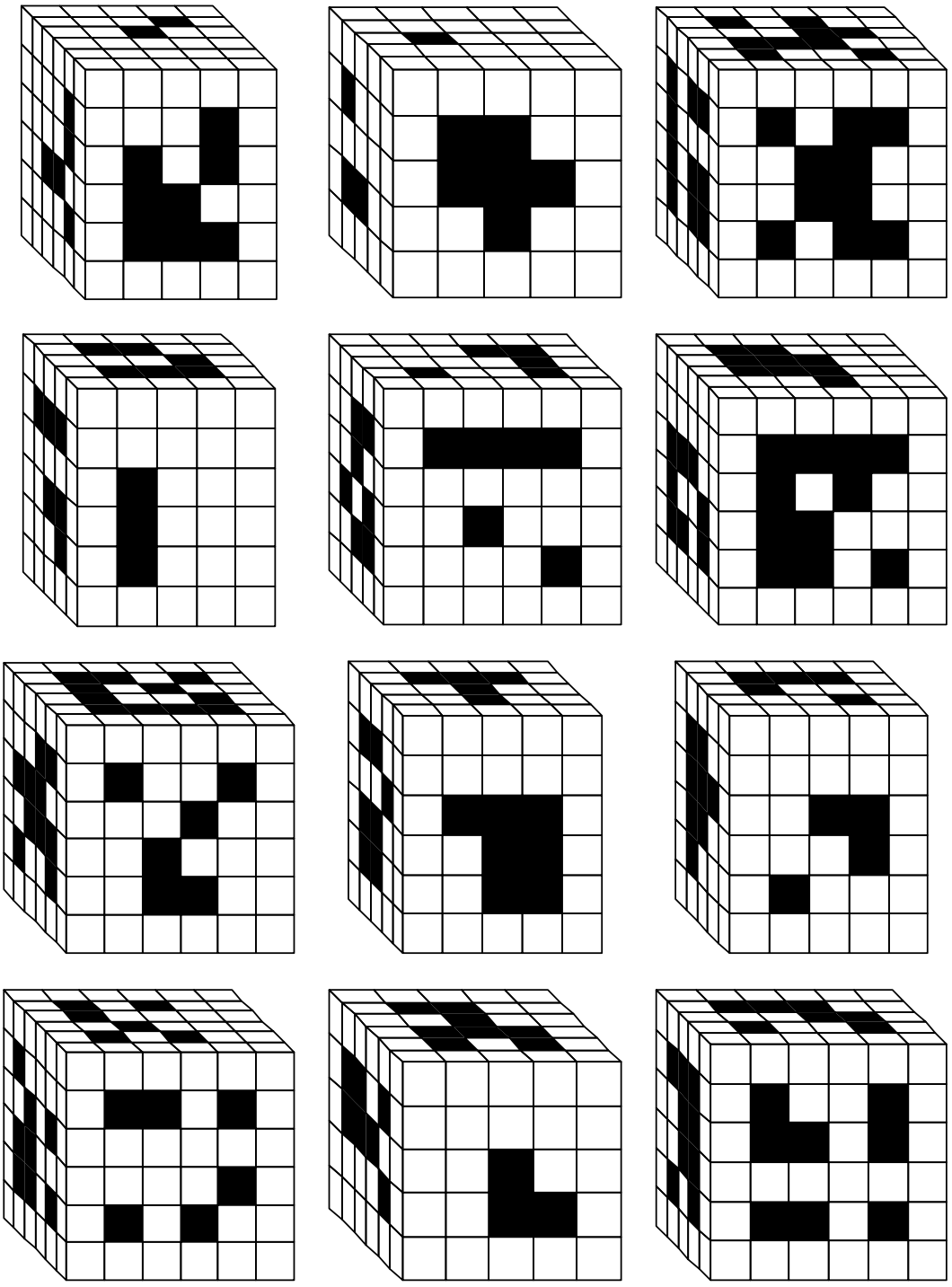




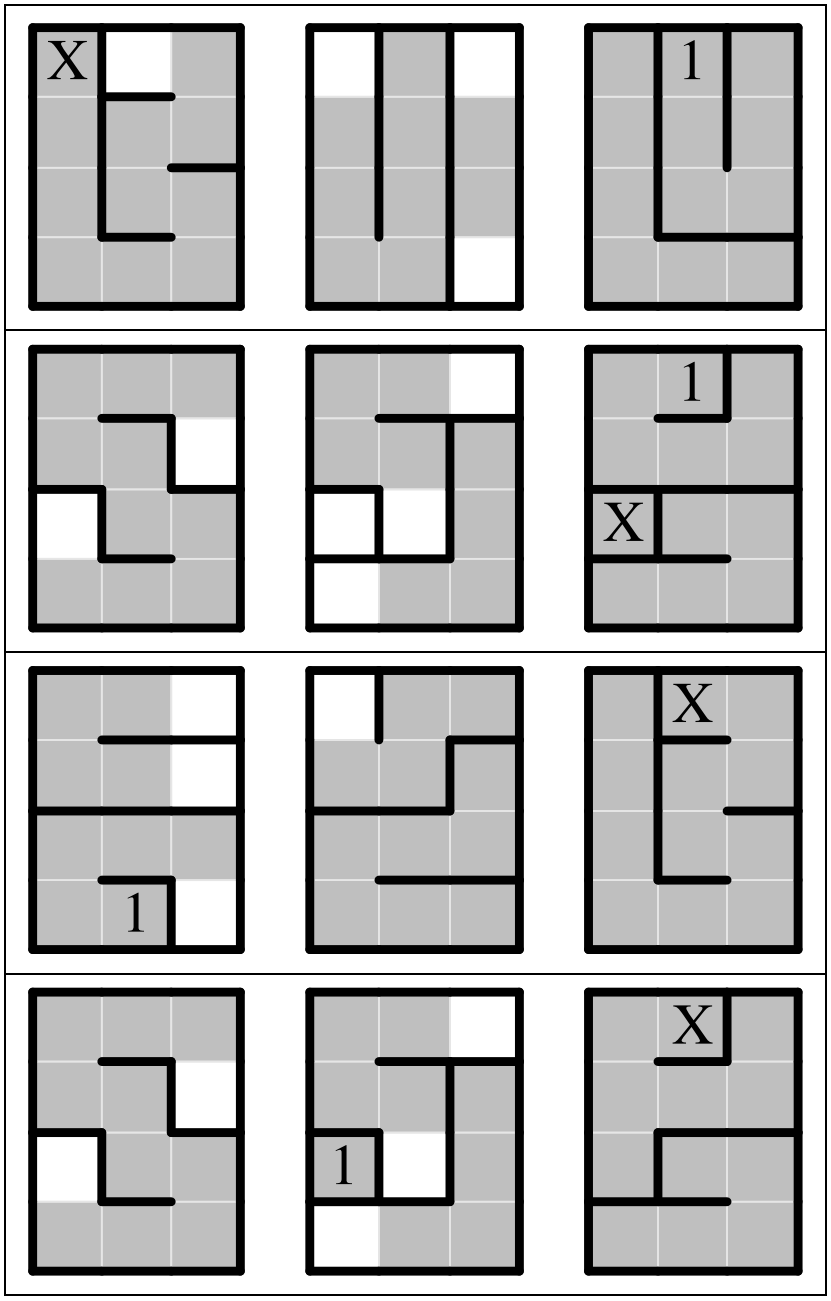


Odstranjene kocke

Dan je kvader, ki sestoji iz kockic. Odstranimo vse kocke, ki so zaznamovane črno od vrha do dna, od leve do desne in od spredaj do zadaj. Koliko kock smo odstranili?



Labirint v kvadru

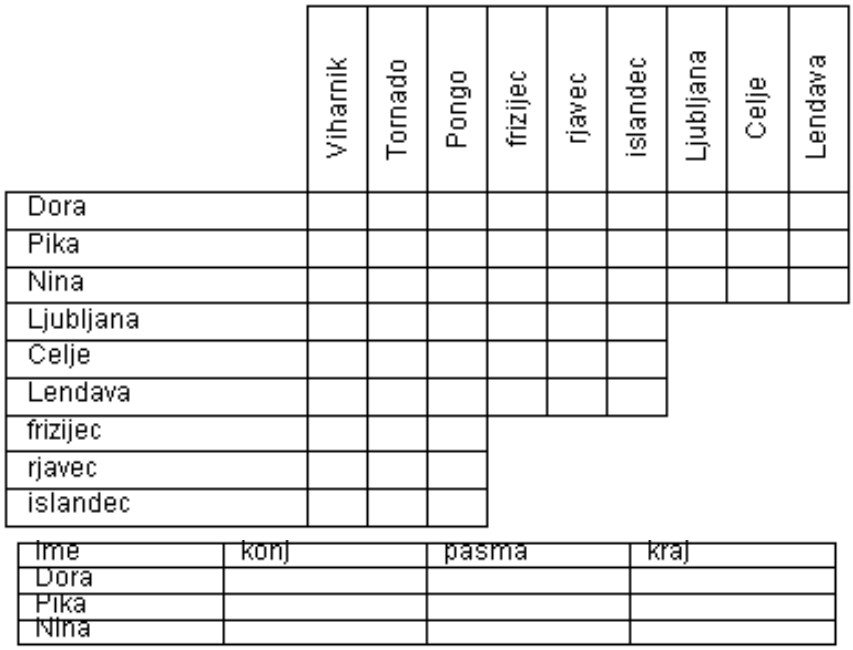


Kvader sestoji iz vodoravnih slojev kockastih oddelkov (zgornji, srednji in spodnji sloj so dani od leve proti desni). Odebeljene črte preprečujejo prehajanje med sosednjima oddelkoma istega sloja. Med oddelkom in oddelkom neposredno pod njim lahko prehajamo, če in samo če je prvi pobarvan belo.
Poišči najkrajšo pot od oddelka z 1 do oddelka z X! Pot označi z zaporednimi naravnimi števili tako, da vsak naslednji sosednji oddelek (kocko) označiš z številom, večjim za 1.

Nagradna logična naloga

Tri prijateljice (Dora, Pika, Nina) imajo različne konje (Viharnik, Tornado, Pongo), ki so različnih pasem (frizijec, rjavec, islandec) in so iz različnih krajev (Ljubljana, Celje, Lendava.
Za vsako določi ime, konja, pasmo konja in kraj bivanja.

1. Pongo ni iz Celja.
2. Tornado je iz Lendave.
3. Rjavec ni iz Celja.
4. Dora ni doma ne iz Celja ne iz Ljubljane.
5. Pika ni doma iz Ljubljane.
6. Rjavec ni iz Ljubljane.
7. Frizijec ni iz Ljubljane.



Rešitev nagradne uganke pošljite do 1.9.2017 na naslov Logika d.o.o., Svetčeva pot 11, 1241 Kamnik, s pripisom »Nagradna uganka«.
Naslednji reševalci nagradne uganke iz 3. številke bodo prejeli poševno prizmo Polydron in Mercatorjevo vrtavko »Disney Frozen«: R. B., M.Š., M.I., R. K. in M.M., vsi iz ILIRSKE BISTRICE.

medex

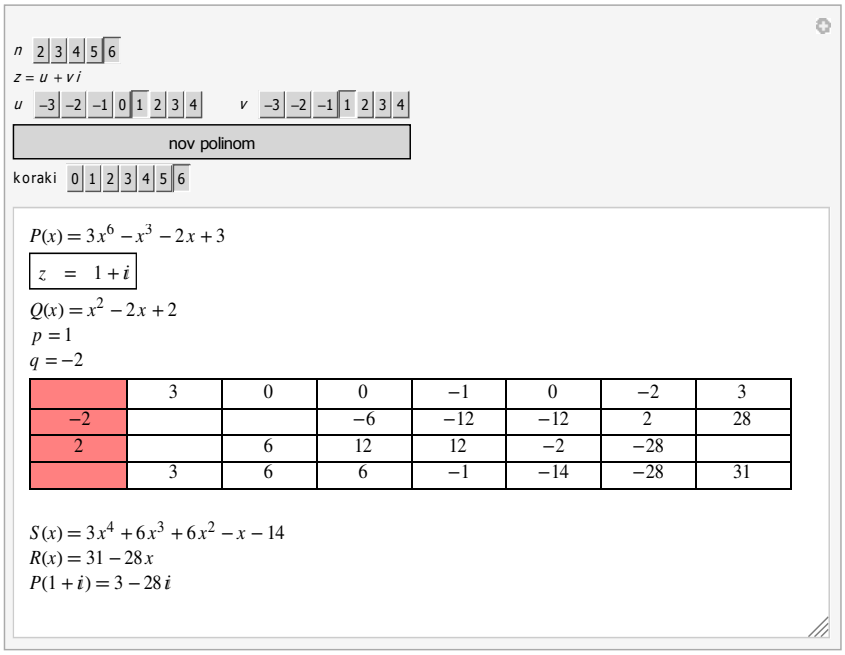
Ruffini-Hornerjev algoritem v kompleksnem

Ruffini-Hornerjev algoritem uporabljamo pri računanju vrednosti polinoma $P(x)=a_n x^n+ a_{n-1} x^{n-1}+...+a_1x+a_0$ pri nekem realnem številu c. Polinom P(x) podelimo z x-c, dobimo ostanek R, ki je konstantna funkcija in velja P(c)=R(c).
Zdaj pa predpostavimo, da imamo polinom $P(x)=a_n x^n+a_{n-1} x^{n-1}+...+a_1x+a_0$ z realnimi koeficienti in kompleksno število c=u+iv. Bodi c*=u-iv konjugirano kompleksno število k c. Zdaj delimo polinom P(x) s kvadratnim polinomom $(x-c)(x-c^*)=x^2-2u\,x+u^2+v^2=x^2-2px-q$, kjer je p=u in $q=-u^2-v^2$.
Ostanek R(x) je linearna funkcija in valja P(c)=R(c).
Kako dobimo koeficiente količnika in ostanka?
Velja $a_n x^n+a_{n-1} x^{n-1}+...+a_1x+a_0=(b_n x^{n-2}+b_{n-1} x^{n-3}+...+b_3x+b_2)(x^2-2px-q)+ b_1x+b_0$. Tu je ostanek $R(x)= b_1x+b_0$, koeficienti količnika pa so b_i, i=2, 3, …, n. Če pomnožimo desno stran in izenačimo koeficiente pri potencah, dobimo:

$a_n = b_n$
$a_{n-1} = b_{n-1}- 2pb_n$
$a_{n-2} = b_{n-2}- 2pb_{n-1}-qb_n$
…
$a_1 = b_1- 2pb_2-qb_3$
$a_0 = b_0-qb_2$

Koeficiente b_i izračunamo po shemi:

	a_n	a_{n-1}	a_{n-2}	…	a_1	a_0
q			qb_n		qb_3	qb_2
2p		$2pb_n$	$2pb_{n-1}$		$2pb_2$	
	$b_n= a_n$	b_{n-1}	b_{n-2}		b_1	b_0



	3	0	0	−1	0	−2	3
−2			−6	−12	−12	2	28
2		6	12	12	−2	−28	
	3	6	6	−1	−14	−28	31

Zgoraj računamo vprednost polinoma P(x) pri 1+i. Ostanek je Rx)=-28x+3, R(1+i)=3-28 i..
Prednost tega postopka je tudi v tem, da računamo s kompleksnimi števili le malo na začetku, pri računanju q in p ter na koncu R(u+iv).

