Logika gyakorlat – 04
Boole-függvények teljes rendszerei

Recap: Teljes (vagy „adekvát” rendszer)

Boole-függvények egy olyan halmaza, melyekkel az összes többi is kifejezhető.

Itt az „összes többi kifejezhető” azt jelenti, hogy ha H Boole-függvényeknek egy halmaza, akkor

- ha kiindulunk az x_1, x_2, …, x_n változókból, ezek (a „projekciók”) kifejezhetőek;
- és ha $g/k \in H$ egy Boole-függvény, és az $f_1/n, \ldots, f_k/n$ függvények kifejezhetőek, akkor a **kompozíciójuk**, azaz az a $g \circ \langle f_1, \ldots, f_k \rangle/n$ függvény, melyre

  $$g \circ \langle f_1, \ldots, f_k \rangle(a_1, \ldots, a_n) \;=\; g(f_1(a_1, \ldots, a_n), f_2(a_1, \ldots, a_n), \ldots, f_n(a_1, \ldots, a_n)),$$

  szintén kifejezhető,
- más kifejezhető függvény nincs, csak amiket így kapunk,

akkor megkapjuk az összes (n-változós) Boole-függvényt.

Amit előadásról tudunk (pl. a Shannon-expanzióból, vagy mert minden Boole-függvényhez létezik őt indukáló CNF):

Az első teljes rendszerünk

A $\{\neg, \vee, \wedge\}$ rendszer teljes.

1. feladat.

Mutassuk meg, hogy a $\{\neg, \wedge\}$ rendszer is teljes!

2. feladat.

Mutassuk meg, hogy a $\{\neg, \vee\}$ rendszer is teljes!

3. feladat.

Mutassuk meg, hogy a $\{\rightarrow, \neg\}$ rendszer is teljes!

4. feladat.

Mutassuk meg, hogy a $\{\rightarrow, \downarrow\}$ rendszer is teljes!

5. feladat.

A $\mathsf{nand}/2$ függvény: $\mathsf{nand}(1,1) = 0$, minden más inputra 1 az értéke (az éselés negáltja).

Mutassuk meg, hogy $\{\mathsf{nand}\}$ is teljes rendszert alkot!

6. feladat.

A $\mathsf{nor}/2$ függvény: $0 \mathbin{\mathsf{nor}} 0 = 1$, minden másra 0 (negált „vagy”).

Mutassuk meg, hogy $\{\mathsf{nor}\}$ is teljes rendszert alkot!

7. feladat.

Mutassuk meg, hogy a $\{\vee, \wedge\}$ rendszer **nem** teljes!

8. feladat.

Legyen a $\oplus/2$ a XOR függvény: $0 \oplus 0 = 1 \oplus 1 = 0$, $0 \oplus 1 = 1 \oplus 0 = 1$.

Mutassuk meg, hogy a $\{\oplus, \uparrow\}$ rendszer **nem** teljes!

9. feladat.

Mutassuk meg, hogy a $\{\vee, \rightarrow, \wedge, \uparrow, \leftrightarrow\}$ rendszer **nem** teljes!

10. feladat.

Legyen $m/3$ a (háromváltozós) többségi függvény: azt a bitet adja vissza, melyből többet kapott, pl. $m(0,0,1) = 0$, $m(1,0,1) = 1$, $m(0,0,0) = 0$ stb.

Mutassuk meg, hogy az $\{m, \neg, \uparrow\}$ rendszer teljes!

1. feladat megoldása.

Két módszer van ilyenkor legalább:

- megmutatjuk, hogy tetszőleges, igazságtáblával adott Boole-függvényt tényleg elő lehet állítani a megadott függvényekkel,
- vagy, és ezt fogjuk most használni, veszünk egy már ismerten teljes rendszert, és annak a függvényeit kifejezzük az aktuális feladatunk függvényeivel.

Most például tudjuk, hogy a $\{\neg, \vee, \wedge\}$ rendszer teljes, így ha a $\neg$, $\vee$, $\wedge$ függvények mindegyikét ki tudjuk fejezni csak a $\{\neg, \wedge\}$ rendszer függvényeivel, akkor kész is vagyunk. (Hiszen ha minden Boole-függvény előáll a $\neg$, $\vee$, $\wedge$ függvények alkalmas kompozíciójával, ezeket meg „ki tudjuk rakni” a $\neg$, $\wedge$ függvényekből, akkor végeredményben mindent elő lehet állítani ezzel a két függvénnyel is).

