Logika gyakorlat – 01
Paradoxonok, ítéletkalkulus szintaxis és szemantika

Warm-upként beszélgessünk pár „paradoxon”ról, amit többen lehet, hogy ismernek:

A falusi borbély

A faluban lakik egy borbély, aki (munkaköri leírásának megfelelően) pontosan azokat a falusiakat borotválja, akik maguk nem borotválkoznak.

Mi a gond ezzel az állítással?

Felmerül a kérdés, hogy maga a borbély borotválkozik-e?

- Ha igen, akkor a „pontosan azokat” *konnektíva* miatt mivel ő borotválkozik (azaz nem igaz rá, hogy nem borotválkozik), így az sem lehet igaz rá, hogy saját magát (mint falusit) borotválná, azaz ekkor ő nem borotválkozik. Ez ellentmondás: azt kaptuk, hogy feltéve, hogy borotválkozik, kijön, hogy nem borotválkozik, tehát ez az ág nem lehetséges.
- Ha nem borotválkozik, akkor (ismét a munkaköri leírás vége szerint) borotválnia kell magát, azaz ekkor meg borotválkozik. Ez megint ellentmondás: ezúttal azt kaptuk, hogy feltéve, hogy nem borotválkozik, kijön, hogy borotválkozik, ez az ág szintén nem lehetséges.

Végeredményben azt kaptuk, hogy se az nem lehet igaz a borbélyra, hogy borotválkozik, se az, hogy nem borotválkozik.

Ebből persze annyi derül ki, hogy ilyen borbély biztosan nem lakik semmilyen faluban, így ez a mondat csak annyira „paradoxon”, hogy biztosan hamis az állítás.

Megjegyzés.

A logika kurzushoz ennek annyiban van köze, hogy (a dimategyről ismert) elsőrendű logikában pl. ezzel a formulával tudjuk leírni a borbély munkaköri leírását:

$$\forall x\Big(\ p(c,x)\ \leftrightarrow\ \neg p(x,x)\ \Big)$$

(nem baj, ha nem tudjuk felparsolni ezt a formulát, később ezen a kurzuson is tanulunk elsőrendű logikát is), és erről a formuláról kiderül (tanulunk majd rá módszereket, amikkel kijön), hogy *kielégíthetetlen* (ami informálisan most annyit fog jelenteni, hogy nem lehet megadni, nem létezik „falusiak olyan halmaza” és benne egy borbély, akit a fenti formula c konstansa jelöl, hogy meg tudnánk adni a „a falusi borotválja b falusit” *predikátum*, amit meg a fenti formula p predikátumjele jelöl úgy, hogy a formula igazzá váljon ebben a faluban). Elsőrendű logikáról később még lesz szó.

A logika könnyű tárgy

Nézzük meg a következő szövegdobozt:

Doboz

Ha ennek a doboznak a tartalma igaz, akkor logikából tanulás nélkül is könnyen ötöst lehet kapni.

Vajon a doboz tartalma igaz vagy hamis? Vajon következik-e bármelyikből, hogy logikából tanulás nélkül is könnyen ötöst lehet kapni? Tudjuk formalizálni ítéletkalkulusban, amit látunk?

Talán itt egyszerűbb, ha a **formalizálás**sal kezdjük (és ezzel átismételjük az ítéletkalkulus szintaxisát[1] és szemantikáját[2]).

Recap: Szintaxis

Ítéletkalkulusban formulák:

a (z ítélet)**változók**, amiket ezen a kurzuson jellemzően $p, p_1, p_2, q, r, \ldots$ jelöl majd,

a (logikai) **konstansok**, kettő lesz: $\uparrow$ és $\downarrow$,

és lehet egyszerűbb formulákból bonyolultabbakat képezni:

- ha F már egy formula, akkor $(\neg F)$ is,
- ha F és G formulák, akkor $(F \vee G)$, $(F \wedge G)$, $(F \rightarrow G)$ és $(F \leftrightarrow G)$ is azok,

más formula pedig nincs, csak amit ezekkel a szabályokkal fel lehet építeni.