Nalogi:

n 2 3 **4** 5 6

$z = u + v\,i$

u −3 −2 −1 0 1 **2** 3 4 v −3 −2 −1 **1** 2 3 4

nov polinom

koraki 0 1 2 **3** 4

$P(x) = 3x^4 - x$

$z = 2 + i$

$Q(x) = x^2 - 4x + 5$

$p = 2$

$q = -5$

	3	0	0	−1	0
−5					
4					
	3				

n 2 3 4 **5** 6

$z = u + v\,i$

u −3 −2 −1 0 1 **2** 3 4 v −3 −2 −1 **1** 2 3 4

nov polinom

koraki 0 1 2 3 **4** 5

$P(x) = 3x^5 - x^2 - 2$

$z = 2 + i$

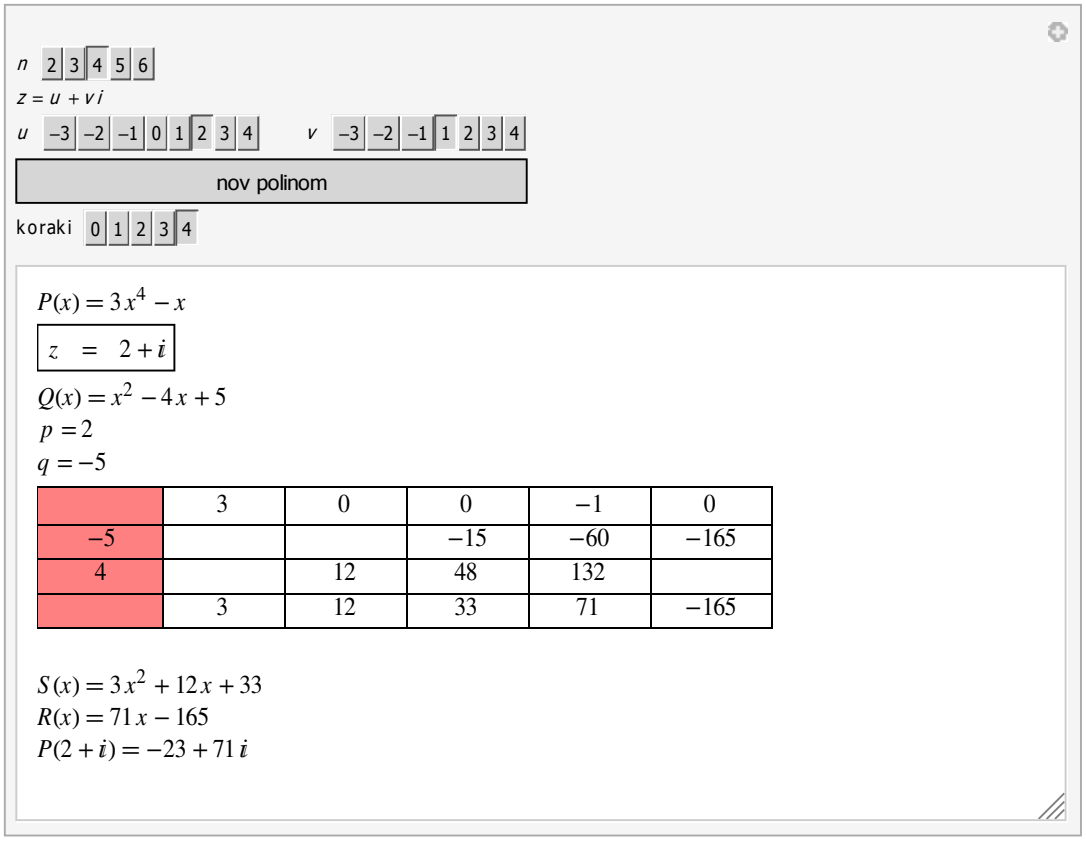
$Q(x) = x^2 - 4x + 5$

$p = 2$

$q = -5$

	3	0	0	−1	0	−2
−5						
4						
	3					

Rešitvi:



$P(x) = 3x^4 - x$

$z = 2 + i$

$Q(x) = x^2 - 4x + 5$

$p = 2$

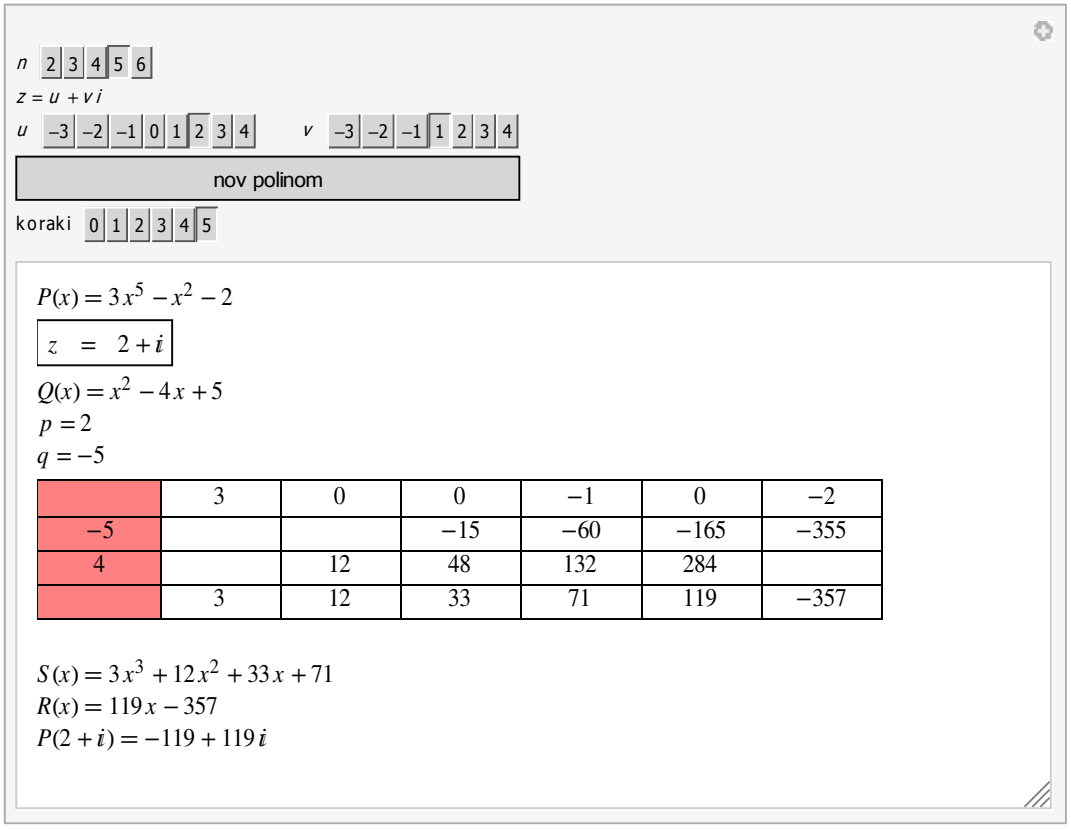
$q = -5$

	3	0	0	−1	0
−5			−15	−60	−165
4		12	48	132	
	3	12	33	71	−165

$S(x) = 3x^2 + 12x + 33$

$R(x) = 71x - 165$

$P(2 + i) = -23 + 71i$



$P(x) = 3x^5 - x^2 - 2$

$z = 2 + i$

$Q(x) = x^2 - 4x + 5$

$p = 2$

$q = -5$

	3	0	0	−1	0	−2
−5			−15	−60	−165	−355
4		12	48	132	284	
	3	12	33	71	119	−357

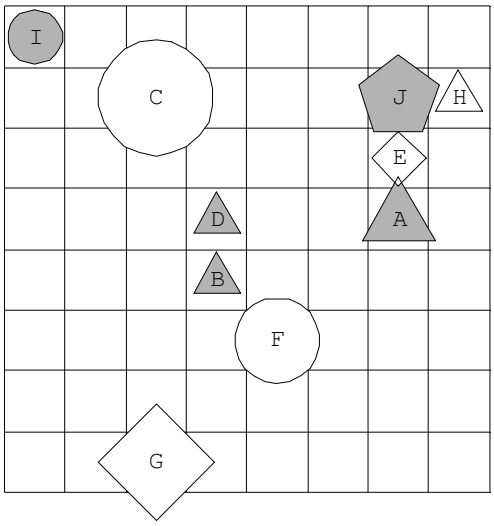
$S(x) = 3x^3 + 12x^2 + 33x + 71$

$R(x) = 119x - 357$

$P(2 + i) = -119 + 119i$

Trovrednostna logika

Do zdaj smo se ukvarjali s klasično dvovrednostno logiko. Vsaka izjava je bila ali resnična ali neresnična. Vrednost sestavljene izjave smo izračunali po pravilih definicij izjavnih povezav. V programu *Hyperproof*, ki po začetnih optimističnih napovedih ni nikoli zaživel v Windows okolju, je postalo možno nekoliko drugačno izračunavanje resničnostnih vrednosti. Podobno kot pri programu *Svet Tarskega* imamo v *hyperproofu* situacijo, ki bi prilagojena naši inačici, to je likom, lahko izgledala takole:



Liki so lahko trikotniki, kvadrati in petkotniki. Kaj pa so I, C in F? Velik krog pomeni, da je lik velik, ne vemo pa, katere oblike. Temu rečemo, da je situacija *nepopolna*. Stavek »Lik C je trikotnik.« ni ne resničen ne neresničen. Zato bomo rekli, da je njegova vrednost nedefinirana ali vprašljiva (ali nedoločena), za kar bomo uporabljali znak U (undefined). V hyperproofu nam nedoločenost pomeni pomanjkanje našega vedenja, pod krogom se namreč skriva pravi lik.

Kako bomo računali z novo vrednostjo? Kakšna je vrednost stavka: Lik D je majhen, F pa je srednje velikosti? Tokrat sta oba stavka resnična, zato je resničen njun stik (konjunkcija). Kaj pa »Lik D je trikotnik in I petkotnik«? Tokrat je drugi del nedefiniran, zato je tudi celota nedefinirana. Podobno je »Lik D je trikotnik ali pa je I petkotnik.« resničen, saj je resničen prvi del, drugega nam sploh ni treba gledati.

Tokrat bomo podali izračun samo za štiri izjavne povezave:

`p in q`	`p & q`	`p and q`
`p ali q`	`p v q`	`p or q (or both)`
`če p, potem q`	`p -> q`	`if p then q`
`p natanko tedaj, kadar q`	`p <-> q`	`p if and only if q`

Tabela izgleda takole:

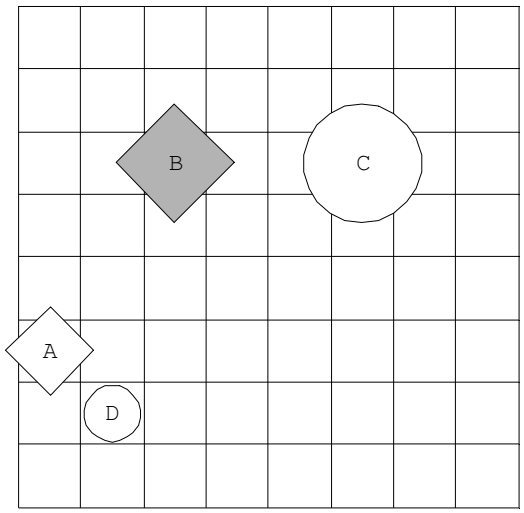
p	q	p & q	p v q	p -> q	p <-> q
R	R	R	R	R	R
R	N	N	R	N	N
R	U	U	R	U	U
N	R	N	R	R	N
N	N	N	N	R	R
N	U	N	U	R	U
U	R	U	R	R	U
U	N	N	U	U	U
U	U	U	U	U	U

Izključujoči "ali ali" bomo obravnavali kot negacijo ekvivalence. Za negacijo velja tabela:
$\neg R = N$, $\neg N = R$, $\neg U = U$.

Opomba: Prvi sistem trovrednostne logike je vpeljal poljski logik Łukasiewicz. Njegova tabela se malenkostno razlikuje od Kleenejeve pri implikaciji in ekvivalenci. Lukasiewicz je resnico označeval z 1, neresnico z 0 in nedoločeno vrednost (pri njem je „možna") z ½. Njegova tabela je takšna:

p	q	p & q	p v q	p -> q	p <-> q
R	R	R	R	R	R
R	N	N	R	N	N
R	U	U	R	U	U
N	R	N	R	R	N
N	N	N	N	R	R
N	U	N	U	R	U
U	R	U	R	R	U
U	N	N	U	U	U
U	U	U	U	R	R

Vrnimo se zdaj k našim situacijam.



