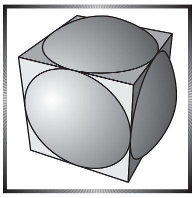


Az A pontversenyben kitűzött nehezebb feladatok (821–823.)

A. 821. *a*) Létezik-e olyan $f: \mathbb{N}^2 \to \mathbb{N}$ függvény, melyre minden $g: \mathbb{N} \to \mathbb{N}$ függvény és m pozitív egész esetén létezik $n \in \mathbb{N}$, melyre a $\{k \in \mathbb{N}: f(n,k) = g(k)\}$ halmaz elemszáma legalább m?

b) Létezik-e olyan $f: \mathbb{N}^2 \to \mathbb{N}$ függvény, melyre minden $g: \mathbb{N} \to \mathbb{N}$ függvény esetén létezik $n \in \mathbb{N}$, melyre a $\{k \in \mathbb{N}: f(n,k) = g(k)\}$ halmaz elemszáma végtelen?

A. 822. Léteznek-e p, q, r racionális számok, melyekre $p+q+r=0$ és $pqr=1$?

Javasolta: *Weisz Máté* (Cambridge)

A. 823. Legyen n pozitív egész, és tekintsük az $S_n = \{(x,y,z): 1 \leqslant x \leqslant n, 1 \leqslant y \leqslant n,\ 1 \leqslant z \leqslant n,\ x,y,z \in \mathbb{N}\}$ kockarácsot. Létezik-e olyan n pozitív egész, melyre ki lehet választani S_n elemei közül több, mint $n\sqrt{n}$-t úgy, hogy bármely két kiválasztott rácspont közül az egyiknek legalább két koordinátája szigorúan nagyobb legyen, mint a másik megfelelő két koordinátája?

Javasolta: *Csóka Endre* (Budapest)

Beküldési határidő: 2022. április 10.
Elektronikus munkafüzet: `https://www.komal.hu/munkafuzet`

❄

A matematikai logika logikusabb, mint gondolnánk II.

Ha alaposan megfigyeljük a cikk első részében kapott ábrát, egy további nyereséget is elkönyvelhetünk: ahogy az algebrában, úgy a logikában is vannak azonosságok. Ezek egyike-másika közismert, ahogy az algebrában is vannak nevezetes azonosságok, például $(a+b)^2 = a^2 + 2ab + b^2$. A logikában ilyen a $\neg(A \wedge B) = \neg A \vee \neg B$, vagy $(A \to B) \wedge (B \to A) = A \Leftrightarrow B$.

Ez utóbbi kapcsán felmerül, hogy miként lehet összetettebb logikai kifejezéseket egyszerűbb alakra hozni. Nos, ez egyáltalán nem bonyolult. A könnyebb megértés kedvéért foglaljuk táblázatba az öt alapművelet eredményének szabályát. Elég csak az egyik állapotot megjegyezni, a többi esetben mindig a tagadása lesz az eredmény: az *és* művelet igaz, ha mind a két részállítás igaz, a *vagy* művelet hamis, ha mind a két részállítás hamis, a *kizáró vagy* művelet igaz, ha a két részállítás logikai értéke különböző, az *azonosság* művelet igaz, ha a két részállítás logikai értéke azonos, végül a *következtetés* művelet hamis, ha az, amiről következtetünk igaz, de az, amire következtetünk hamis. Összefoglalva:

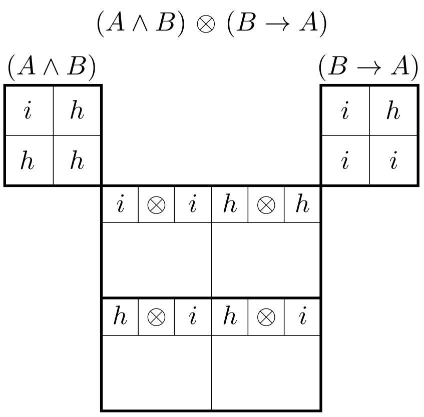
a művelet	az egyik	a másik	az eredmény
	részállítás		
És ($A \wedge B$)	i	i	i
Vagy ($A \vee B$)	h	h	h
Kizáró vagy ($A \otimes B$)	x	y	i
Azonosság ($A \Leftrightarrow B$)	x	x	i
Következtetés ($A \rightarrow B$)	i	h	h

Végezetül a tagadások okozta cseréket is rendszerezzük:

$A \wedge B$	$\neg A \wedge B$	$A \wedge \neg B$	$\neg A \wedge \neg B$	$\neg(A \wedge B)$
$\begin{array}{cc\|cc} \multicolumn{2}{c\|}{A \wedge B} & \multicolumn{2}{c}{A} \\ & & i & h \\ \hline B & i & i & h \\ & h & h & h \end{array}$	$\begin{array}{cc\|cc} \multicolumn{2}{c\|}{\neg A \wedge B} & \multicolumn{2}{c}{A} \\ & & i & h \\ \hline B & i & h & i \\ & h & h & h \end{array}$	$\begin{array}{cc\|cc} \multicolumn{2}{c\|}{A \wedge \neg B} & \multicolumn{2}{c}{A} \\ & & i & h \\ \hline B & i & h & h \\ & h & i & h \end{array}$	$\begin{array}{cc\|cc} \multicolumn{2}{c\|}{\neg A \wedge \neg B} & \multicolumn{2}{c}{A} \\ & & i & h \\ \hline B & i & h & h \\ & h & h & i \end{array}$	$\begin{array}{cc\|cc} \multicolumn{2}{c\|}{\neg(A \wedge B)} & \multicolumn{2}{c}{A} \\ & & i & h \\ \hline B & i & h & i \\ & h & i & i \end{array}$
$\begin{array}{c\|c} i & h \\ \hline h & h \end{array}$	$\begin{array}{c\|c} h & i \\ \hline h & h \end{array}$	$\begin{array}{c\|c} h & h \\ \hline i & h \end{array}$	$\begin{array}{c\|c} h & h \\ \hline h & i \end{array}$	$\begin{array}{c\|c} h & i \\ \hline i & i \end{array}$
$A \wedge B$	$\neg(B \rightarrow A)$	$\neg(A \rightarrow B)$	$\neg(A \vee B)$	$\neg(A \wedge B)$
az eredetihez képest	oszlopcsere (OCS)	sorcsere (SCS)	átlós csere (ÁCS)	betűcsere (BCS)

amit tagadunk	mozgatás
A	oszlopcsere (OCS)
B	sorcsere (SCS)
A és B	átlós csere (ÁCS)
a művelet	betűcsere (BCS)

Az összetett állításoktól a számítógépig



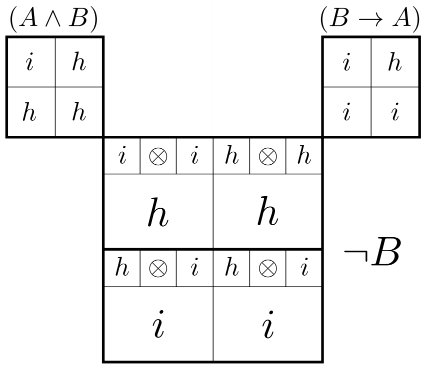
A formalizálás banálisan egyszerű, a szomszédos *ábra* a modell. A bal felső részbe a bal oldalon lévő kifejezés ($A \wedge B$), a jobb felső részbe pedig a jobb oldali ($B \rightarrow A$) logikai sakktáblában található értékei kerüljenek, ezeket a középső mező négy szegmensének bal és jobb felső részébe másoljuk át, középre pedig a két oldal közötti művelet jelét ($\otimes$) illesszük be.

Ezek után szegmensenként külön-külön értékeljük ki a műveleti szabályokban foglaltak alapján: $i \otimes i$ $(= h)$, $h \otimes h$ $(= h)$, $h \otimes i$ $(= i)$ és $h \otimes i$ $(= i)$.

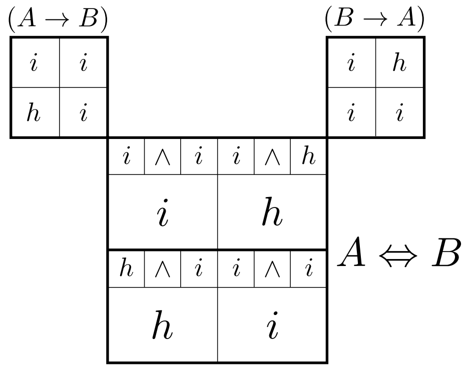
Az eredményt rögzítsük az eddig üres cellákba, majd a logikai sakktáblában keressük ki, hogy ez melyik művelet értéktáblázata ($\neg B$).

Másik magyarázó példaként álljon itt az $(A \to B) \wedge (B \to A) = A \Leftrightarrow B$ nevezetes logikai azonosság igazolása.

$(A \wedge B) \otimes (B \to A)$



$(A \to B) \wedge (B \to A)$



Nézzük meg egy olyan kifejezés kiértékelését, ahol az előbbieket kicsit módosítva jutunk el a megoldáshoz.