A jelek („logikai konnektívák") kimondva: $\neg$ (negáció, „nem"), $\vee$ (diszjunkció, „vagy"), $\wedge$ (konjunkció, „és"), $\rightarrow$ (implikáció, „nyíl") és $\leftrightarrow$ (ekvivalencia, „duplanyíl").

[1] azaz most: „milyen stringekre mondhatjuk rá, hogy formulák, hogy épülnek fel", de a szintaxis a „jelentésükről" nem mond semmit

[2] azaz most: ha adott a logikai változóknak egy értékadása, akkor egy formulának amellett mi az értéke

Recap: Szemantika

Egy formulának **értéke** magában nincs, ahhoz kell egy **értékadás**, ami a változók mindegyikéhez rendel egy logikai értéket a $\{0,1\}$ halmazból (0 a „hamis”, 1 az „igaz”). Egy ilyen értékadást általában $\mathcal{A}$-val, $\mathcal{B}$-vel stb. fogunk jelölni. Tehát pl. $\mathcal{A}(p) = 1$ azt jelenti, hogy a p változó értéke az $\mathcal{A}$ értékadásban / értékadás mellett igaz.

A logikai konnektívák mindegyikéhez tartozik egy **függvény** is, amit ugyanúgy jelölünk, ezt táblázattal adjuk meg:

A	B	$\neg A$	$A \vee B$	$A \wedge B$	$A \to B$	$A \leftrightarrow B$
1	1	0	1	1	1	1
1	0	0	1	0	0	0
0	1	1	1	0	1	0
0	0	1	0	0	1	1

Tehát pl. $1 \vee 0 = 1$, $1 \to 0 = 0$, $0 \to 1 = 1$.

(Azokat a függvényeket, melyek $\{0,1\}$-vektorokat várnak és $\{0,1\}$-beli értéket adnak vissza, mint ezek is, **Boole-függvénynek** hívjuk.)

Ha pedig van egy $\mathcal{A}$ értékadásunk, és egy F formulánk, akkor F értéke $\mathcal{A}$ mellett, amit $\mathcal{A}(F)$ jelöl, a következő lesz:

- ha $F = \uparrow$, akkor $\mathcal{A}(F) = 1$ (ez a „konstans igaz” formula),
- ha $F = \downarrow$, akkor $\mathcal{A}(F) = 0$ (ez a „konstans hamis”),
- ha $F = p$ egy változó, akkor $\mathcal{A}(F) = \mathcal{A}(p)$ (a változók értéke az értékadásból jön),
- ha $F = \neg G$, akkor $\mathcal{A}(F) = \neg\mathcal{A}(G)$ (kiértékeljük az eggyel kisebb formulát, és negáljuk az eredményt),
- ha pedig $F = G \circ H$, ahol $\circ$ a $\vee$, $\wedge$, $\to$, $\leftrightarrow$ konnektívák valamelyike, akkor $\mathcal{A}(F) = \mathcal{A}(G) \circ \mathcal{A}(H)$ (kiértékeljük az eggyel kisebb formulákat, és az eredményeken alkalmazzuk a konnektíva által megadott Boole-függvényt).

Ezekkel a jelekkel a dobozbeli szöveg „ha ennek a doboznak a tartalma igaz, akkor logikából tanulás nélkül is könnyen ötöst lehet kapni” -et leíró formulát ha megpróbáljuk felépíteni, így gondolkodhatunk:

- először, ami itt (dimatból) „atomi állítás” címen futott, abból lesznek a változók: most lehet pl. egy p változónk, aminek az értéke 1 lesz, ha a doboz tartalma igaz és 0, ha a doboz tartalma hamis, és egy q változónk, ami meg a (hosszú) „logikából tanulás nélkül is könnyen ötöst lehet kapni” rész igazságértékét veszi majd fel
- eztán egy „ha p igaz, akkor q is” alakú mondatot látunk a dobozban, ezt pont a $\to$ konnektíva írja le, így a doboz tartalmára a $p \to q$ formulát kapjuk
- (itt jön a problémás rész) és mivel ez a $p \to q$ mondat magának a doboznak a tartalma (amit p-vel jelöltünk), így most első közelítésben[3] egy $p \leftrightarrow (p \to q)$ formulát kapunk („a doboz tartalma pont akkor igaz, ha a doboz tartalma igaz”), aminek igaznak „kell” lennie

[3] később kiderül, hogy egy $p \equiv p \to q$ felírás lehet, hogy korrektebb lenne ebben az esetben, de a $\equiv$ jellel ma még nem foglalkozunk

De ha $\mathcal{A}(p \leftrightarrow (p \to q))$ igaz, akkor ez kétféleképp lehet:

- ha $\mathcal{A}(p) = 1$, azaz ha a doboz tartalma igaz, akkor $\mathcal{A}(p \to q)$ is 1 kell legyen, ami – mivel $\mathcal{A}(p) = 1$ – a $\to$ táblája szerint csak úgy lehet, ha $\mathcal{A}(q) = 1$, ebben az esetben tehát logikából könnyű tanulás nélkül ötöst kapni.
- ha $\mathcal{A}(p) = 0$, azaz ha a doboz tartalma hamis, akkor $\mathcal{A}(p \to q)$ is 0 kell legyen, ezúttal pedig a $\to$ műveleti táblájáról és abból, hogy $\mathcal{A}(p) = 0$, azt olvashatjuk le, hogy ez az eset nem lehetséges.

Látszólag tehát levezettük azt (pusztán abból, hogy felírtunk egy mondatot egy szövegdobozba), hogy annak a mondatnak igaz **kell** legyen az értéke, és az szintén egy igaz állítás, hogy logikából tanulás nélkül könnyen lehet ötöst kapni.

Persze itt valahol elkövettünk egy hibát, aki többet szeretne tudni erről, Curry paradoxon néven fut, röviden: a probléma az „önhivatkozással” van, általánosságban nem megengedett pl. egy változóval „elnevezni” egy olyan formulát, melyben ez a változó valahol belül szerepel is.

Váratlan akasztás

A bíró a következő ítéletet hozza egy tárgyaláson: az elítéltet kivégzik, a következő hét valamelyik hétköznapján délben, de még aznap reggel se fogja tudni, hogy biztosan aznap végzik ki majd.

Mit gondolunk erről az ítéletről?

Ami adja magát: ha eljön a péntek reggel és emberünk még mindig életben van, akkor tudni fogja, hogy aznap végzik ki, hiszen arról volt szó, hogy valamelyik hétköznap végzik ki, az összes többi nap meg már elmúlt. Ezért emberünk fejben ki is tudja zárni, hogy péntek lesz a kivégzés.

Ezek után úgy okoskodhat, hogy ezek szerint ha eljön a csütörtök reggel, tudva azt, hogy péntek kizárva, már tudná, hogy csütörtök lesz a kivégzés, de a bíró azt mondta, hogy nem fogja tudni, tehát így fejben kizárja a csütörtököt is.

Így haladva visszafelé kizárja a szerdát, a keddet és a hétfőt is, levezeti, hogy tehát egyik nap se végezhetik ki az ítélet szerint, hátradől és megnyugszik.

Szerda délben meg rákopog a hóhér, hogy jöjjön vele, ő pedig ezen rettenetesen meglepődik, mert erre aztán igazán nem számított és rájön, hogy a bírónak mégis igaza volt.

Na de hol tévedett?