V tem svetu so resnične izjave: Lik A je kvadrat ali lik D ni petkotnik. Če je lik D petkotnik, potem je C velik. Če je lik B trikotnik, potem je lik C tudi trikotnik. Dva lika sta velika.
Neresnične so na primer izjave: Lik A je trikotnik in lik D je kvadrat. Lik D je velik in je trikotnik. Ali je A trikotnik ali je B trikotnik. Samo en lik je velik.
Nedoločene so izjave: C in D sta trikotnika. B in C sta kvadrata. A je trikotnik ali pa je D trikotnik. Če je D trikotnik, potem je tudi C trikotnik. Če je D trikotnik, potem je A petkotnik. Vsi liki so kvadrati. Obstaja vsaj en trikotnik.

Oba omenjena sistema trovrednostne logike sta razširitvi klasične dvovrednostne logike v smislu, da če zbrišemo vrstice z U, dobimo klasične tabele.

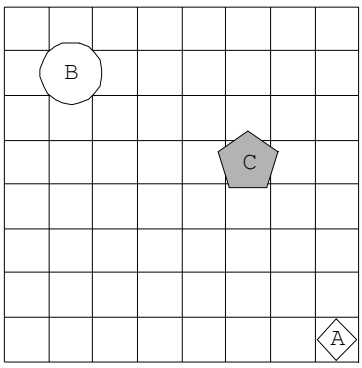
Naloge

1. Ugotovi resničnostno vrednost danih stavkov, podanih v 2 svetovih

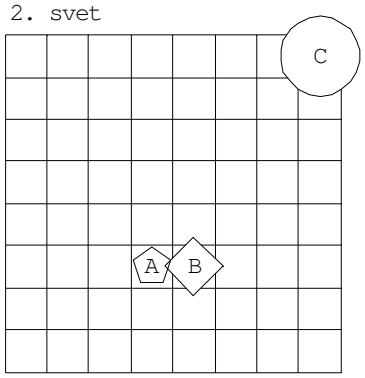
1. Lik B je srednje velikosti.
2. Lik B je kvadrat.
3. Lik A je bel.
4. Lik C je petkotnik.
5. Lik A je velik.
6. Lik C je kvadrat.
7. Lik C je trikotnik.
8. Lik C je majhen.
9. Lik B je velik.
10. Lik C je siv.
11. Lik B je kvadrat, če in samo če lik A ni srednje velikosti.
12. Lik C ni majhen ali lik A ni velik.
13. Ali lik C ni trikotnik ali lik B ni kvadrat.
14. Lik B je trikotnik, če in samo če lik C ni majhen.
15. Ali lik C ni srednje velikosti ali lik C ni trikotnik.
16. Lik B ni kvadrat, če in samo če je lik B bel.
17. Lik B je srednje velikosti, če in samo če je lik B siv.
18. Če lik A ni bel, potem lik C ni bel.
19. Če lik B ni kvadrat, potem lik C ni bel.
20. Lik B ni kvadrat in lik B je bel.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1																				
2																				

1. svet



2. svet

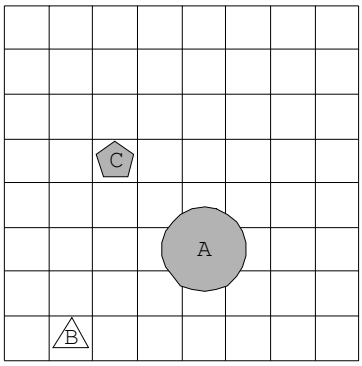


2. Ugotovi resničnostno vrednost danih stavkov, podanih v 2 svetovih

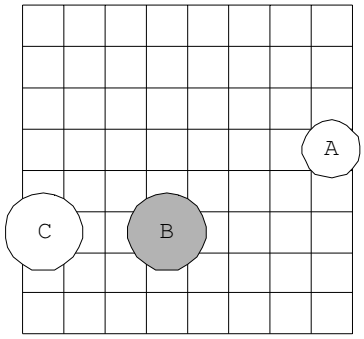
1. Lik C ni bel, če in samo če lik A ni kvadrat.
2. Lik A je siv ali je lik A petkotnik.
3. Lik A je trikotnik in lik A ni petkotnik.
4. Lik A ni trikotnik in lik B je srednje velikosti.
5. Lik B je kvadrat in lik B ni petkotnik.
6. Lik C ni bel ali lik A ni trikotnik.
7. Ni res, da: ali je lik C siv ali je lik B kvadrat.
8. Ni res, da: ali lik C ni trikotnik ali lik C ni petkotnik.
9. Ni res, da: lik B je kvadrat in lik C je majhen.
10. Obstaja tak x, da za vsak y velja: lik x je desno od y.
11. Obstaja tak x, da za vsak y velja: lik x je nad y.
12. Obstaja tak x, da za vsak y velja: lik x je nad y.
13. Za vsak x obstaja tak y, da velja: lik x je majhen in lik y je velik.
14. Za vsak x obstaja tak y, da velja: lik x je petkotnik, če in samo če lik y ni siv.
15. Za vsak x obstaja tak y, da velja: lik x ni trikotnik in lik y je srednje velikosti.
16. Obstaja tak x, da za vsak y velja: lik x ni bel, če in samo če je lik y srednje velikosti.
17. Obstaja tak x, da za vsak y velja: če lik x ni velik, potem je lik y majhen.
18. Obstaja tak trikotnik x, da za vsak bel lik y velja: lik x je pod y.
19. Obstaja tak bel lik x, da za vsak bel lik y velja: lik x je pod y.
20. Za vsak siv lik x obstaja tak kvadrat y, da za vsak bel lik z velja: lik x je desno od y in lik y je manjši kot z.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1																				
2																				

1. svet



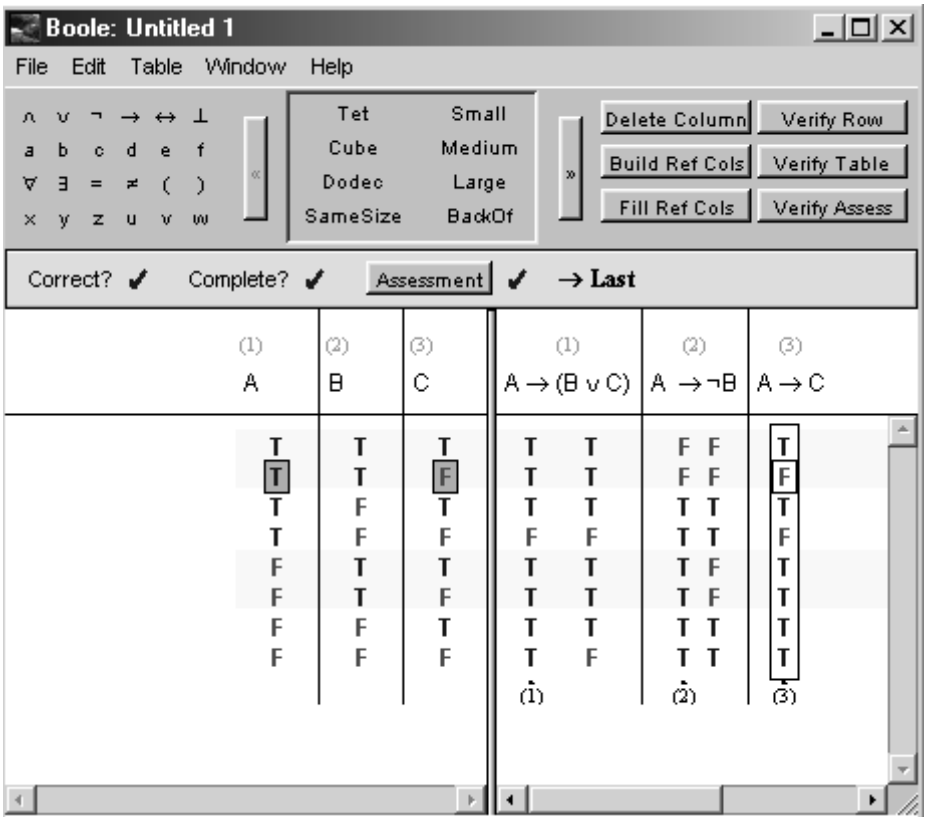
2. svet



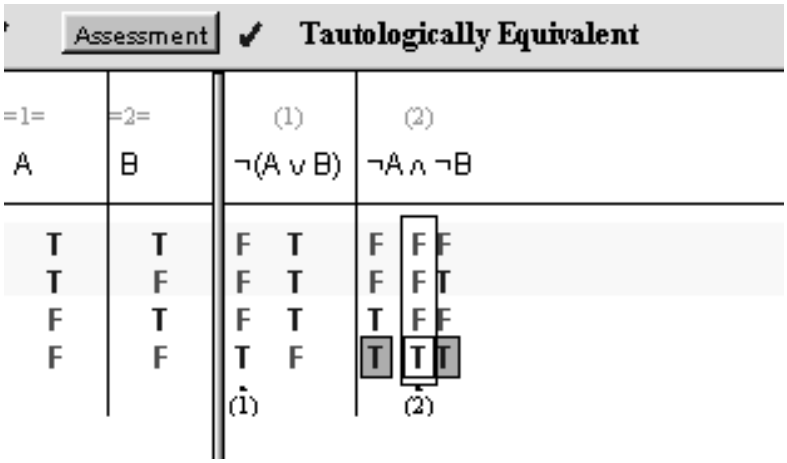
Program Boole

Knjigo Language, Proof and Logic, ki sta jo napisala Barwise in Etchemendy (CSLI Publications, Stanford, 1999) uporabljajo za poučevanje logike na več kot 100 univerzah po celem svetu. Knjigi je dodan še paket s tremi programi in možnost reševanja nalog, ki jih preverja strežnik, ki pošlje rezultat tako študentu kot učitelju. Celoten paket je zgled za sodoben učni pripomoček, kakršnega bi lahko imeli tudi drugi predmeti.

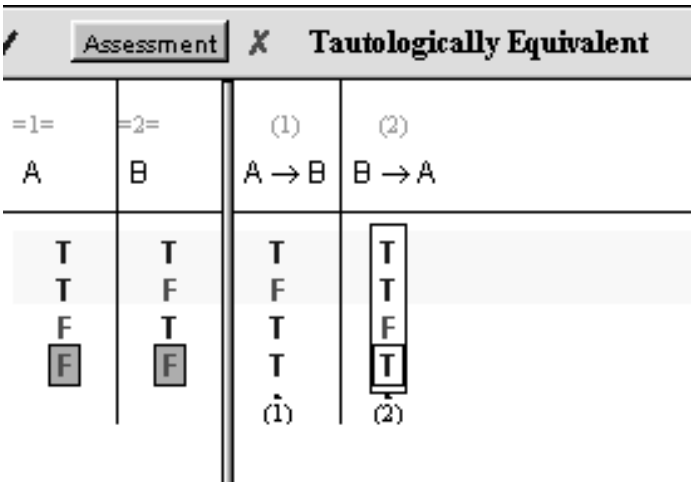
Tri računalniške aplikacije so Tarski's world, program za razumevanje jezika 1. reda, Fitch, program za konstrukcijo dokazov, in Boole, program za izdelavo resničnostnih tabel. O prvem programu smo že večkrat pisali. Tokrat se posvetimo programu Boole (ime je dobil po slavnem logiku Georgu Booleu).



Zgornja slika predstavlja okno programa s tipičnim primerom. V levi del vhodnega okna vnesemo osnovne (atomarne) izjave. Ni nam treba vnašati vseh naborov, ker to naredimo z gumbom “Fill Ref Cols”. Pač pa moramo izračunati vrednosti sestavljenih izjav na desni. Pravilnost verificiramo z gumboma “Verify Row” ali “Verify Table”. Cilji (assessment), ki jih zasledujemo, so: ali je izjava tavtologija, ali sta dani dve izjavi tavtološko ekvivalentni, ali zadnja izjava tavtološko sledi iz prejšnjih, ali so izjave neprotislovne (vse so resnične pri vsaj enem naboru). Verificiranje De Morganovega zakona prikazuje spodnja slika.



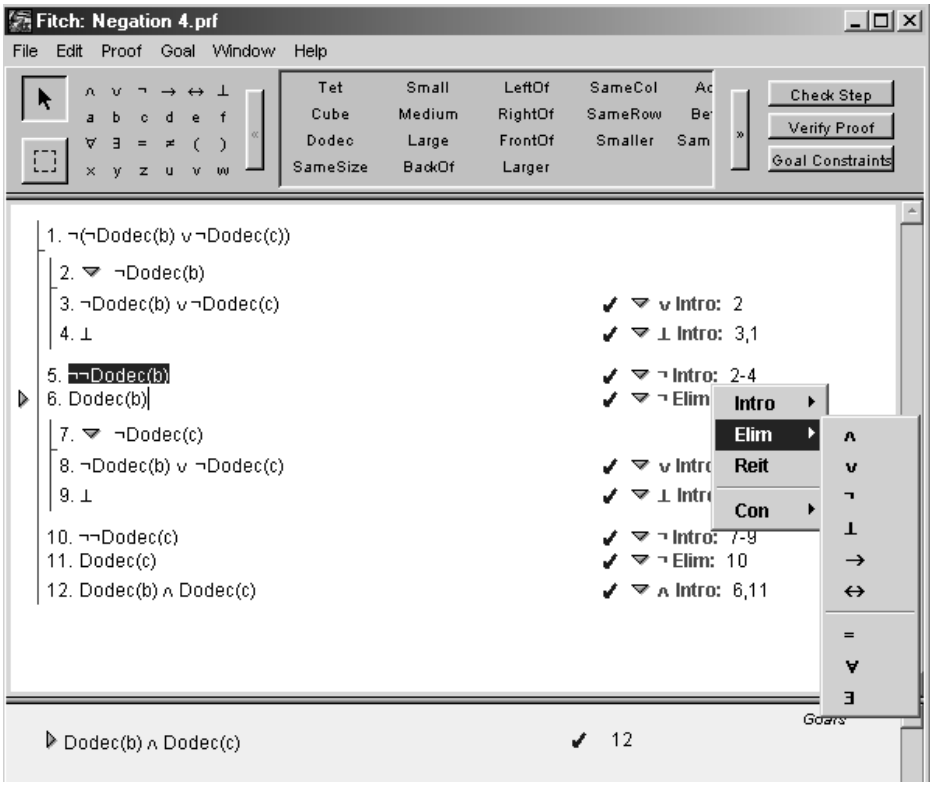
Zgled na prejšnji strani pa je primer, ko je zadnja izjava (3) tavtološka posledica prejšnjih (1,2). Zgled, ko dve izjavi nista tavtološko ekvivalentni, je podan spodaj.



