

Van-e a nyuszinak sapkája?

A számítógépes jelentésábrázolás problémái

Szécsényi Tibor

SZTE Általános Nyelvészeti Tanszék

1. Bevezetés

A tudomány és a technika fejlődésében két emberi tulajdonság is igen fontos szerepet játszik. Az első a kíváncsiság: az emberek kíváncsiak, hogy mi van a hegyen túl, hogy honnan jön a folyó, hogy miből lesz az eső, hogy miért esnek le a dolgok a magasból, a madarak pedig miért nem, hogy miért beszélnek az emberek más nyelveken is – és elindulnak megnézni, próbálgatnak, kísérleteznek, gondolkoznak. Én például szeretek rejtvényt fejteni, kíváncsi vagyok, hogy mi a vicc poénja. Vannak a rejtvényűjságokban olyan típusú rejtvények, amiben meg van adva négy emberről mindenféle tulajdonság, hogy Jenő postás, de a vízvezeték-szerelő még nem olvasta a Harry Pottert, azt csak a skodás olvasta, és egy táblázatban be lehet ikszelni a megfelelő mezőbe az információkat, és a végén kiderül, hogy Lujza szereti-e a spenótot. És kíváncsiak vagyunk, hogy szereti-e. Én logikus gondolkodású embernek tartom magam, még tanítok is ilyeneket, úgyhogy nekiállok megfejteni, de sokszor elrontom.

A másik emberi tulajdonság a lustaság. Lusták vagyunk elgyalogolni a Balatonig, így hát inkább kitaláljuk, hogyan juthatnánk el oda gyaloglás nélkül is. Lusták vagyunk vonalakat rajzolni a víztorony falára egyméterenként (meg nem is engedik), hogy megtudjuk, milyen magas, inkább kitaláljuk a trigonometriát. Lusták vagyunk megfejteni a rejtvényt, meg úgyis elrontanánk, ezért (bár egyszerűen a rejtvényűjság végére is lapozhatnánk, de nem, az kispályás, hanem) írunk egy számítógépprogramot, ami megoldja helyettünk a feladatot. Ha már úgyis tanítunk ilyeneket is.

Nézzük tehát a rejtvényt, aminek a megfejtésére kíváncsiak vagyunk, és amit a lustaságunknak hála unortodox módon fejthetünk meg jelen tanulmány során!

Adva van tehát három állat, egy nyúl, egy róka meg egy farkas, és tudjuk, hogy az egyiket Arnoldnak hívják, a másikat Bruce-nak, a harmadikat pedig Sylvesternek. Mindegyikőjüknek van egy csak rá jellemző ruhadarabja, egy sál, egy sapka és egy kesztyű, az egyik piros, a másik barna, a harmadik pedig fekete. És tudjuk még ezeken kívül a következő információkat:

- (1) Bruce-on sapka van, de nem barna.
- (2) A nyuszi pirosat visel.
- (3) Arnold nem róka, és nem visel sálát.
- (4) A farkasnak kesztyűje van, de nem fekete.

A kérdés az, ami a cikk címében is szerepel, hogy *vajon van-e a nyuszinak sapkája?*

Aki kíváncsi, de nem lusta, az gyorsan meg is oldhatja a rejtvényt a következő táblázat kitöltésével:

	nyuszi	róka	farkas	sapka	sál	kesztyű
Arnold						
Bruce						
Sylvester						
piros						
barna						
fekete						

Aki nem kíváncsi, az valószínűleg el sem jutott eddig a cikk olvasásában. Aki kíváncsi és eléggé lusta is, az a cikk végére lapozva megtalálhatja a választ a kérdésre. Akiben viszont a kíváncsiság és a lustaság pont a kellő arányban található meg, az a cikket végigolvasva megtudhatja, hogy hogyan segíthet a formális logika, a dinamikus szemantika és a számítógép-programozás az ilyen feladatok megoldásában.

2. A formális logika

Az imént ismertetett rejtvényt logikai rejtvénynek tartjuk. Nem szükséges hozzá különösebb lexikai tudás, nem beugratós, pusztán logikával megoldható. Nézzük meg tehát, hogy mi is az a logika, és hogyan segíthet nekünk a feladat megoldásában!

A logika (vagy formális logika) egy nyelv, aminek a segítségével állításokat fogalmazhatunk meg, csakúgy, mint a magyar vagy bármelyik természetes nyelv felhasználásával. A logika és a természetes nyelvek között azonban van egy nagy különbség. A logika nyelve formalizált: erősen szabályozott, előre rögzített szókinccse van, ezeket a szavakat csak jól meghatározott módon lehet kombinálni, hogy mondatokat alkossanak. A logika szavainak a „jelentését” is pontosan ismerjük, az összetett logikai kifejezések jelentése pedig egyértelműen meghatározható a részek jelentésének ismeretében,

vagyis a logika kompozicionális. A természetes nyelvekre ezek nem feltétlenül jellemzőek.