Ez a paradoxon a váratlan akasztás paradoxon címen (is) ismert, ebben is van rejtetten (legalábbis az előzőhöz képest) egy „önhivatkozás”: az állítás (a bírói ítélet) maga arról mond valamit, hogy őbelőle mi következik és mi nem. Öt helyett két nappal ez kb. úgy fogalmazható meg (ahol p az ítélet, q_1 az, hogy hétfőn végzik ki, q_2 pedig, hogy kedden), hogy „p pontosan azt mondja ki, hogy (q_1 és p-ből nem vezethető le q_1) vagy (q_2 és nem q_1 és p-ből és $\neg q_1$-ből nem vezethető le q_2)” – ennek a paradoxonnak a feloldása messze túlmutat az első heti anyagon, érdeklődők pl. itt és számos más helyen olvashatnak róla.

Azt láthattuk az előzőekből, hogy vannak olyan állítások, helyzetek, amiket ha logikailag próbálunk formalizálni, kezelni, könnyen hibát követhetünk el, szerencsére ez azért nem ennyire általános.

Ha egy problémát szeretnénk megoldani (általánosan megfogalmazva: kapunk valamilyen inputot, vissza kell adnunk rá valamiféle megoldást, ha van, vagy lejelenteni, hogy nincs), azt sokszor „vissza tudjuk vezetni” egy ítéletkalkulus-beli formula **modelljének** (modellje az F formulának: egy olyan $\mathcal{A}$ értékadás, mely kielégíti F-et, vagyis melyre $\mathcal{A}(F) = 1$) a keresésére, azaz:

- adunk egy algoritmust, ami az adott probléma inputjából egy logikai formulát készít,
- úgy, hogy a készített formula modelljeiből vissza tudjuk kapni az eredeti problémánk inputjának egy-egy megoldását.

Ezt úgy is mondhatjuk, hogy „formalizáljuk” az eredeti problémánkat (ítéletkalkulusban).

Nézzük meg ezt is egy példán:

1. feladat.

Formalizáljuk azt, hogy egy input gráfban keresünk teljes párosítást!

(Ha valaki nem emlékszik: egy gráfban egy „teljes párosítás” annyit tesz, hogy kiválasztunk éleket a gráfból úgy, hogy minden csúcshoz pontosan egy rá illeszkedő él legyen kiválasztva.)

2. feladat.

Formalizáljuk azt, hogy egy input gráf csúcsait próbáljuk színezni két színnel úgy, hogy szomszédos csúcsok különböző színt kapjanak!

3. feladat.

(Nehezebb.)

Formalizáljuk azt, hogy egy input gráf csúcsait próbáljuk színezni három színnel úgy, hogy szomszédos csúcsok különböző színt kapjanak!

4. feladat.

(Még nehezebb.)

Formalizáljuk azt, hogy egy input gráfban keresünk Hamilton-utat (azaz olyan utat, mely minden csúcsot pontosan egyszer érint!)

1. feladat megoldása.

Először azon is érdemes elgondolkodni, hogy hogyan tudunk elkódolni logikai változókkal egy **lehetséges** megoldást? Most egy megoldásban az eredeti gráfnak az éleit mindegyiket vagy beválasztjuk a párosításba, vagy nem; ez egy bináris döntés, ha ilyet látunk, az jó jel arra, hogy ezekre vezessünk be egy-egy változót.

Tegyük is ezt: ha (u, v) egy él a gráfban, akkor vegyünk fel egy (mondjuk) $p_{u,v}$ változót.

Ezzel már el tudunk kódolni **lehetséges** megoldásokat: élhalmazokat, amik vagy teljes párosítások, vagy nem és biztos, hogy minden egyes teljes párosítás meg is felel (pontosan) egy értékadásának a változóknak (a kiválasztott élekhez kapcsolt változók lesznek igazak, a nemkiválasztottakhoz kapcsolt változók pedig hamisak).