Ehhez tehát fel kell írjuk a $\neg x$, $x \wedge y$ és $x \vee y$ függvényeket a $\neg$ és $\wedge$ függvények egy-egy alkalmas kompozíciójaként. Az első kettőt nem nehéz:

$$\neg x = \neg x$$
$$x \wedge y = x \wedge y$$

és a deMorgan-azonosságokat alkalmazva kijön az is, hogy

$$x \vee y = \neg(\neg x \wedge \neg y)$$

tehát mivel az ismerten teljes $\{\neg, \wedge, \vee\}$ rendszer mindegyik elemét ki tudtuk fejezni csak a $\neg$ és $\wedge$ függvényeket használva, így a $\{\neg, \wedge\}$ rendszer is teljes.

Fontos: miután választunk egy ismerten teljes H rendszert, akkor

- H függvényeit fejezzük ki, legyenek ők mondjuk a bal oldalon,
- úgy, hogy a jobb oldalon meg az aktuális rendszerünk függvényeit használjuk csak a változókon,
- és a bal oldalon meg magukat a H függvényeit! (Ne pl. $\neg(x \vee y)$-t fejezzük ki, hanem konkrétan $x \vee y$-t.)

Note: itt **egyenlőség** van a két oldal közt, mert ezek most Boole-függvények, amik nem „ekvivalensek”, hanem „egyenlőek” (függvények akkor egyenlőek, ha minden inputra ugyanaz az outputjuk).

2. feladat megoldása.

Most már két teljes rendszert is ismerünk: az egyik a $\{\neg, \wedge, \vee\}$, a másik a $\{\neg, \wedge\}$.

Bármelyiknek az összes függvényét kifejezzük, abból következik, hogy a $\{\neg, \vee\}$ is teljes; a $\{\neg, \wedge\}$ kisebbnek látszik, ezért mondjuk ennek a műveleteit próbáljuk meg kifejezni. Sikerül:

$$\neg x = \neg x \qquad \text{(amikor egy művelet szerepel mindkét halmazban, azzal könnyű a dolgunk)}$$
$$x \wedge y = \neg(\neg x \vee \neg y) \qquad \text{(ez az ötlet pl deMorganből jöhet)}$$

Itt is fontos, hogy a $\{\neg, \wedge\}$ rendszer teljességét ha használjuk, akkor az egyik oldalon konkrétan $\neg x$ és $x \wedge y$ kell szerepeljen, a másik oldalon meg (most) a $\neg$ és $\vee$ függvények tetszőleges kompozíciója állhat.

---

3. feladat megoldása.

Már három teljes rendszert ismerünk: $\{\neg, \vee, \wedge\}$, $\{\neg, \wedge\}$ és $\{\neg, \vee\}$. Bármelyiknek sikerül kifejezni az összes operátorát, kész vagyunk; ebből a $\neg$ az könnyű, hiszen benne van a $\{\rightarrow, \neg\}$ rendszerben is. Ha pl. a $\{\neg, \vee\}$ rendszer teljességét akarjuk használni, akkor rájöhetünk, hogy:

$$\neg x = \neg x$$
$$x \vee y = \neg x \;\rightarrow\; y \qquad \text{(akár próbálgatással)}$$

és ezért a $\{\neg, \rightarrow\}$ is teljes, hiszen ki tudtuk fejezni a $\{\neg, \vee\}$ rendszer összes elemét velük.

Ha a $x \wedge y \;=\; \neg(x \rightarrow \neg y)$ egyenlőségre jövünk rá hamarabb, akkor persze az is elég, hiszen így meg a $\{\neg, \wedge\}$ rendszer teljessége alapján lesz teljes a $\{\neg, \rightarrow\}$.

Még egyszer: az **fontos**, hogy most **nem** az $x \rightarrow y \;=\; \neg x \vee y$ az, amit felírunk, mert nem a $\rightarrow$ kifejezése a $\{\neg, \vee\}$ operátorokkal az, ami megmutatja a $\{\neg, \rightarrow\}$ teljességét, hanem pont az $x \vee y$ kifejezése a $\{\neg, \rightarrow\}$ operátorokkal!

4. feladat megoldása.

Ez az első rendszerünk, melyben nincs negálás, helyette a (0-változós) $\downarrow$ függvényt kaptuk meg.

Az eddig ismert teljes rendszerek közül leginkább a $\{\rightarrow, \neg\}$ hasonlít rá, meg abból az egyik jelet „ingyen” meg is kapjuk, tehát lehet értelme ezzel próbálkozni:

$$\begin{aligned} x \rightarrow y &= x \rightarrow y \\ \neg x &= x \rightarrow \downarrow \qquad\qquad \text{(akár próbálgatással)} \end{aligned}$$

Tehát mivel a már ismerten teljes $\{\rightarrow, \neg\}$ minden operátorát ki tudtuk velük fejezni, így ők is teljes rendszert alkotnak.