A rejtvény megoldásának az első lépése az, hogy a megadott magyar nyelvű információkat a formális logika nyelvére fordítjuk le. Pontosabban: valamilyen formális logikai nyelvre fordítjuk le, ugyanis többféle logika is elképzelhető. Szerencsére a rejtvény állításai nem túl bonyolultak, ebben az esetben elegendő az elsőrendű predikátumlogikát segítségül hívni.

2.1. Az elsőrendű predikátumlogika nyelve

A logika nyelvében a természetes nyelvekhez hasonlóan különböző típusú, „szófajú” szavakat használunk. A legegyszerűbb típusú szavak a nevek, amelyek a természetes nyelv tulajdonneveihez hasonlóak. A nevek mindig egy-egy dologra, individuumra utalnak, de egy szituációban, modellben mindig ugyanarra. A mi rejtvényükhöz három névre lesz szükség: arnold, bruce, és sylvester.¹

A nevekhez hasonlóan individuumokra utalnak a logikai változók, az individuumváltozók is. Abban különböznek a nevektől, hogy míg azoknak a jelölete állandó, mindig ugyanarra az individuumra utalnak, a változók jelölete nem rögzített, mint ahogy az *ő* magyar névmás sem mindig ugyanazt a személyt jelöli. Logikánkban a magyar ábécé végéről származó nagybetűk lesznek a változók: X, Y, Z...

A következő típus a predikátum² típus, aminek a segítségével az individuumok tulajdonságait adhatjuk meg. A természetes nyelvben tulajdonságokat fejeznek ki a melléknemek (*Arnold izmos*), a főnevek (*Arnold oroszlán*) és az igék közül az intranszitiv igék (*Ő fut*). Szóval a logikában is lesznek szavak, amelyek tulajdonságot fejeznek ki, az ún. egyargumentumú predikátumok: izmos, oroszlán, fut.

Ezekkel, valamint a nevekkal és a változókkal máris összeállíthatjuk a legegyszerűbb predikátumlogikai állításokat:

1. Ha p egy predikátum, a pedig egy név vagy egy változó, akkor $p(a)$ egy állítás.
pl. izmos(arnold), oroszlán(arnold) és fut(X),

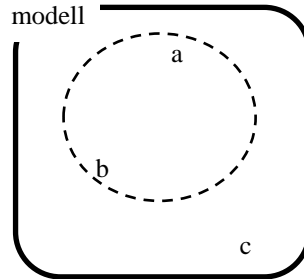
amik az iménti példamondatok logikai megfelelői, fordításai.

Egy-egy ilyen állítás igazsága nem magától értetődő, bizonyos szituációban igaz, máskor meg nem. Az állítások igazságértékét mindig egy modellben határozhatjuk meg. A modell individuumokból áll, és a logika nevei és változói ezekre az individuumokra

¹ A logikai kifejezéseket más betűtípussal különböztetem meg a magyar nyelv szavaitól, a neveket pedig kis betűvel kezdem – ez majd a számítógépes feldolgozásnál lesz érdekes.

² Pontosabban az egyargumentumú predikátum. A predikátumlogikákban szoktak többargumentumú predikátumokat is használni, de nekünk most nem lesz rájuk szükségünk.

utalnak. A predikátumokat is a modellben lehet értelmezni: egy egyargumentumú predikátum jelölete ennek az individuumhalmaznak egy részhalmaza. Ha a modell az a , b és c individuumokat tartalmazza, az arnold az a individuumot jelöli, és az a és b individuumok az izmosak (vagyis az izmos jelölete az $\{a, b\}$ halmaz, akkor az izmos(arnold) logikai állítás igaz (ebben a modellben), mert $a \in \{a, b\}$. Ha a sylvester név jelölete a c individuum, akkor az izmos(sylvester) mondat hamis.