Miután ezt a kódolást kitaláltuk, még egy (nehezebb) lépés hátravan: olyan formulát kell felírnunk, ami **ellenőrzi**, hogy egy adott értékadás tényleg egy megoldást (most egy teljes párosítást) kódol-e el. Azaz olyan formulát kell alkossunk, ami egy $\mathcal{A}$ értékadás mellett igaz lesz, ha az $\mathcal{A}$ értékadás tényleg egy teljes párosítást kódol el, és hamis lesz, ha nem azt kódol el.

Itt most tehát azt kell lényegében kifejeznünk formulával, hogy „minden csúcsra igaz, hogy a rá illeszkedő élekhez kapcsolt változók közül **pontosan egy** igaz". Ez pont azt fogja jelenteni, hogy minden csúcsra pontosan egy kiválasztott él illeszkedik.

Ugyan azt látjuk, hogy „minden csúcsra igaz", ebből mégsem egy – dimatból ismert – $\forall$ kvantor lesz (az egyébként sincs ítéletkalkulusban), hanem egy (hosszú) éselésben érdemes ilyenkor gondolkodni: fel fogjuk írni külön, hogy „az első csúcsra illeszkedő élek változóiból is pontosan egy igaz ÉS a második csúcsra illeszkedő élek változóiból is pontosan egy igaz ÉS ...", az összes csúcsra. Mivel a gráfunk persze véges, ez megoldható.

Tehát a feladat maradék része az, hogy ha adott néhány változó (annyi, ahány él illeszkedik egy csúcsra), mondjuk $p_1, p_2, \ldots, p_n$, akkor milyen formulával tudjuk „azt mondani", hogy ezek közül pontosan egy igaz? (azaz: írjunk fel egy formulát, aminek pont akkor modellje egy értékadás, ha az az értékadás ezekből a változókból pontosan egyet állít 1-re, a többit 0-ra?)

Ha valaki ismeri a XOR műveletet, az hiheti azt, hogy a p_1 XOR p_2 XOR ... XOR p_n formula „jó lesz". elvégre két változóra tényleg akkor ad 1-et, ha közülük pont az egyik 1. Csakhogy, ha kiszámoljuk az 1 XOR 1 XOR 1 értéket, mondjuk előbb az első xort kiértékelve azt kapjuk, hogy 0 XOR 1, azaz 1, erre is 1-et ad, pedig nem egy egyes van az inputban, hanem három. Valójában az ilyen „XOR lánc" **paritást** számol, akkor lesz 1, ha páratlan sok tag értéke 1-es, ezért ez nem lesz nekünk most jó.

Egy megoldás persze felírni ennek a **diszjunktív normálformáját**, ha ismerünk ilyet: tételesen felsoroljuk, hogy mikor lesz igaz a formula, ez pl. ha p_1, p_2 és p_3 a változóink, a következő lesz:

$$(p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge p_3),$$

általában ha van n változónk, akkor ez most n darab, egyenként n „literál" (literál: változó, vagy negált változó) éseléséből álló tag vagyolása, összesen n^2 hosszú.

Ez a megoldás jó, valóban formalizálja a problémát, azonban később látni fogjuk, hogy a diszjunktív normálforma helyett pont hogy a konjunktív normálformák lesznek azok, melyeken a logikai következtető motorok hatékonya(bba)n tudnak üzemelni, ezért próbáljunk meg konjunktív normálformát felírni rá!

Sajnos ha ebből a formulából szeretnénk konjunktív normálformát készíteni, az kb. 2^n tagból fog állni, az meg gombócból is sok.

Jöhet az ötlet, hogy ha azt is leírjuk, hogy „legalább egy igaz közülük” és azt is, hogy „legfeljebb egy igaz közülük”, az pont azt jelenti, hogy „pontosan egy igaz közülük”, és ezek közül a „legalább egy” az könnyű:

$$p_1 \vee p_2 \vee \ldots \vee p_n$$

tényleg pont akkor lesz igaz egy értékadás mellett, ha a változók közül legalább az egyiket igazra állítja az értékadás.