Čeprav je program Boole od vseh treh najenostavnejši, ga lahko koristno uporabimo pri pouku in za reševanje domačih nalog

Program Fitch in sistem naravne dedukcije

Program Fitch nam omogoča konstrukcijo dokazov v jeziku prvega reda. Pravila sklepanja, ki jih uporablja, so znana pod imenom naravna dedukcija. Pravila sklepanja delimo na pravila uvedbe (introdukcije) in pravila izločitve (eliminacije). Za vsak sistem logike je nujno, da lahko izpeljemo le veljavne izjave (to je izrek zdravosti) in da izpeljemo vse veljavne izjave (izrek o polnosti ali popolnosti). Pri aksiomatiki pogosto vztrajamo, da je pravil čim manj. Seveda pa to pomeni daljše dokaze. V praksi seveda izpeljane izreke uporabljamo tako kot aksiome in na ta način dokazi postajajo krajši. Edino in glavno vprašanje pri programu je, ali neka izjava logično sledi iz danih izjav. Če uspemo zgraditi izpeljavo izjave iz danih predpostavk, potem je odgovor pritrdilen. Kaj pa, če izjava ni izpeljiva iz predpostavk. V tem primeru moramo poiskati svet, v katerem so vse predpostavke resnične, prva izjava pa ni. Tega pa ne moremo narediti s programom Fitch, lahko pa to naredimo s programom Tarski's world.



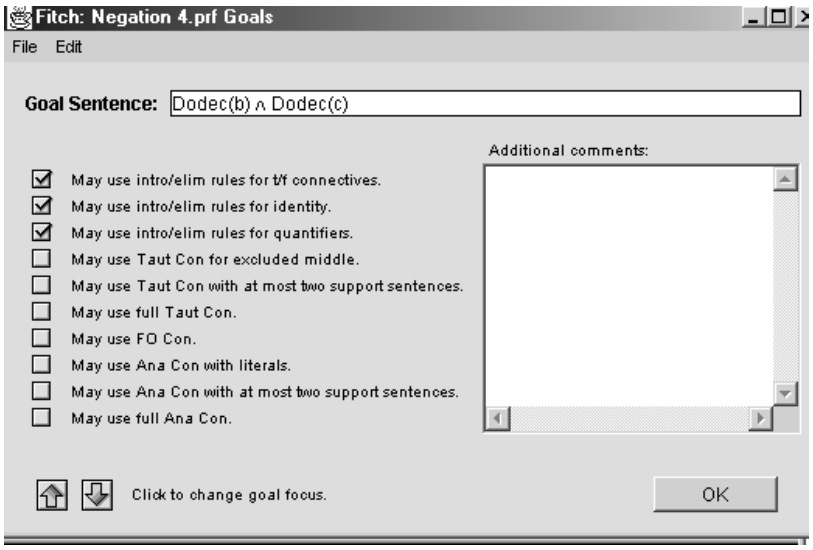
Slika prikazuje tipično okno programa Fitch. V resnici gre za dokaz, da $A \wedge B$ logično sledi iz $\neg(\neg A \vee \neg B)$.
Program omogoča začetek nove naloge, odpiranje obstoječih dokazov, shranjevanje (File); izrezovanje, kopiranje in prestavljanje delov dokaza(Edit); dodajanje korakov v dokazu, uvajanje poddokazov, preverjanje dokazov (Proof), postavljanje ciljev in omejitev (Goal) in dostop do oken (Window).

Orodna vrstica vsebuje gumbe za vnašanje stavkov, verifikacijo koraka, verifikacijo dokaza in določitev omejitev.

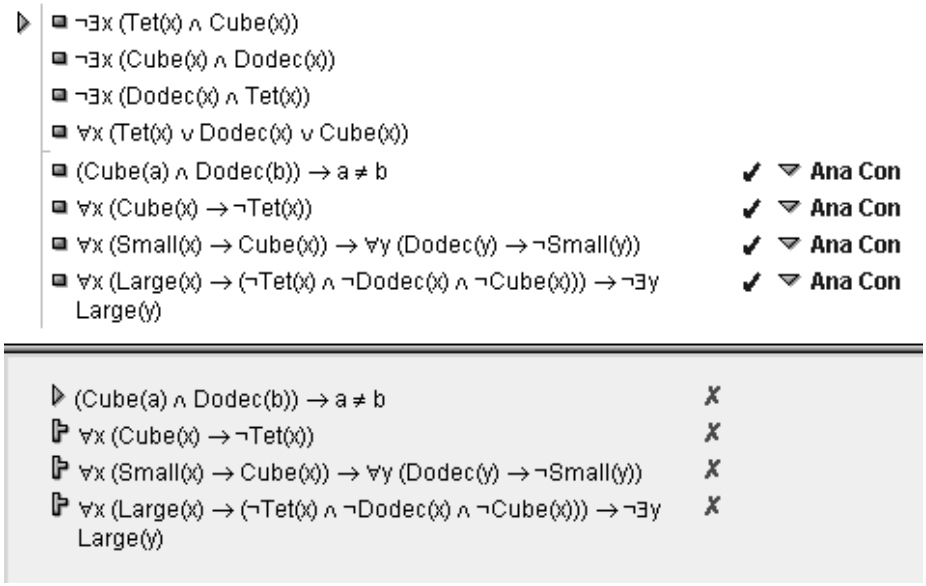
Glavno delo se izvaja v dokaznem oknu, ki je razdeljeno na dva dela. Na levem delu zgornjega dela vnašamo korake dokaza, na desnem pa razlago, to je, povemo na osnovi katerega pravila sklepanja in iz katerih prejšnjih korakov sledi sklep. Ko vnašamo nov korak, se nam pojavi zahteva po pravilu (Rule?), ki ga izberemo iz menija. Pravila naravne dedukcije so, kot vemo, pravila uvedbe in izločitve.

Spodnji del nam pove, kaj je cilj dokaza. V primeru napake lahko dobimo sporočilo v spodnji vrstici.

Navpična črta na levi se imenuje Fitchova palica, majhna vodoravna črtica loči predpostavke od drugega dela dokaza. Urejujemo lahko le tekočo vrstico, ki je določena s trikotnim premikajočim drsnikom na levi. Nove vrstice vrinemo preko menija (Proof). Pri pojasnevanju koraka moramo razen pravila navesti tudi predhodne vrstice, iz katerih korak sledi. Podporne vrstice izberemo s klikom nanje.

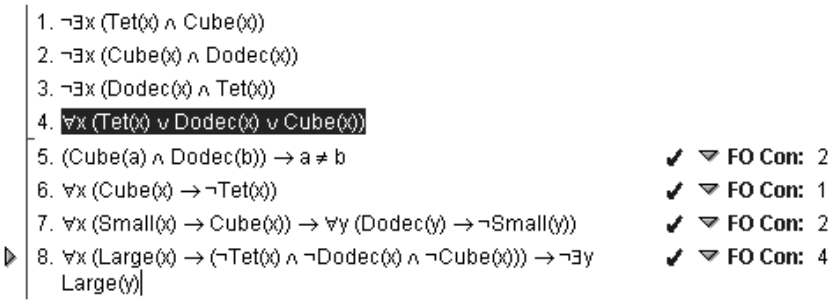


V ciljnem oknu zapišemo izjave, ki jih želimo dokazati (ne nujno samo zadnjo) in navedemo omejitve glede uporabe pravil. Zdaj navedimo nekaj primerov.



Poleg osnovnih pravil naravne dedukcije imamo na razpolago še tri zelo močna pravila: tavtološka posledica, logična posledica in analitična posledica. Tavtološka posledica pomeni, da izjava sledi samo z uporabo pravil za izjavne povezave. Logična posledica (FO Con) pomeni, da izjava sledi na osnovi vseh logičnih pravil (pravil logike prvega reda). Analitična posledica pa vključuje tudi pravila, ki veljajo za telesa v svetu Tarskega (npr. če je telo veliko, potem ni majhno).

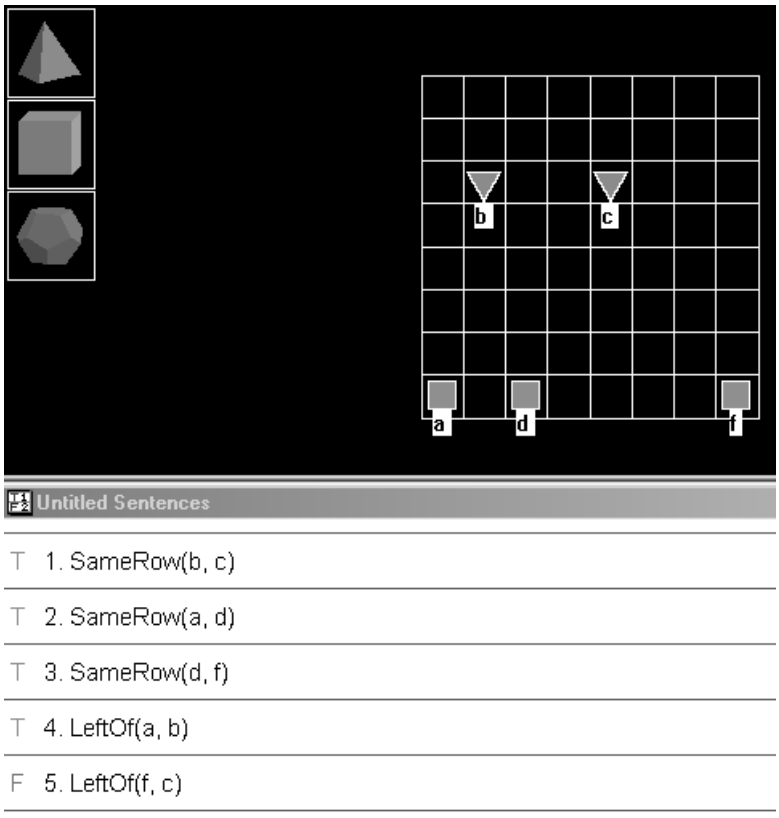
Zgornje posledice res veljajo v svetu Tarskega in to neodvisno od predpostavk. Vendar pa cilji niso doseženi. Ne smemo namreč uporabiti pravila analitične posledice, to je lastnosti sveta Tarskega. V resnici pa so logične posledice, kar je razvidno iz dokaza



V naslednjem primeru zadnja izjava ni analitična posledica predhodnih izjav.



To nam pove * v dokaznem polju. Vendar pa ne vemo, zakaj. Odgovor na to vprašanje je konstrukcija sveta Tarskega, v katerem so vse štiri predpostavke resnične, sklep pa napačen. Tule je svet.



Sintetična in analitična metoda reševanja logičnih nalog

Sinteza pomeni ustvarjanje sestavljenih objektov iz enostavnejših, ***analiza*** pa razstavljanje sestavljenih na enostavne. Torej sta to dve nasprotni operaciji.

Oglejmo si naslednjo znano logično nalogo. Andrej, Boris in Cene so bili osumljeni nekega kriminalnega dejanja. Na sodišču so dali tele izjave:
Andrej: »Boris je kriv, Cene pa ni.«
Boris: »Če je Andrej kriv, potem je tudi Cene kriv.«
Cene: »Jaz nisem kriv, toda vsaj eden od drugih dveh je kriv.«
Kdo je kriv, če so bile njihove izjave pred sodnikom resnične?

Za atomarne (enostavne) bomo vzeli naslednje izjave, ki jih bomo označili s črkami A, B in C:
A = Andrej je kriv. B = Boris je kriv. C = Cene je kriv. Če uporabimo simbole $\wedge$, $\vee$, $\rightarrow$ ($\Rightarrow$), $\neg$ za »in«, »ali«, »če, potem« in »ne«, potem lahko izjave osumljenih zapišemo simbolično, tako kot je zapisano v treh desnih stolpcih spodnje tabele. (V logiki povezavam rečemo konjunkcija, disjunkcija, implikacija in negacija. Večkrat pa uporabimo tudi ekvivalenco $\Leftrightarrow$.)
Ker imamo 3 atomarne izjave (vsaka je resnična ali neresnična), imamo 8 različnih naborov resničnostnih vrednosti. Pri vsakem naboru izračunamo vrednosti sestavljenih izjav (vrednosti vpišemo pod ustrezen simbol).

(1) A	(2) B	(3) C	(1) B ∧ ¬C	(2) A → C	(3) ¬C ∧ (A ∨ B)
T	T	T	F F	T	F F T
T	T	F	T T	F	T T T
T	F	T	F F	T	F F T
T	F	F	F T	F	T T T
F	T	T	F F	T	F F T
F	**T**	F	T T	T	T T **T**
F	F	T	F F	T	F F F
F	F	F	F T	T	T F F
			(1)	(2)	(3)

Vse tri sestavljene izjave so resnične samo v 6 vrstici. Torej Andrej in Cene nista kriva, Boris pa je.
Temu sintetičnemu postopku se v logiki reče izdelava resničnostne tabele.