Az így kapott elemi mondatokat kombinálhatjuk a \sim , \wedge , \vee és \rightarrow logikai műveletekkel, amik a természetes nyelvben rendre a tagadás (*nem igaz az, hogy...*), az és, *de, pedig, a vagy, illetve a ha... akkor...* logikai megfelelői:

2. Ha α egy állítás, akkor $\sim\alpha$ is állítás.
pl: \sim izmos(arnold)
 \approx *Nem igaz az, hogy Arnold izmos. \approx Arnold nem izmos.*
3. Ha α és β egy-egy állítás, akkor $\alpha \wedge \beta$ is állítás.
pl: \sim izmos(arnold) \wedge fut(bruce)
 \approx *Arnold nem izmos, de Bruce elfutott.*
4. Ha α és β egy-egy állítás, akkor $\alpha \vee \beta$ is állítás.
pl: izmos(arnold) \vee fut(arnold)
5. Ha α és β egy-egy állítás, akkor $\alpha \rightarrow \beta$ is állítás.
pl: oroszlán(arnold) \rightarrow \sim fut(arnold)
 \approx *Ha Arnold oroszlán, akkor nem fut.*

Az ilyen összetett állítások igazságértékét csak a bennük szereplő részállítások igazságértéke határozza meg. Ha az α állítás egy modellben igaz, akkor $\sim\alpha$ hamis ugyanabban a modellben, és fordítva. Az $\alpha \wedge \beta$ állítás pontosan akkor igaz egy modellben, ha α és β is igaz ugyanott, az $\alpha \vee \beta$ állítás pontosan akkor hamis, ha sem α , sem pedig β nem igaz, végül pedig $\alpha \rightarrow \beta$ pontosan akkor hamis, ha az α igaz, de β hamis.

Az eddig felsorolt kifejezéseken kívül még két szót, szimbólumot használunk a predikátumlogikában: az \forall univerzális kvantort és az \exists egzisztenciális kvantort. Ezekkel egy tulajdonság univerzalitását vagy valamilyen tulajdonságú individuum létezését fejezhetjük ki:

6. Ha X egy változó és α egy állítás, akkor $\forall X\alpha$ is állítás.
 pl: $\forall X(\text{izmos}(X))$
 \approx Minden(ki) izmos. \approx Minden individuum (olyan, hogy ő) izmos.
7. Ha X egy változó és α egy állítás, akkor $\exists X\alpha$ is állítás.
 pl: $\exists X(\text{oroszlán}(X))$
 \approx Van oroszlán. \approx Van legalább egy olyan individuum, ami oroszlán.

Vegyük észre,³ hogy az alapkifejezésekből az 1–7. szabályok szerint összeépített állítások igazsága nem függ semmi mástól, csak attól, hogy a bennük szereplő nevekhez és predikátumokhoz a modellben milyen individuumokat, illetve individuumhalmazokat társítottunk. A logika ilyen felépítése miatt nevezzük a logikát kompozicionálisnak.

2.2. A rejtvény állításainak formalizálása

A bevezetésben megadott tények logikai megfelelőinek megadásához először is el kell döntenünk, hogy milyen neveket és predikátumokat fogunk használni. Ezt nagyban befolyásolja az, hogy a modellben hány elemet tételezünk fel.

Lehet egyrészt egy olyan modellt elképzelni, amelyben van három állat-individuum, továbbá három ruhadarab-individuum. Az, hogy csak az állatoknak van nevük a modellben, a ruháknak meg nincs, nem probléma, a való életben sem szoktunk minden dolognak külön nevet adni, a zoknijaink például általában név nélkül léteznek. Ha hat individuum van a modellben, akkor viszont valahogyan ki kell tudnunk fejezni azt, hogy az egyik birtokolja a másikat, például hogy a nyuszinak van sapkája. Az előző alfejezetben ismertetett predikátumlogikában ezt nem tudjuk megtenni, ahhoz kétargumentumú predikátumokra is szükségünk lenne. Szerencsére azonban kizárólag egyargumentumú predikátumok, vagyis tulajdonságok használatával is tudjuk formalizálni a rejtvény állításait.

Álljon tehát a modellünk három individuumból: a -ból, b -ből és c -ből! Őket el is nevezhetjük rögtön arnold-nak, bruce-nak és sylvester-nek. Az individuumok mind

³ A predikátumlogika ezen vázlatos ismertetésén túl sokat segíthet az észrevevésben egy részletesebb, formálisabb tankönyv áttanulmányozása, mint például az Alberti (2006) vagy a Partee et al. (1990).

valamilyen állatok, vagyis szükségünk lesz három predikátumra: róka, farkas és nyuszi. Az individuumok továbbá rendelkezhetnek valamilyen ruhadarabbal is, ezért további három predikátumra lesz szükségünk: sapka, sál és kesztyű.⁴ A harmadik predikátumhármas a színeknek megfelelő három predikátum lesz: piros, barna és fekete. Ez azért lesz megfelelő, mert bár a színek a ruhadarabok tulajdonságai, nem pedig az állatokéi, de a most úgy egyszerűsíthetjük a leírást, mintha az állatokat tekintenénk színesnek.