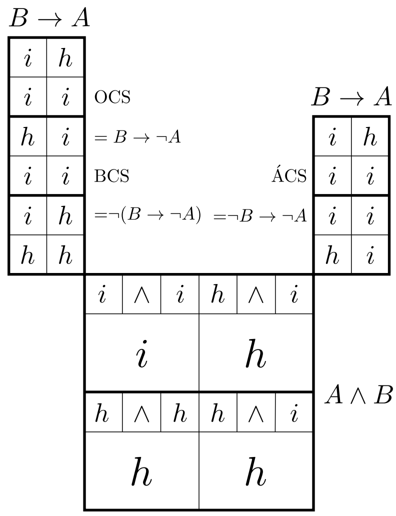
Egyszerűsítsük le a

$$\neg(B \to \neg A) \wedge (\neg B \to \neg A)$$

kifejezést. Elsőre talán ijesztőnek tűnhet, de a tagadásokat könnyen megoldhatjuk cserékkel.

Egyszer érdemes belegondolni, hogy ez mit is jelent. A cikk előző részében az A állítás az volt, hogy *Kedd van,* a B állítás pedig az, hogy *Esik az eső.* Ezeket felhasználva ez a kifejezés így szól: *Nem igaz az, hogy, ha esik az eső, akkor nem kedd van, és ha nem esik az eső, akkor nem kedd van.* Leegyszerűsítve: *Kedd van és esik az eső.*

$\neg(B \to \neg A) \wedge (\neg B \to \neg A)$



A vállalkozó kedvűeknek ajánlok néhány feladatot:

1. $(A \wedge B) \vee (A \otimes B)$.

2. $(A \vee B) \wedge (B \to A)$.

3. $\neg(B \to A) \otimes A$.

4. $(A \Leftrightarrow B) \Leftrightarrow \neg A$.

5. $(A \vee \neg B) \to \neg(\neg A \vee B)$.

6. $\neg\big[A \to (A \otimes B)\big]$.

7. $\big[\neg B \otimes (A \wedge B)\big] \wedge \big[B \to \neg(A \wedge B)\big]$.

Ideje rátérnünk arra, hogy ez az egész miként kapcsolódik a számítógépekhez.

A számítógépek elektromos árammal működnek. És nem csak energiaforrásként használják az elektromos energiát, az információt is elektromosan továbbítják. Feldolgozáskor a leggyorsabb eldöntési technika a nyerő, márpedig az elektromos technológiában a folyik vagy nem folyik áram eldöntése lényegesen gyorsabb és még egyszerűbb is, mint az áram nagyságát megmérni. Ez két állapot megkülönböztetését jelenti, tehát a kétállapotú rendszerek használata gyorsabb és egyszerűbb is.

Ilyen rendszert kettőt is ismerünk, a matematikai logika igaz/hamis rendszere és a kettes számrendszer, ahol két különböző számjegy, a 0 és az 1 létezik. Már az elektronika hajnalán képesek voltak olyan áramköröket készíteni, amelyek modellezik a logikai műveleteket, ezeket *logikai kapuknak* nevezik. A teljesség igénye nélkül említsünk meg néhányat. A *tagadás* megfelelője az egy bemenettel és egy kimenettel rendelkező NOT kapu, a kimenetén folyik áram, ha a bemenetén nem, és a kimeneten nem folyik áram, ha a bemeneten igen. A két bemenettel és egy kimenettel rendelkező AND, OR, XOR kapu pedig az *és,* a *vagy* és a *kizáró vagy* művelet szabálya szerint működik.

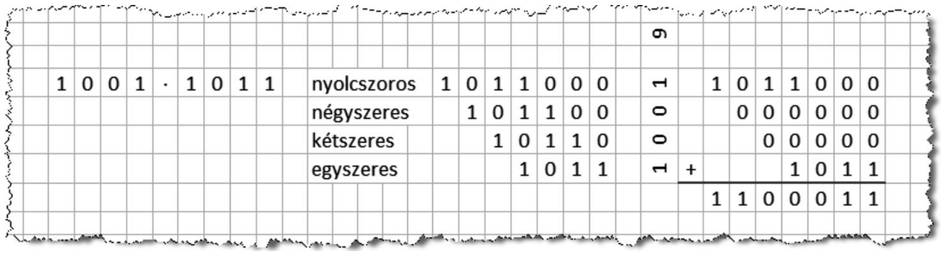
A számítógép neve a számításra és nem a matematikai logikára utal, hiszen akkor talán ítélőgépnek hívnánk. De az állítások kiértékelése a mindennapokban sokkal kisebb szerephez jut, mint a számítások elvégzése. Abba azonban ritkán gondolunk bele, hogy a kettes számrendszerben mennyire egyszerű a műveletek elvégzése, ezért ehhez mondanék néhány adalékot.

A kettes számrendszerben az összeadás és a szorzás szabálya is elég egyszerű egy számjegy esetén, akár egy sorban el is mondható: $0+0=0$, $0+1=1$, $1+0=1$ és $1+1=0$, valamint $0\cdot 0=0$, $0\cdot 1=0$, $1\cdot 0=0$ és $1\cdot 1=1$.