Zdaj pa začnimo s sestavljenimi izjavami. Kdaj bo resnična Andrejeva izjava. Odgovor: Če sta resnični izjavi B in ¬ C. Izjava A → C je resnična, če je A neresnična ali je C resnična. Ker C ni resnična, A ni resnična. Cenetova izjava je resnična, če je resnična Andrejeva izjava. Naše sklepanje zapišemo v obliki drevesa:

$$B \wedge \neg C$$
$$A \rightarrow C$$
$$\neg C \ \wedge (A \vee B)$$
$$B$$
$$\neg C$$
$$\neg A \quad | \quad C$$
$$X$$

Najprej smo napisali predpostavke, to je dane tri izjave. Potem smo analizirali Andrejevo izjavo. Za njeno resničnost morata biti resnični izjavi B in ¬C. Za resničnost Borisove izjave pa imamo dve možnosti. Prvotno drevo se razcepi v dve veji: ¬A | C. Toda veja, ki gre na C, vsebuje tudi ¬C. Taki veji pravimo, da je zaprta ali protislovna in jo označimo z X. Imamo pa eno odprto vejo, iz katere razberemo vrednosti atomarnih izjav. Cenetove izjave sploh nismo analizirali, saj je resnična, če je resnična Andrejeva izjava.

Opisani analitični metodi v logiki pravijo metoda *semantičnih tabel* (dreves).

Ni težko iz semantičnega drevesa prebrati dokaza:
Vzemimo, da so vse tri izjave osumljenih resnične. Zaradi Andrejeve izjave je Boris kriv, Cene pa ni. Borisova izjava je resnična, če Andrej ni kriv ali če je kriv Cene. Toda Cene ni kriv, torej tudi Andrej ni kriv.

Dokaz lahko zapišemo v nekem formalnem sistemu (npr. program Fitch):



V zgornjem dokazu so koraki oštevilčeni, v spodnjem niso.



Dokaz je zaporedje stavkov. Vsak stavek zapišemo v svojo vrstico. Najprej so dane predpostavke (premise), ločene pa so od drugih stavkov z oznako na desni črti. Za vsako nadaljnjo vrstico moramo povedati, po katerem pravilu sklepanja in iz katerih predhodnih vrstic sledi. Fitchov sistem je t.i. sistem naravne dedukcije (prva sta ga uvedla Gentzen in Jaskowski). Pravila sklepanja se delijo na pravila uvedbe (introdukcije) in pravila izločitve (eliminacije). Vsaka logična operacija ima po dve pravili. Vrstici 5 in 6 sta dobljeni z eliminacijo konjunkcije. To je preprosto pravilo, ki ga v vsakdanjem dokazovanju kar preskočimo. Vrstice 6-8 imenujemo poddokaz. Vrstica 6 je dodatna predpostavka (v vsakdanjem jeziku rečemo: pa naj velja še A). C smo dobili z izločitvijo implikacije. To pravilo je v logiki znano kot modus ponens. Toda zdaj smo prišli do tega, da veljata C in ¬C. Temu pravimo protislovje, njegova oznaka pa je ⊥ (obrnjeni T). Če smo iz A izpeljali

protislovje, mora veljati ¬A. Temu pravilu se reče reductio ad absurdum. Zdaj vemo za vrednosti vseh enostavnih izjav, velja torej njihova konjunkcija (*stik* po prof. Križaniču).

Opazimo, da je metoda semantičnih dreves precej hitrejša, zato jo kaže kar največkrat uporabljati. Metodo resničnostnih tabel včasih poimenujejo tudi mehanična metoda. Z njo enostavno pregledamo vse možnosti. Seveda nam to lahko porabi veliko časa.

Odgovorimo še na eno vprašanje: Če so vsi trije nedolžni, kdo je lagal. V resničnostni tabeli si ogledamo vrstico z naborom FFF. Lagala sta Andrej in Cene. Tu seveda analitična metoda ni uporabna, saj ne vemo vrednosti sestavljenih izjav. Vendar pa tudi ni potrebno izdelati celotne resničnostne tabele, ampak le eno vrstico.

Za sestavo resničnostnih tabel smo uporabili program Boole. Tu F(false) pomeni neresnico T(truth) pa resnico.

Kako pa bi odgovorili na vprašanje: Če krivi lažejo in nedolžni govorijo resnico, kdo je kriv?
V resničnostni tabeli moramo pregledati vse vrstice in ugotoviti ali je kakšna vrstica, v keteri imajo atomarne izjave nasprotno vrednost od sestavljenih. To je v 3 vrstici. Kriva sta Andrej in Cene.
Kaj pa analitična metoda? Tokrat se iščejo vrednosti atomarnih izjav, za katere so resnične naslednje izjave:
$\neg A \Leftrightarrow B \wedge \neg C$
$\neg B \Leftrightarrow A \rightarrow C$
$\neg C \Leftrightarrow \neg C \wedge (A \vee B)$.
Tokrat je semantično drevo kar obsežno in je takorekoč ekvivalentno pregledovanju vseh možnosti.

Lahko pa pridemo do rešitve s sklepanjem: Recimo, da Andrej ni kriv. Potem je zaradi njegove izjave Boris kriv, Cene pa ne. Ker je Borisova izjava napačna, mora biti Andrej kriv (Cene pa ne). To je protislovje. Andrej je torej kriv. Ker je A resnična izjava (ne A pa napačna) je B neresnična ali C resnična. Ali je lahko B resnična izjava. Potem je A resnična, C pa ne. To pa je protislovje.

Tokrat smo imeli 3 enostavne izjave, zato je v resničnostni tabeli 2^3=8 vrstic. Če pa bi imeli 8 atomarnih izjav, bi imeli 256 vrstic, ki bi jih le težko izračunali v 30 min, kolikor je na voljo na tekmovanju. Nalogo z 10 atomarnimi izjavami dobimo, če smo na otoku vitezov in oprod in srečamo 5 otočanov, vsak med njimi pa ima lahko cekin. Če ima cekin natanko eden pa lahko število atomarnih izjav zmanjšamo za 1. Če štirje nimajo cekina, ga ima peti. Če ga ima peti, ga nima nobeden od prvih štirih.

Pravila za resničnostne tabele.

(1) A	(2) B	(1) A∧B	(2) A∨B	(3) A→B	(4) A↔B
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T
		(1)	(2)	(3)	(4)

Pravila semantičnih dreves.

$A\wedge B$	$\neg(A\wedge B)$	$A\vee B$	$\neg(A\vee B)$	$A\rightarrow B$	$\neg(A\rightarrow B)$	$A\Leftrightarrow B$	$\neg(A\Leftrightarrow B)$
A	$\neg A \mid \neg B$	A | B	$\neg A$	$\neg A \mid B$	A	A | $\neg A$	A | $\neg A$
B			$\neg B$		$\neg B$	B | $\neg B$	$\neg B$| B

Zgornja pravila (razen za ekvivalenco) lahko razdelimo na pravila podaljševanja in pravila razcepa. K tem pravilom moramo dodati še odpravo dvojne negacije: $\neg\neg A\rightarrow A$.
Zdaj lahko podamo algoritem za izgradnjo semantičnega drevesa za dano množico izjav:
1. Zapišemo dane izjave drugo pod drugo.
2. Izberemo eno od še ne izbranih izjav v drevesu in uporabimo ustrezno pravilo na vseh odprtih vejah drevesa pod to izjavo.
3. Veje, na katerih nastopa kakšna izjava in njena negacija, označimo kot zaprte (protislovne).
4. Če imamo še kakšno odprto vejo in vsaj eno še ne izbrano izjavo, se vrnemo na točko 2.
5. Ko nimamo več neizbranih izjav, se lahko zgodi: a) Vse veje so zaprte. Tedaj je prvotna množica izjav protislovna. b) Če pa imamo odprto vejo, potem lahko najdemo takšno prireditev, da bodo vse izjave na veji resnične. Če na veji nastopa enostavna izjava A, dobi vrednost T, če nastopa $\neg A$, pa F.

Ko izbiramo izjavo, dajemo prednost izjavam, ki drevo podaljšujejo. Zakaj se drevo širi, če imamo v izjavah ekvivalenco?

Oglejmo si zdaj nekaj preprostih nalog iz knjige Poznate naslov te knjige?
Na otoku vitezov in oprod vitezi vedno govorijo resnico, oprode pa vedno neresnico. Če torej neki otočan (Andrej) da izjavo X in označimo A = Andrej je vitez, potem vemo, da velja $A\Leftrightarrow X$. To je, če je otočan vitez, je X resnica, če je X resnica, je otočan vitez.

1. Srečamo otočana (recimo jima A in B). A izjavi, da sta on in B oprodi. Kaj lahko sklepamo?
Vemo: $A\Leftrightarrow(\neg A\wedge\neg B)$. Zgradimo drevo.

$A\Leftrightarrow(\neg A\wedge\neg B)$
$A \mid \neg A$
$\neg A\wedge\neg B \mid \neg(\neg A\wedge\neg B)$
$\neg A \mid \neg\neg A\vee\neg\neg B$
$\neg B \mid A\vee B$
X | A | B
X

Edina odprta veja vsebuje $\neg A$ (A je oproda) in B (B je vitez).

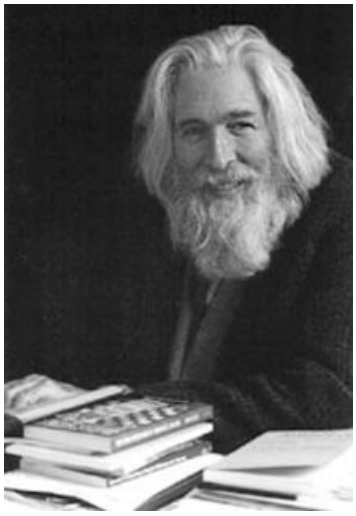
2. A je dejal: "Če sem jaz vitez, potem je tudi B vitez.".
Tokrat je drevo takole.

$A\Leftrightarrow A\rightarrow B$
$A \mid \neg A$
$A\rightarrow B \mid \neg(A\rightarrow B)$
$\neg A \mid B \mid A$
X $\neg B$
X

Na edini odprti veji imamo A in B (oba sta viteza).

Raymond Smullyan (25.5.1919-6.2.2017)

Letos 6. februarja je umrl znameniti sestavljavec logičnih ugank, ameriški logik Raymond Smullyan. Sprva mu je bila namenjena kariera pianista, vendar mu je to preprečila neka bolezen. Pred 2. svetovno vojno se je v glavnem preživljal kot čarovnik. Doktoriral je leta 1959 pri Alonzu Churchu na Univerzi Princeton. Nato je poučeval na različnih ameriških univerzah. Leta 1978 je izšla njegova prva knjiga logičnih ugank »What is the name of this book?«
Pozneje je skoraj vsako leto napisal kakšno knjigo. V slovenskem prevodu smo dobili najprej Alico v deželi ugank, za prevod katere je prevajalka Gitica Jakopin prejela Levstikovo nagrado.





Slovenski prevodi Smullyanov knjig so izhajali pri DZS v zbirki Z logiko v leto 2000. Zaveza, da bodo tekmovanja iz logike pripomogla k boljši prodaji knjig te zbirke, ni preprečila ukinitve te zbirke kmalu po osamosvojitvi Slovenije. Še štirje prevodi in nova izdaja knjige “Kako je naslov tej knjigi”, so izšli v zbirki Univerza za 21. stoletje pri založbi Logika d.o.o.
V tej zbirki naj bi izhajali prevodi študentov pod mentorstvom univerzitetnih profesorjev kot diplomska dela (z obširnim uvodnikom). Predlog je dobil en vljuden negativen odgovor, češ da zadeva ni v skladu s tradicijo diplomskih nalog in nekaj bolj neprimernih komentarjev., kar pa ni presenetljivo, če vemo, da so proti logiki kot izbirnem predmetu nastopili nekateri “znani” pedagogi in celo neki akademik.

Referenca:

Raymond Smullyan, https://en.wikipedia.org/wiki/Raymond_Smullyan

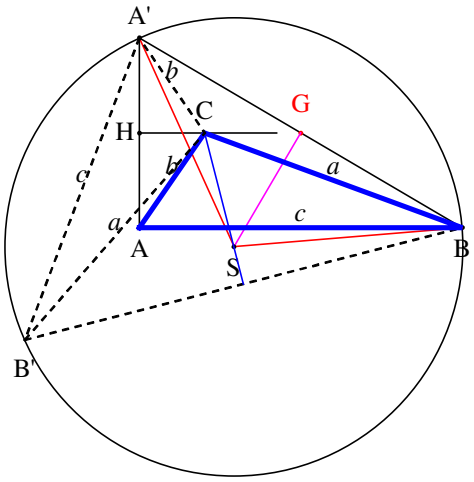
Profesor Josip Plemelj (1873-1967)

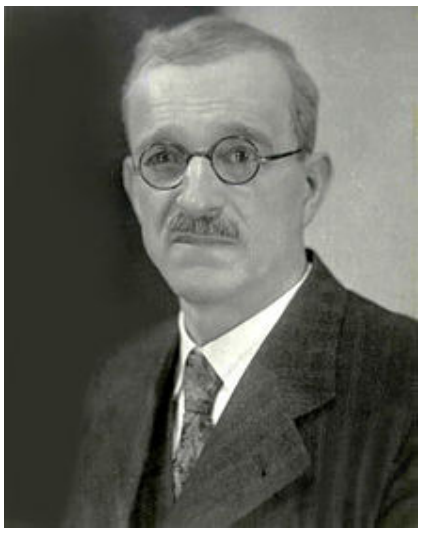
Letos na 22. maj mineva 50 let od smrti največjega slovenskega matematika in prvega rektorja Univerze v Ljubljani, Josipa Plemlja [fotografija je vzeta iz 5]. Že kot učenec 4. razreda klasične gimnazije (danes 9. razred OŠ) je obvladal vso takratno gimnazijsko matematiko. Zato se ne čudimo, da je v 5. razredu uspel rešiti konstrukcijsko nalogo s pomočjo trigonometrije iz 7. razreda, ki jo je potem predelal v geometrijsko konstrukcijo za 5. razred. Naloga se glasi:

Konstruiraj trikotnik ABC, če je dana dolžina osnovnice |AB|=*c*, v, dolžina višine iz C na osnovnico AB in razlika kotov $\delta=\alpha-\beta$ pri A in B.