(A Hilbert-kalkulus nevű következtető rendszer formulái pl. ezt a két operátort fogják tartalmazni csak; ez azért nem lesz gond, mert ezek szerint ők ketten teljes rendszert alkotnak.)

---

5. feladat megoldása.

Most csak a nand függvényt használva kéne előállítani dolgokat. Ha teljes rendszer, akkor pl. a negálást is meg kéne tudjuk kapni belőle, ami egyváltozós, szóval sok lehetőségünk nincs (megnézzük, hogy ha nandolunk egy x-en, abból mi lesz) és azt kapjuk, hogy

$$\neg x = x \ \mathsf{nand}\ x$$

(hiszen $x\ \mathsf{nand}\ x\ = \neg(x \wedge x) = \neg x$)

Eddig jó, kéne még egy jel, akár a $\rightarrow$, akár a $\vee$, akár a $\wedge$ kifejezése megoldja a feladatot, hiszen a $\{\neg, \rightarrow\}$, $\{\neg, \vee\}$ és $\{\neg, \wedge\}$ rendszerekről már tudjuk, hogy teljesek.

Itt az juthat eszünkbe, hogy $x \wedge y \ = \ \neg\neg(x \wedge y) \ = \ \neg(x\ \mathsf{nand}\ y)$, ami már vezet valahova, hiszen negálni már tudunk nanddal:

$$x \wedge y = (x\ \mathsf{nand}\ y)\ \mathsf{nand}\ (x\ \mathsf{nand}\ y)$$

és így ki tudjuk fejezni a $\{\neg, \wedge\}$ (ismerten teljes) rendszer összes operátorát csak a nandot használva, ezért nand önmagában is teljes rendszert alkot.

6. feladat megoldása.

Az előző feladat tanulságai alapján itt a $\{\neg, \vee\}$ rendszer teljességéből lehet érdemes kiindulni és kapjuk, hogy

$$\begin{aligned} \neg x &= x \ \mathsf{nor}\ x \\ x \vee y &= (x \ \mathsf{nor}\ y)\ \mathsf{nor}\ (x \ \mathsf{nor}\ y), \end{aligned}$$

ezért valóban, a nor művelet önmagában is teljes rendszert alkot.

---

7. feladat megoldása.

Ha az az állítás, hogy **nem** teljes, akkor mutatnunk kell egy Boole-függvényt, ami **nem** fejezhető ki a $\vee$ és $\wedge$ függvények (meg a projekciók) alkalmas kombinációjaként.

Az első kérdés, hogy vajon melyik függvény lehet nem kifejezhető; itt az első ötlet, ami eszünkbe juthat, a $\neg$ lehet (hiszen ha $\neg$ kifejezhető lenne, $\vee$ és $\wedge$ pedig eleve megvan, akkor $\{\vee, \wedge\}$ teljes rendszer lenne – tehát ha nem az, akkor $\neg$ biztos, hogy nem kifejezhető).

Azt, hogy $\neg$ **nem** fejezhető ki, több módon is meg lehet mutatni: az egyik, hogy egy $\neg x = \langle$éselések, vagyolások, $x\rangle$ kifejezés felírásakor a jobb oldalon csak x-et használhatnánk változóként, és a $\vee$ és $\wedge$ operátorokkal köthetjük össze őket. De mivel $x \vee x = x$ és $x \wedge x = x$ is igaz, így bármilyen komplex kifejezést is rakunk ki x-ből, $\vee$-ből és $\wedge$-ből, végeredményben garantáltan az x függvényt kapjuk vissza, a $\neg x$-et pedig nem.

Egy másik megoldás lehet pl. észrevenni, hogy $0 \vee 0 = 0 \wedge 0 = 0$, azaz bármilyen függvényt is építünk össze a változókból, $\vee$-ből és $\wedge$-ből, az eredményként kapott függvény ha csupa 0-t kap, akkor az outputja is 0 lesz, pedig a negálás esetében 0-kból 1-et kéne kapjunk. (1-esekkel is működik: $1 \vee 1 = 1 \wedge 1 = 1$ miatt 1-ből nem lesz 0.)

8. feladat megoldása.

Itt az előző feladat alapján esetleg eszünkbe juthat, hogy $\neg$ talán nem kifejezhető, de az: $\neg x = x \oplus \uparrow$.