Az algebra szabályai függetlenek a számrendszertől: a számrendszer alapszámának k-adik hatványával szorzás k darab 0 számjegynek a szám végére írását jelenti, így például kettes számrendszerben az $101_{②}(=5_{⑩})$ kétszerese, négyszerese, nyolcszorosa: $1010_{②}$, $10100_{②}$ és $101000_{②}$.

A kettes számrendszerben a szorzás könnyen kiváltható összeadással. Az egyik tényező kettes számrendszerbeli alakja pont azt mutatja meg a különböző helyiértékeken álló 1 számjegyeivel, hogy a másik tényezőnek melyek azok kettőhatványszorosai, amelyeket össze kell adnunk. Nézzük meg a $9 \cdot 11_{(10)}$ szorzást.

$1001 \cdot 1011_{(2)}$. Ebben az $1011_{(2)}$ szám egyszeresét és nyolcszorosát kell összeadni, hiszen $9 = 1 + 8_{(10)}$. Csak három nullát kell a végére biggyeszteni, hogy meglegyen a nyolcszoros, és azt az eredetivel összeadni. $101000_{(2)} + 1011_{(2)}$.



Ez valóban nagy könnyebbség, csak ki kell dolgozni azokat az áramköröket, amelyek kettes számrendszerben elvégzik az összeadást. Vagy mégsem?

Nem bizony.

Elég csak szemrevételezni az alábbi *ábrát,* és feltűnik, hogy az imént említett AND és XOR logikai kapuk lehetővé teszik az összeadást. Tehát nem kell ezeket kidolgozni, hiszen ezek a logikai kapuk már megvannak.

$A \wedge B$		A	
		i	h
B	i	i	h
	h	h	h

$A \otimes B$		A	
		i	h
B	i	h	i
	h	i	h

összeadás: $A+B$		A	
		1	0
B	1	1	0
	0	0	0

átvitel

összeadás: $A+B$		A	
		1	0
B	1	0	1
	0	1	0

eredmény

és	kizáró vagy	bináris összeadás:
$h \wedge h = h$	$h \otimes h = h$	$0 + 0 = 00$
$h \wedge i = h$	$h \otimes i = i$	$0 + 1 = 01$
$i \wedge h = h$	$i \otimes h = i$	$1 + 0 = 01$
$i \wedge i = i$	$i \otimes i = h$	$1 + 1 = 10$

Ha elég sok, elég apró és elég gyors logikai kaput tudunk összedolgozni, akkor a $9 \cdot 11_{(10)}$ szorzásnál lényegesen összetettebb műveleteket is el tudunk végezni.

És a mikroprocesszor icipici logikai kapuk millióit hordozza, és többszáz milliárd műveletre képes másodpercenként, ezért tudnak ezek a gépek többféle, az ember számára bonyolult vagy unalmas munkát kiváltani.

Ezért is érdemes legalább egy főhajtással tisztelegni *George Boole* (1815. november 2.– 1864. december 8.) angol matematikus előtt, aki lerakta a matematikai logika alapjait.

Az arckép forrása: `https://www.sciencephoto.com/media/223560/view/english-mathematician-george-boole`.

Tóth Tamás
Budapest

Informatikából kitűzött feladatok

I. 559. Egy számítógépes játékban két háromfős csapat játszik egymással. A játékosok a korábbi játszmák eredményei alapján pontszámokkal rendelkeznek, melyeket a győzelmeik és vereségeik alapján számít ki a játékprogram. A kapott pontok minden esetben pozitív egész számok.

A számítógépes játékban egy játszmába hat játékos jelentkezik be. Kezdetben az első három bejelentkező játékos az első, a másik három pedig a második csapatba kerül. A program igyekszik a pontszámok alapján egyenlő erősségű csapatokat létrehozni. A csapatok erősségét a játékosok pontszámának összegével adjuk meg. A csapatok elosztását úgy végzi a program, hogy legföljebb egy játékost az egyik csapatból kicserél egy másik játékosra a másik csapatból. A játszmában a 6 játékos úgy alkot két csapatot, hogy legföljebb egy ilyen cserével a két csapat erőssége a lehető legkevésbé térjen el egymástól.

Készítsünk programot, amely a 6 játékos pontszáma alapján megad egy elosztást. Ha több ilyen elosztás lehetséges, akkor bármelyik megadható.

A bemenet egyetlen sorában a játékosok pontszáma szerepel egy-egy szóközzel elválasztva. Az i-edik szám az i-edik játékos pontszáma.

A kimenet egyetlen sorában a csapatok elosztása szerepel: az első három szám az egyik csapat, míg a második három szám a második csapat tagjait jelenti. Amennyiben nem szükséges cserélni, akkor az `1 2 3 4 5 6` számsorozatot kell kiírni szóközzel elválasztva. Ha történt csere, akkor a megfelelő helyen lévő sorszámokat kell cserélni.