Plemljeva gimnazijska rešitev je bila nekoliko zapletena in je zelo presenetila njegovega profesorja Vincenca Borštnerja, ki je imel v zbirki nalog enostavnejšo rešitev. Pozneje je Plemelj uvidel, da njegovo razmišljanje privede do enostavnejše rešitve. Ker se je večkrat spomnil te naloge, je vsega našel 9 različnih rešitev, poleg dveh na osnovi nalog iz zbirk.

Plemljeva gimnazijska konstrukcija.

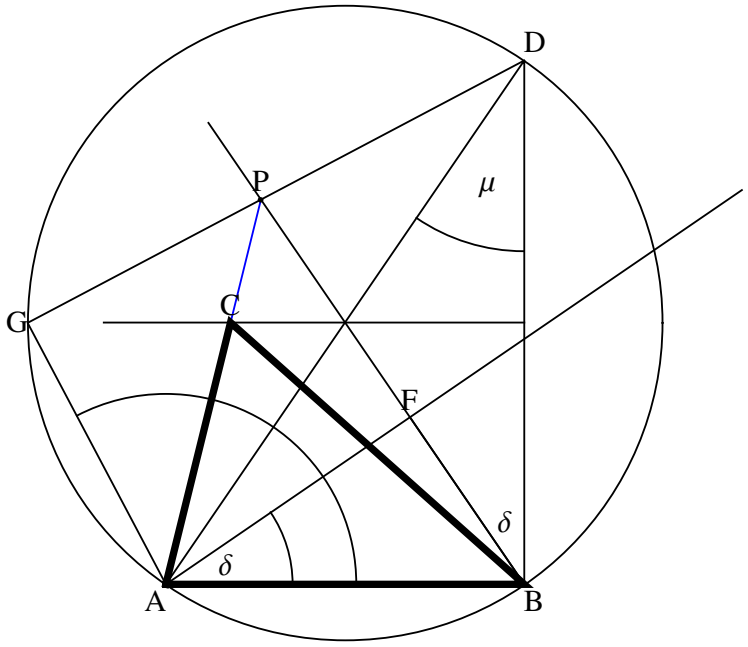




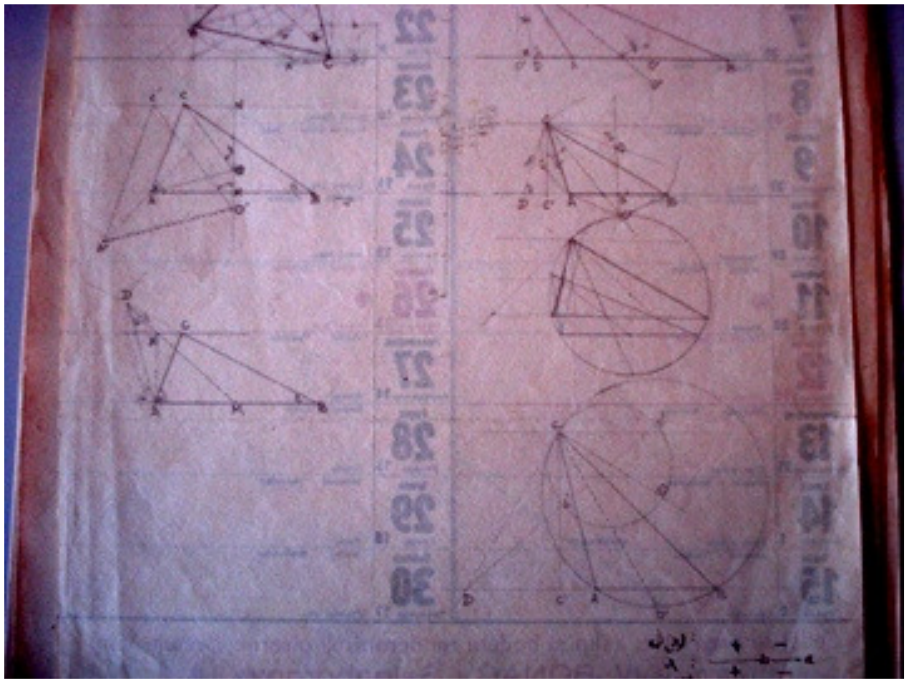
Nariši daljico AB dolžine *c*. Na daljico AB nariši pravokotnico AA' dolžine 2 *v*, s središčem H. Konstruiraj tako krožnico, da se hipotenuza A'B vidi pod kotom $\pi/2 - \epsilon$ ter poišči točko B' na njej, ki je oddaljena za *c* od točke A'. Točka C je presek premice, ki je vzporedna daljici AB in in gre skozi H ter simetrale daljice BB'.
Dokaz pravilnosti. Trikotnik A'B'C je skladen trikotniku ABC (SSS izrek: imata enake stranice). Kot A'B'B je enak $\pi/2-\delta$, kot CB'B pa je enak $\pi/2-\delta-\beta$. Kot B'CS je zadnjemu komplementaren, torej je enak $\delta+\beta$ in je enak kotu SCB. Kot A'CB (>π) je enak $\gamma+2\delta+2\beta=2\alpha+\gamma$, $\delta+\beta=\alpha$. Torej $\delta=\alpha-\beta$.

V članku [1, str. 190], ki je govor J. Plemlja na 1. kongresu zveze jugoslovanskih društev matematikov, fizikov in astronomov na Bledu l. 1949, Plemelj pojasni, kako je prišel do te rešitve: Z višino iz oglišča C na osnovnico AB razpade ta na dva dela $v \cot\alpha$ in $v \cot\beta$ in je tedaj $v(\cot\alpha + \cot\beta) = c$ ali $v \sin(\alpha+\beta) = c \sin\alpha \sin\beta$. To se da pisati

$2v \sin(\alpha + \beta) = c[\cos(\alpha - \beta) - \cos(\alpha + \beta)]$.
Ker je $\alpha + \beta = \pi - \gamma$, se glasi ta enačba
$2\ v \sin\gamma - c \cos\gamma = \cos(\alpha - \beta)$. (Ta enačba se da privesti v kvadratno enačbo za $\sin\gamma$, kar že omogoča konstrukcijo).
Če vpeljemo pomožini kot μ tako da je $2\ v = m \cos\mu$, $c = m \sin\mu$ (m je hipotenuza pravokotnega trikotnika s katetama $2v$ in c).
Dobimo enačbo:
$m \sin(\gamma-\mu)= c \cos(\alpha - \beta)$. Na to enačbo je Plemelj gledal kot na sinusni izrek in prišel do zgornje rešitve.
Zadnjo enačbo pa lahko pišemo $\sin(\gamma-\mu)= \sin\mu \cos(\alpha - \beta)$.
Ta nam omogoča konstrukcijo kota γ, ki je Plemelj v članku ne daje in ni omenjena niti v [2] niti v [3], niti je nismo našli (Nada in Marko Razpet ter Izidor Hafner) v Arhivu RS [5].



Naj bo $|AB|=c$, $|BD|=2v$ in pravokotno na AB, $\delta = \alpha - \beta$, $m=\sqrt{((2v)^2+c^2}$, $\mu=\arctan(c/(2v))$. Potem je $c/m=\sin(\mu)$.
Vzemimo krog z premerom AD enakim m. $|AF|$ je $\sin(\mu) \cos(\alpha - \beta)m$. Naj bo G takšna točka na krožnici, da je $|AG|=|AF|$. Potem je kot GDA enak $\arcsin(\gamma-\mu)$. Torej je kot GDB enak γ, kot GAB pa je $\alpha + \beta$.
Točka C je presečišče premice simetrale daljice GF in vzporednice z AB na višini v.
Ker je $|AF|=|AG|$ in sta kota pri G in F prava, je trikotnik AFP skladen trikotniku AGP (z nasprotno orientacijo). Torej je kot CBA enak kot CAB – kot δ.



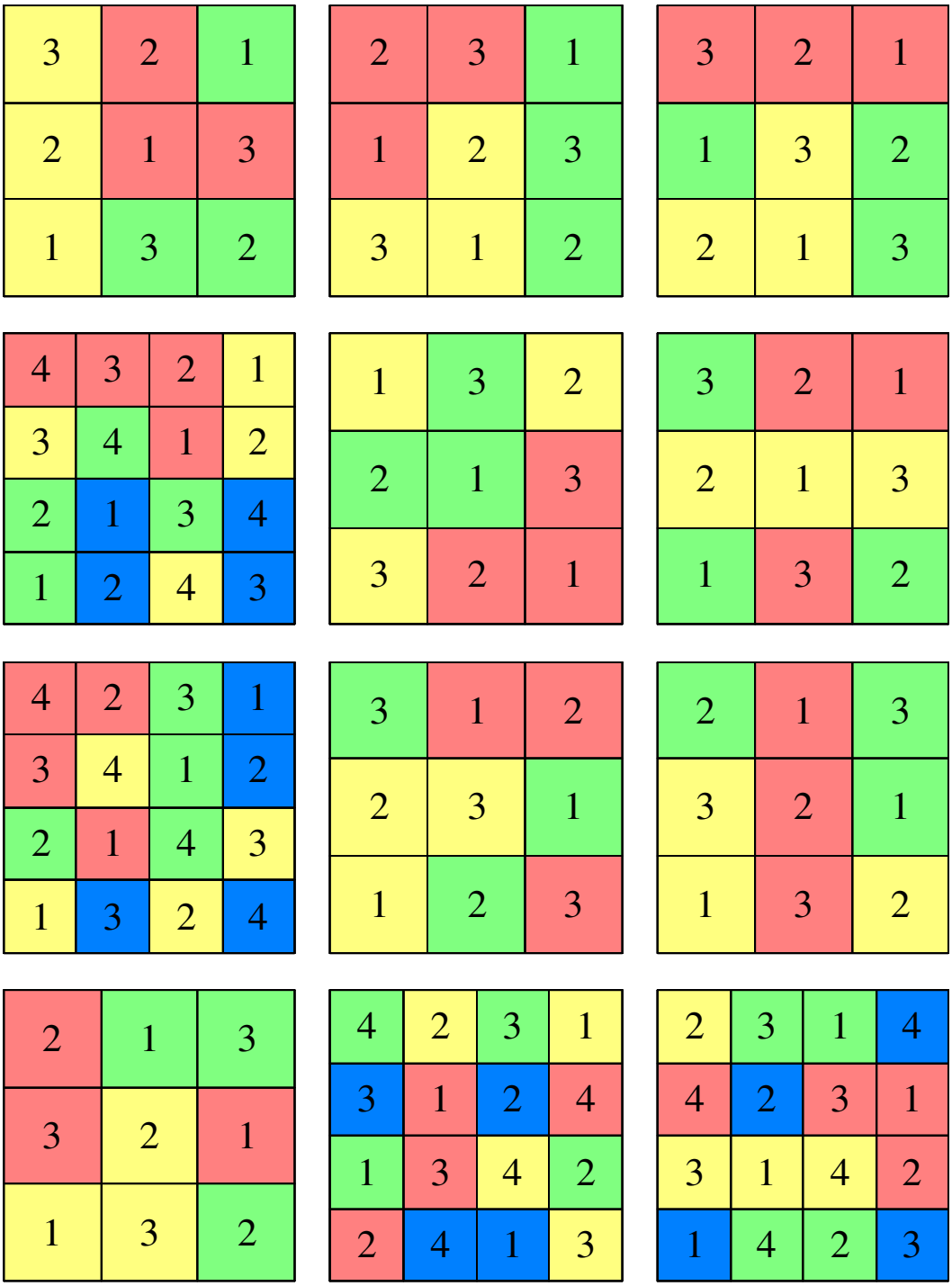
Zgoraj je fotografija zadnje strani koledarja za november 1939, kjer najdemo 3 rešitve Plemljeve naloge [5].