Ez azt jelenti, hogy ha a $\{\oplus, \uparrow\}$ rendszer nem teljes, akkor se a $\vee$, se a $\wedge$, se a $\rightarrow$ művelet nem fejezhető ki csak az $\oplus$ és $\uparrow$ művelettel (hiszen ha bármelyiket ki tudnánk fejezni, akkor a negálással együtt lenne egy kifejezett teljes rendszerünk).

Az előző feladat második módszere sem működik: mivel $\uparrow$ a 0 inputból is 1-et készít, az $\oplus$ pedig 1-esekből 0-t (így jöhetett ki a negálás), így a „0-őrzés” és „1-őrzés” tulajdonságokat se tudjuk használni.

Ilyenkor **egy módszer lehet** megpróbálni felírni az összes olyan kétváltozós függvényt, ami kifejezhető egyáltalán a projekciókból ezzel a két jellel. Ha elkezdjük felírni a művelettáblát:

x	y	$\uparrow$	$x \oplus y$
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	0

akkor a kifejezhető függvényeket (most) úgy kapjuk, hogy ahányféleképp csak lehet,

- veszünk két oszlopot
- össze$\oplus$oljuk őket
- ha ezzel olyan oszlopot kapunk, amit korábban még nem, akkor azt is felírjuk,
- ezt addig csináljuk, amíg már nem tudunk új oszlopot kapni.

Befejezve a kompozícióval előálló kétváltozós függvények tábláját ezt kapjuk:

x	y	$\uparrow$	$x \oplus y$	$x \oplus x$	$x \oplus \uparrow$	$y \oplus \uparrow$	$\uparrow \oplus (x \oplus y)$
0	0	1	0	0	1	1	1
0	1	1	1	0	1	0	0
1	0	1	1	0	0	1	0
1	1	1	0	0	0	0	1

és bármelyik két oszlopnak vesszük ezek után a $\oplus$át, mindig olyan oszlopot kapunk, mely már szerepel, így ez az összes kétváltozós függvény, amit az $\uparrow$ és $\oplus$ függvényekből (és a projekciókból) kompozícióval elő tudunk állítani. Mivel ez nem az összes (a fele hiányzik, többek közt a $\vee$, $\wedge$ és $\rightarrow$ függvények), ezért $\{\oplus, \uparrow\}$ nem teljes rendszer.

Itt még pl. az is működne, hogy észrevesszük, hogy az x, y és $\uparrow$ függvények mindegyikében (mint kétváltozós függvényekben) páros sokszor kapunk 1-et (és páros sokszor 0-t), és két oszlop $\oplus$olásakor ez a tulajdonság megőrződik, így nem lehet olyan oszlopot ezekből $\oplus$okkal megkapni, melyekben páratlan sok 1-es szerepelne.

9. feladat megoldása.

Ha ez a rendszer nem teljes, akkor a $\neg$ függvény biztos, hogy nem lehet kifejezhető velük (hiszen ha az lenne, akkor egy csomó teljes rendszer összes műveletét megkapnánk).

Itt az előző, „művelettáblás” módszerrel felírva az összes **egyváltozós** függvényt azt kapjuk, hogy

x	$\uparrow$
0	1
1	1

és hogy bármilyen sorrendben is alkalmazzuk az oszlopokon a $\vee$, $\rightarrow$, $\wedge$, $\leftrightarrow$ műveleteket, mindig a meglévő két oszlopunkat kapjuk vissza, így sem a $\neg$, sem a $\downarrow$ függvények nem fejezhetők ki ezekkel az operátorokkal.

(Az is működik, ha észrevesszük, hogy ezek a függvények mind „őrzik az 1-et”, és ezért nem lehet belőlük felépíteni olyat, ami ne tenné.)

---

10. feladat megoldása.

A $\neg$ már megvan, így elég kifejeznünk akár a $\vee$, akár a $\wedge$ függvényt és vagy a $\{\neg, \vee\}$, vagy a $\{\neg, \wedge\}$ rendszer teljessége miatt kész vagyunk. Kis próbálgatás után rájöhetünk, hogy

$$\neg x = \neg x$$
$$x \vee y = m(x, y, \uparrow)$$

(hiszen ha $x = y = 0$, akkor ők lesznek többen, ha meg valamelyikük is 1, akkor a $\uparrow$ argumentum miatt már m az 1-esből kap többet) és ezért mivel $\{\neg, \vee\}$ teljes, így $\{m, \neg, \uparrow\}$ is az.