Kérdés, hogy a „legfeljebb egy igaz” hogyan kezelhető? Úgy, hogy **páronként letiltjuk**, hogy egyszerre kettő igaz legyen. Ez pl. megint a p_1, p_2 és p_3 változókat használva:

$$\neg(p_1 \wedge p_2) \;\wedge\; \neg(p_1 \wedge p_3) \;\wedge\; \neg(p_2 \wedge p_3),$$

az első azt mondja, hogy egyszerre nem lehet igaz a p_1 és a p_2, a második, hogy egyszerre nem lehet igaz p_1 és p_3 stb., ha így készítjük el a formulát, akkor ha bármilyen értékadásban egyszerre lenne igaz p_i és p_j, az hamissá tenné a $\neg(p_i \wedge p_j)$ tagot, és mivel ez egy hosszú éselés, így az egész formulát is, ez tehát tényleg a „legfeljebb egy igaz közülük”-et formalizálja.

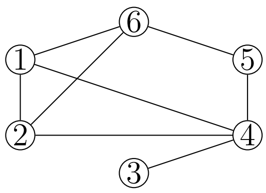
Mondjuk nem konjunktív normálforma, de könnyen azzá tehető:

$$(\neg p_1 \vee \neg p_2) \;\wedge\; (\neg p_1 \vee \neg p_3) \;\wedge\; (\neg p_2 \vee \neg p_3),$$

általában az összes párra egy ilyen „klózt” (literálok vagyolása) építünk és az összes generált klózt összeéseljük.

Tehát a formulánkat a következőképp kapjuk: minden csúcson végigmegyünk és a rá illeszkedő élekhez kapcsolt változókra felírjuk a „legalább egy igaz közülük” és a „legfeljebb egy igaz közülük” formulákat, a végén az egészet összeéseljük.

Például ebből a gráfból:



ezt a formulát kapjuk (az első két sor a „legalább egy”, a többi a „legfeljebb egy” rész, utóbbi csúcsonként új sorba szedve):

$$\begin{aligned}
&(p_{12} \vee p_{14} \vee p_{16}) \;\wedge\; (p_{12} \vee p_{24} \vee p_{26}) \;\wedge\; (p_{34}) \;\wedge\\
\wedge\, &(p_{14} \vee p_{24} \vee p_{34} \vee p_{45}) \;\wedge\; (p_{45} \vee p_{56}) \;\wedge\; (p_{16} \vee p_{26} \vee p_{56}) \wedge\\
\wedge\, &(\neg p_{12} \vee \neg p_{14}) \;\wedge\; (\neg p_{12} \vee \neg p_{16}) \;\wedge\; (\neg p_{14} \vee \neg p_{16}) \wedge\\
\wedge\, &(\neg p_{12} \vee \neg p_{24}) \;\wedge\; (\neg p_{12} \vee \neg p_{26}) \;\wedge\; (\neg p_{24} \vee \neg p_{26}) \wedge\\
\wedge\, &\text{ (a hármas csúcsra csak egy él illeszkedik, nem generál ilyen tagot) } \wedge\\
\wedge\, &(\neg p_{14} \vee \neg p_{24}) \wedge (\neg p_{14} \vee \neg p_{34}) \wedge (\neg p_{14} \vee \neg p_{45}) \wedge (\neg p_{24} \vee \neg p_{34}) \wedge (\neg p_{24} \vee \neg p_{45}) \wedge (\neg p_{34} \vee \neg p_{45}) \wedge\\
\wedge\, &(\neg p_{45} \vee \neg p_{56}) \wedge\\
\wedge\, &(\neg p_{16} \vee \neg p_{26}) \;\wedge\; (\neg p_{16} \vee \neg p_{56}) \;\wedge\; (\neg p_{26} \vee \neg p_{56}).
\end{aligned}$$

2. feladat megoldása.

TODO

3. feladat megoldása.

TODO

4. feladat megoldása.

TODO