Reference:
[1] Josip Plemelj, Iz mojega življenja in dela, Obzornik mat. fiz. 39 (1992) 6, str. 188-192
[2] Dušan Stanislav Modic, Trikotniki, založba Math, 2009, str. 93.
[3] Proteus, 12, št. 4-5/166, 7/243, (/285, 9/323.
[4] Ivan Pucelj, Plemljev trikotnik in negibne točke transformacij, Obzornik mat. fiz. 62(2015) 1, str. 12-14.
[5] Plemljev fond, Arhiv Republike Slovenije (SI AS 2012), PE19, Škatla 19 (rokopisi).
[6] Josip Plemelj, https://sl.wikipedia.org/wiki/Josip_Plemelj
[7] Izidor Hafner
"The Plemelj Construction of a Triangle"
http://demonstrations.wolfram.com/ThePlemeljConstructionOfATriangle/
Wolfram Demonstrations Project
Published: May 22, 2017

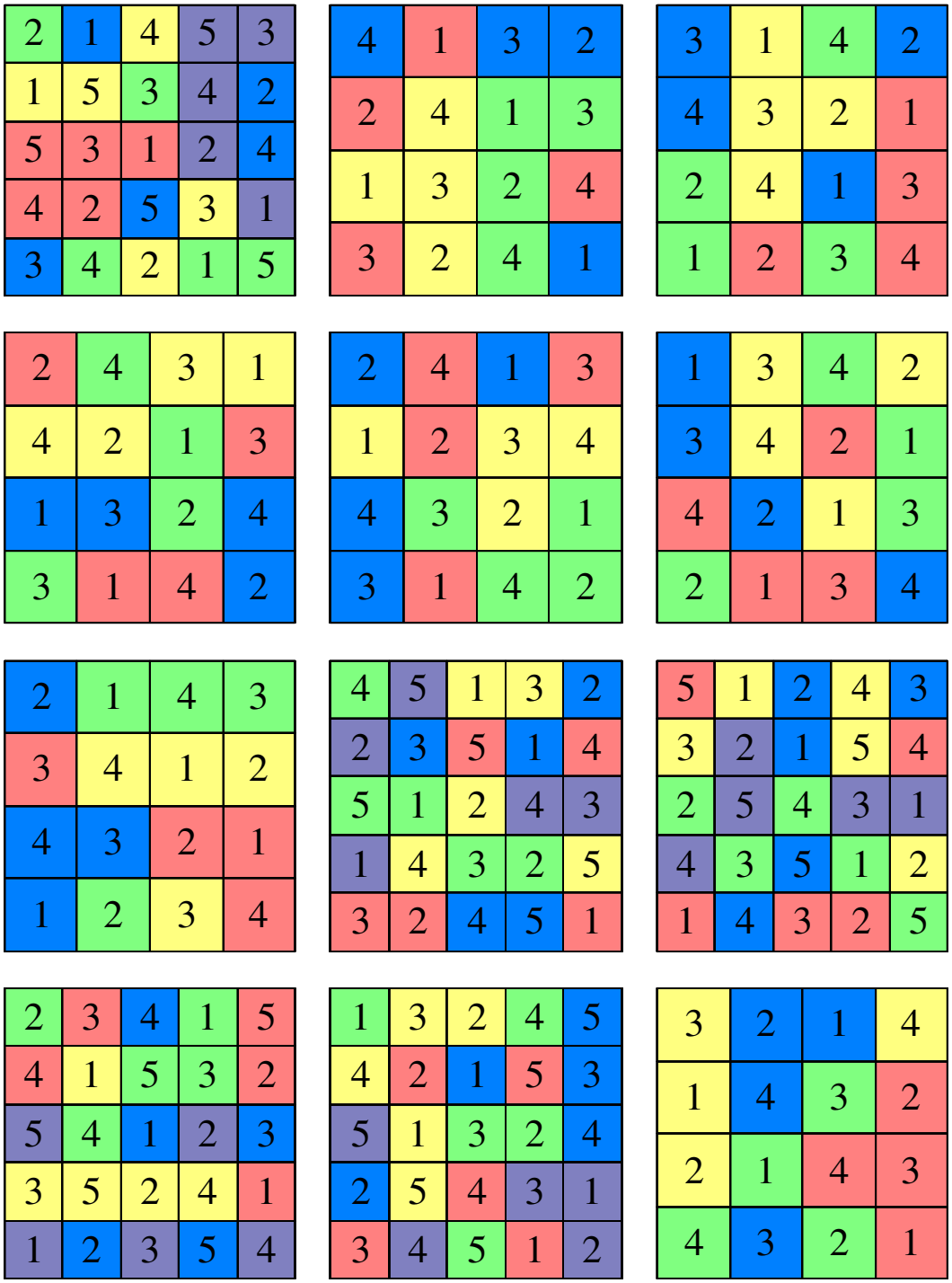
Rešitve

Barvni sudoku

1.



2.


2 1 4 5 3
1 5 3 4 2
5 3 1 2 4
4 2 5 3 1
3 4 2 1 5
4 1 3 2
2 4 1 3
1 3 2 4
3 2 4 1
3 1 4 2
4 3 2 1
2 4 1 3
1 3 4 2
2 4 3 1
4 2 1 3
1 3 2 4
3 1 4 2
2 4 1 3
1 2 3 4
4 3 2 1
3 1 4 2
1 3 4 2
3 4 2 1
4 2 1 3
2 1 3 4
2 1 4 3
3 4 1 2
4 3 2 1
1 2 3 4
4 5 1 3 2
2 3 5 1 4
5 1 2 4 3
1 4 3 2 5
3 2 4 5 1
5 1 2 4 3
3 2 1 5 4
2 5 4 3 1
4 3 5 1 2
1 4 3 2 5
2 3 4 1 5
4 1 5 3 2
5 4 1 2 3
3 5 2 4 1
1 2 3 5 4
1 3 2 4 5
4 2 1 5 3
5 1 3 2 4
2 5 4 3 1
3 4 5 1 2
3 2 1 4
1 4 3 2
2 1 4 3
4 3 2 1

Latinski kvadrati

E	D	C	A	B
C	E	A	B	D
B	C	E	D	A
D	A	B	E	C
A	B	D	C	E

D	E	B	C	A
E	C	D	A	B
A	D	C	B	E
B	A	E	D	C
C	B	A	E	D

B	E	D	C	A
E	B	C	A	D
D	A	E	B	C
C	D	A	E	B
A	C	B	D	E

B	A	D	C
A	B	C	D
C	D	A	B
D	C	B	A

B	A	E	D	C
A	D	B	C	E
E	C	A	B	D
D	B	C	E	A
C	E	D	A	B

C	D	A	B	E
D	C	E	A	B
E	A	B	D	C
A	B	C	E	D
B	E	D	C	A

D	B	C	A
C	A	B	D
B	D	A	C
A	C	D	B

B	A	D	C
C	B	A	D
A	D	C	B
D	C	B	A

D	B	A	C	E
B	E	D	A	C
A	D	C	E	B
E	C	B	D	A
C	A	E	B	D

C	B	D	A
B	A	C	D
A	D	B	C
D	C	A	B

A	C	D	B
C	D	B	A
D	B	A	C
B	A	C	D

D	E	C	B	A
A	B	D	E	C
B	A	E	C	D
E	C	A	D	B
C	D	B	A	E

Sudoku s črkami

4 B	1 B	3 B	2 D
1 D	2 C	4 A	3 A
2 B	3 D	1 A	4 D
3 C	4 C	2 A	1 C

1 B	4 C	2 C	3 B
3 C	1 D	4 A	2 D
2 A	3 D	1 C	4 B
4 D	2 B	3 A	1 A

1 D	3 D	4 B	2 C
4 A	1 C	2 B	3 A
2 D	4 D	3 C	1 A
3 B	2 A	1 B	4 C

4 B	1 D	2 D	3 D
2 C	4 A	3 A	1 A
1 B	3 B	4 D	2 B
3 C	2 A	1 C	4 C

1 C	3 B	2 B	4 B
2 D	1 D	4 C	3 A
3 C	4 A	1 B	2 A
4 D	2 C	3 D	1 A

2 C	3 A	1 A	4 A
4 D	2 B	3 C	1 B
1 C	4 C	2 A	3 B
3 D	1 D	4 B	2 D

1 B	3 D	2 A	4 D
4 A	1 C	3 C	2 D
2 C	4 B	1 D	3 A
3 B	2 B	4 C	1 A

3 C	1 D	4 D	2 B
4 A	2 D	3 A	1 C
1 A	4 C	2 C	3 D
2 A	3 B	1 B	4 B

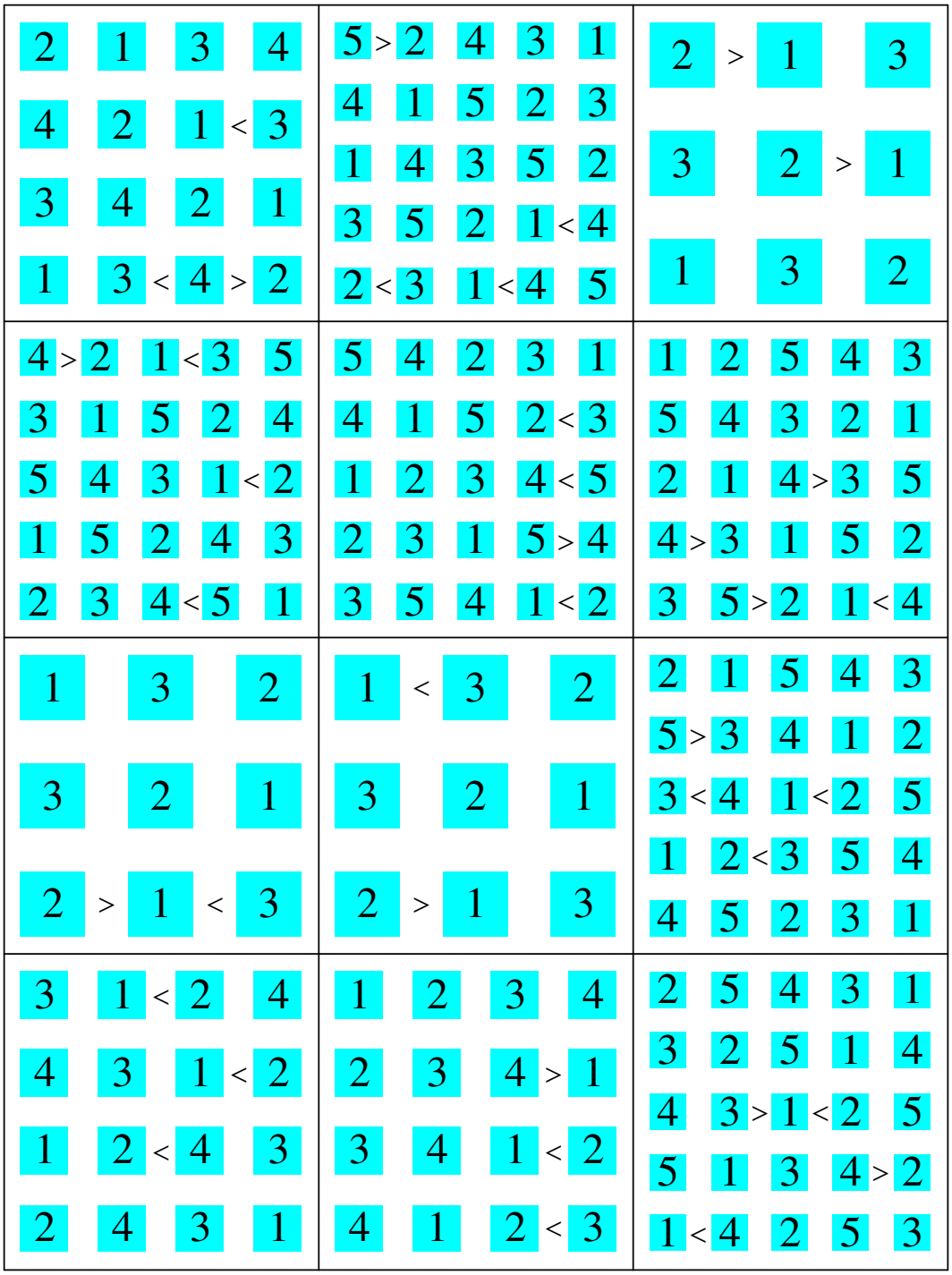
2 D	1 D	3 B	4 D
4 B	2 A	1 A	3 A
3 C	4 A	2 C	1 C
1 B	3 D	4 C	2 B

4 A	3 D	2 C	1 A
1 C	2 A	3 B	4 D
3 C	4 C	1 D	2 B
2 D	1 B	4 B	3 A

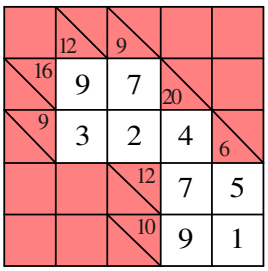
4 C	2 C	1 C	3 D
1 B	3 C	4 D	2 B
2 D	4 A	3 B	1 A
3 A	1 D	2 A	4 B

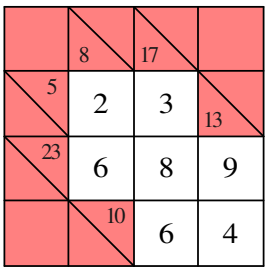
4 C	2 D	3 A	1 C
1 A	3 B	2 B	4 A
3 D	1 B	4 B	2 A
2 C	4 D	1 D	3 C

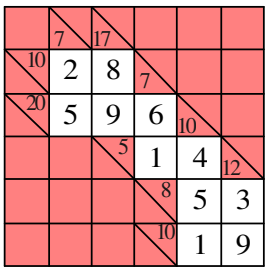
Futošiki

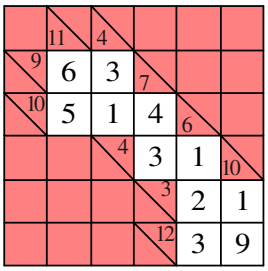

2 1 3 4
4 2 1 < 3
3 4 2 1
1 3 < 4 > 2
5 > 2 4 3 1
4 1 5 2 3
1 4 3 5 2
3 5 2 1 < 4
2 < 3 1 < 4 5
2 > 1 3
3 2 > 1
1 3 2
4 > 2 1 < 3 5
3 1 5 2 4
5 4 3 1 < 2
1 5 2 4 3
2 3 4 < 5 1
5 4 2 3 1
4 1 5 2 < 3
1 2 3 4 < 5
2 3 1 5 > 4
3 5 4 1 < 2
1 2 5 4 3
5 4 3 2 1
2 1 4 > 3 5
4 > 3 1 5 2
3 5 > 2 1 < 4
1 3 2
3 2 1
2 > 1 < 3
1 < 3 2
3 2 1
2 > 1 3
2 1 5 4 3
5 > 3 4 1 2
3 < 4 1 < 2 5
1 2 < 3 5 4
4 5 2 3 1
3 1 < 2 4
4 3 1 < 2
1 2 < 4 3
2 4 3 1
1 2 3 4
2 3 4 > 1
3 4 1 < 2
4 1 2 < 3
2 5 4 3 1
3 2 5 1 4
4 3 > 1 < 2 5
5 1 3 4 > 2
1 < 4 2 5 3

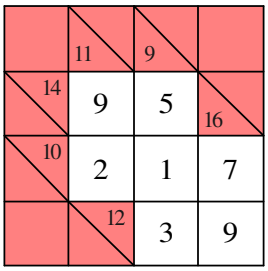
Križne vsote

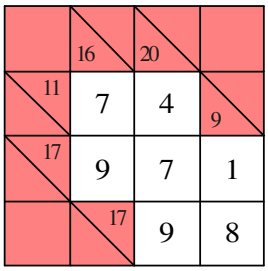


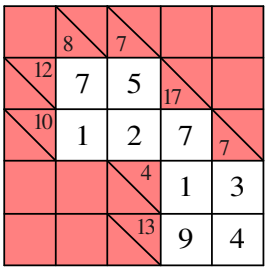


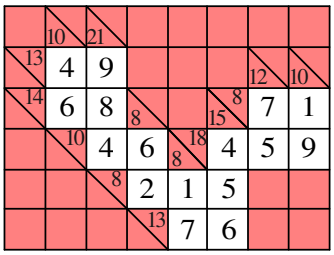


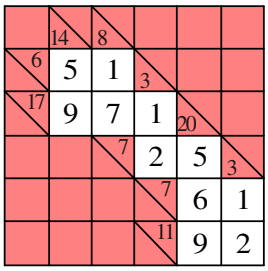


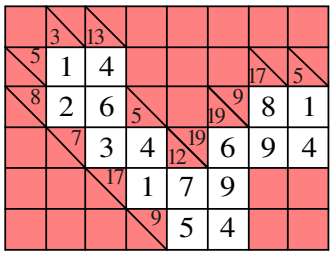


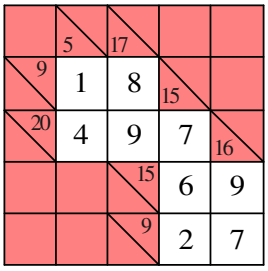


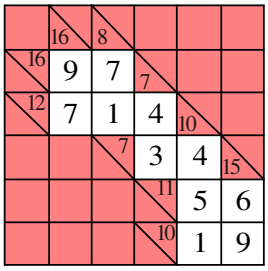




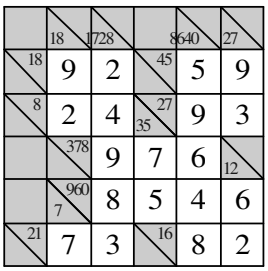


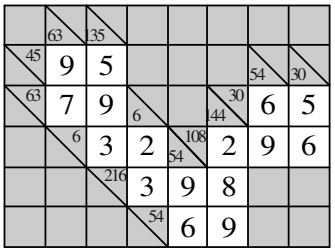


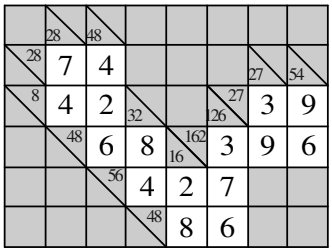


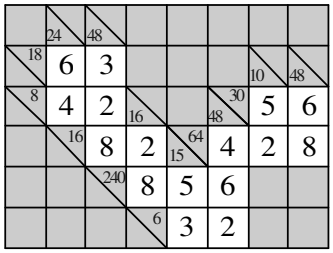


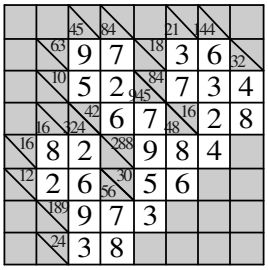
Križni produkti

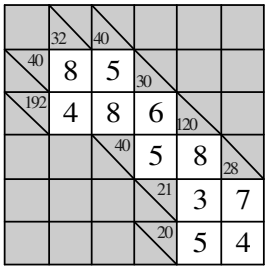


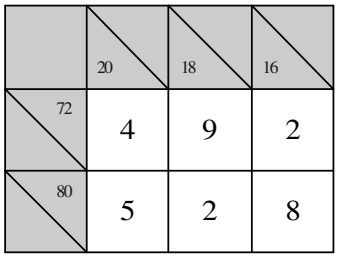


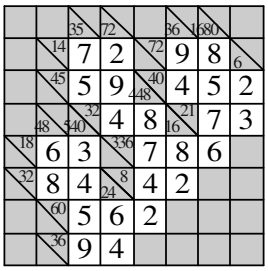


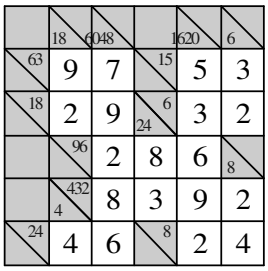


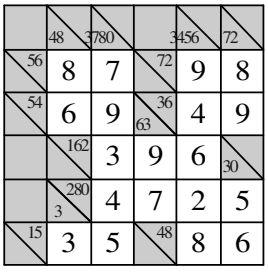


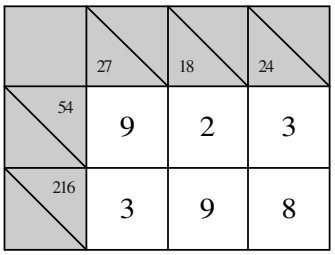


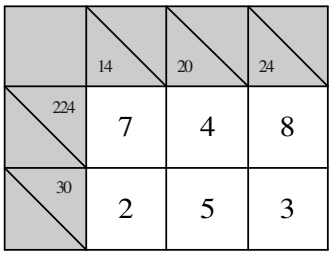




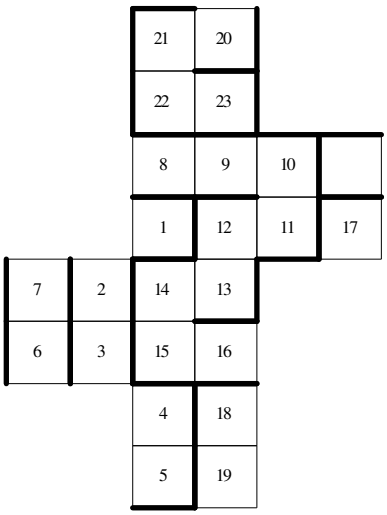


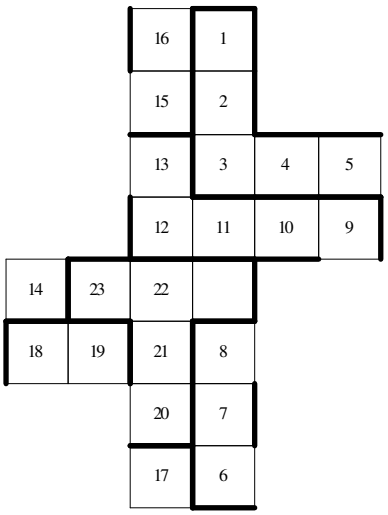


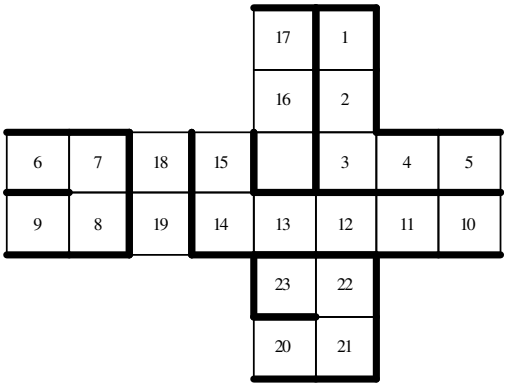


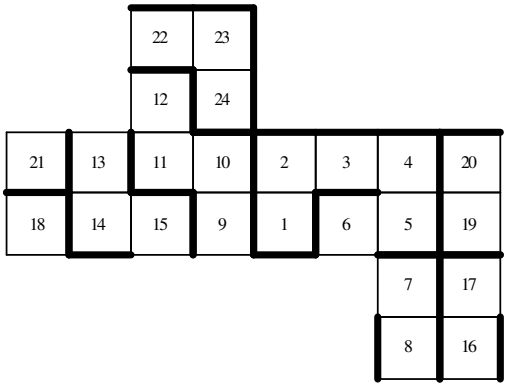


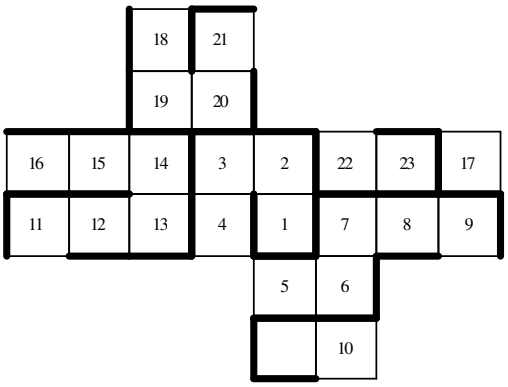
Labirint na kocki

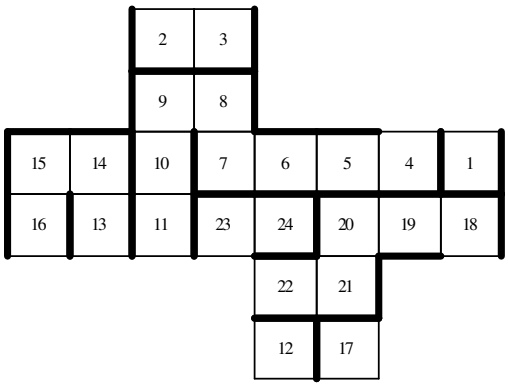




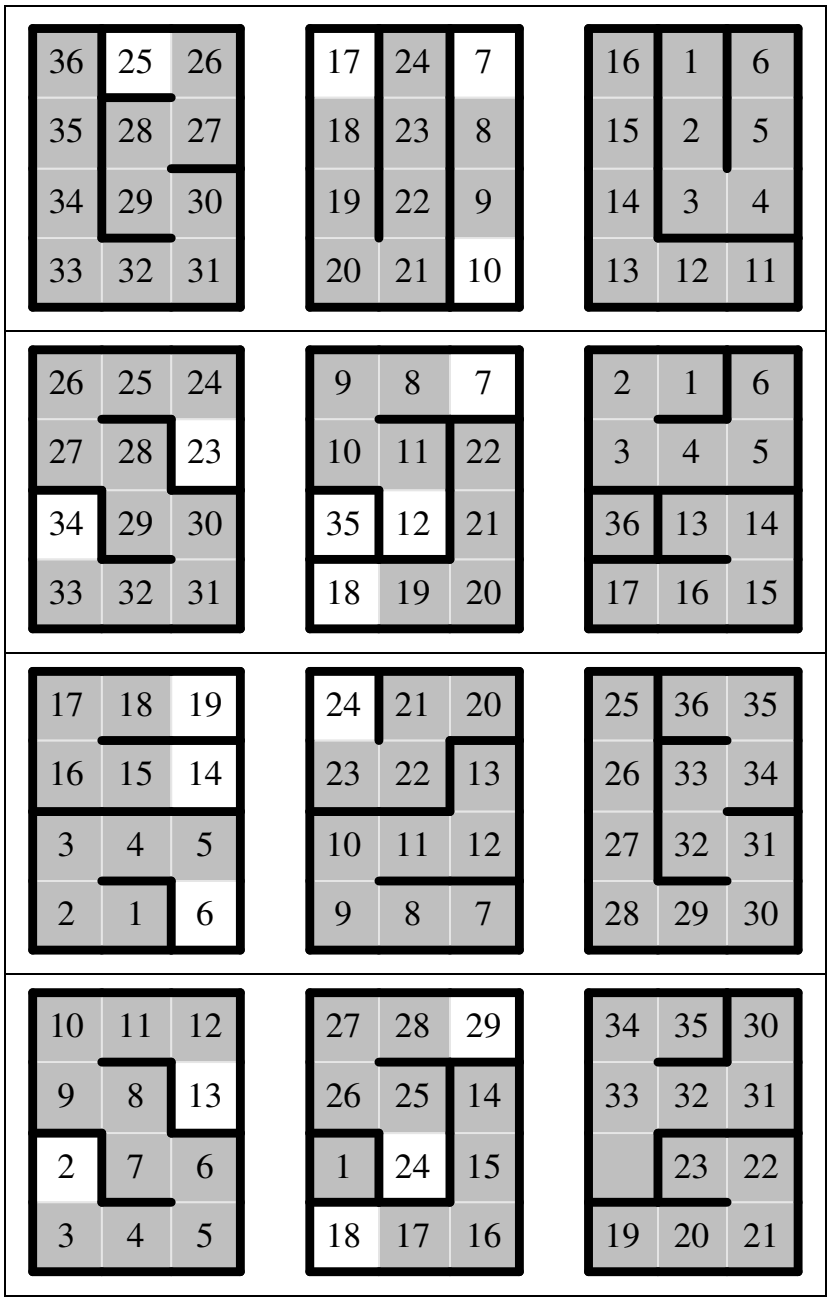








Labirint v kvadru



Odstranjene kockice

70	43	117
61	76	116
114	70	63
93	71	95

Izdaja: Založniško podjetje **LOGIKA d.o.o., Svetčeva pot 11, 1241 Kamnik**. Poslovni račun pri NLB: 02312-0016592829. Davčna številka: SI56917309. Podjetje je zavezanec za DDV po zakonu o DDV.
Za izdajatelja*: Izidor Hafner*.
E-mail: **logika@siol.net**.
Spletna stran: **http://www.logika.si**.
Revija *Logika & razvedrilna matematika* je vpisana v register medijev pri Ministrstvu za kulturo pod številko 759. Strokovni pokrovitelj: *Inštitut za matematiko, fiziko in mehaniko - oddelek za teoretično računalništvo*.

Glavni in odgovorni urednik: *dr. Izidor Hafner* (**http://mat03.fe.uni-lj.si/html/people/izidor/homepage/**)
Člana časopisnega sveta: *prof. dr. Tomaž Pisanski in Darjo Felda, prof.*
Recenzent: *Vilko Domajnko, prof.*
Sodelavci: *mag. Urša Demšar, dr. Gregor Dolinar, Monika Kavalir, dr. Meta Lah, Boštjan Kuzman,Teja Oblak, Hiacinta Pintar, Maja Pohar, mag. Katka Šenk in dr. Aleš Vavpetič.*
Oblikovanje: *Ana Hafner*
Jezikovni pregled: *Besana*

Za objavljene prispevke ne plačujemo honorarjev.

© 2017 LOGIKA d.o.o.
ISSN 2350-532X
LOGIKA & RAZVEDRILNA MATEMATIKA, letnik XXVI, št. 4 od 4, 2016/2017
Elektronska izdaja. Cena revije: 0 €.