Vegyük tehát sorra a rejtvény állításait, és fogalmazzuk meg azokat a mi elsőrendű predikátumlogikánk nyelvén!

(1) Bruce-on sapka van, de nem barna.

A *Bruce-on sapka van* egy egyszerű atomi mondattá fordítható: *sapka(bruce)*. A *nem barna* tagmondat az előző tagmondat sapkájára vonatkozik, ti. hogy az nem barna, de mivel tudjuk, hogy Bruce-nak van sapkája, egyszerűen rá vonatkoztatjuk: *~barna(bruce)*. A két tagmondat a *de* kötőszóval kapcsolódik egymáshoz, ennek a logikában az \wedge felel meg, az első állítás logikai fordítása tehát a következő:

(1') $sapka(bruce) \wedge \sim barna(bruce)$

A következő mondatban nem szerepel név, hanem csak úgy utalunk az egyik szereplőre, hogy *a nyuszi*:

(2) A nyuszi pirosat visel.

Mivel azonban tudjuk, hogy csak egyetlen nyuszi szerepel a feladatban,⁵ a mondatot nyugodtan átfogalmazhatjuk úgy, hogy *minden nyuszi pirosat visel*, vagy még inkább úgy, hogy *minden olyan dologra, amire igaz az, hogy ő nyuszi, az is igaz, hogy ő pirosat visel*. Ez már olyan nyakatekert, hogy szinte logikailag van:

(2') $\forall X(nyuszi(X) \rightarrow piros(X))$

A harmadik, Arnoldos mondatunk az első mondathoz hasonlóan könnyen megfogalmazható a logika nyelvén:

⁴ Bár a kesztyű(arnold) állítás elég bután fest, mintha azt mondanánk, hogy *Arnold egy kesztyű*, a logikai formulát most tekintjük inkább az *Arnold kesztyűs* mondat logikai megfelelőjének.

⁵ Különben nem is utalhatnánk rá úgy, hogy *a nyuszi*. Gondoljunk csak bele, ha lenne két nyuszi is, mit jelentene az, hogy *a nyuszi*?

(3) Arnold nem róka, és nem visel sálát.

(3') \sim róka(arnold) \wedge \sim sál(arnold)

A negyedik mondat pedig a második mondatához hasonlóan parafrázálható logikai közeli mondattá: *minden olyan dologra, amire igaz az, hogy ő farkas, igaz az is, hogy ő kesztyűs, mégpedig (=és) nem fekete.*

(4) A farkasnak kesztyűje van, de nem fekete.

(4') $\forall X(\text{farkas}(X) \rightarrow (\text{kesztyű}(X) \wedge \sim \text{fekete}(X)))^6$

És ezzel készen is vagyunk a rejtvény állításainak a formalizálásával!

3. Számítógépes reprezentáció

Azt, hogy egy állítás egy megadott modellben igaz-e, vagy sem, a logika kompozicionalitása miatt automatikusan megállapíthatjuk: elegendő rekonstruálni, hogy az állítás hogyan épül föl az 1–7 szabályok szerint, majd lépésről lépésre meghatározni a részkifejezések igazságértékét a modellben, s így végül eljutunk a teljes állítás igazságértékéhez. Ez a folyamat annyira automatikus, hogy könnyen lehet olyan számítógépes programot készíteni, ami ugyanezt végrehajtja.

Először is meg kell adnunk a modellt, amiben interpretálni kívánjuk a logikai kifejezéseinket. Ez egyrészt egy individuumhalmazból áll, valamint annak a rögzítéséből, hogy a logikában használt nevekhez és predikátumokhoz ennek az individuumhalmaznak milyen részeit társítjuk. A nevek mindegyikéhez egy-egy individuumot, a predikátumokhoz pedig egy-egy individuumhalmazt. Ezzel meg is adtuk a logikai nyelvünk szótárát a szavak „jelentésével” együtt.

A következő lépésként a logikai kifejezéseket számítógéppel is könnyen kezelhető formára kell hozni: bár a papírra könnyen le tudjuk írni az univerzális kvantor \forall jelét, a számítógép billentyűzetén elég körülményes bevinni. Tehát ki kell dolgozni még a logikai formuláknak a számítógépes megjelenítését is.

Végül meg kell írni a programot, ami a megadott logikai formulák interpretációját határozza meg a megadott modellben.

⁶ Látható, hogy a $\text{kesztyű}(X) \wedge \sim \text{fekete}(X)$ kifejezés zárójelbe került, így ez a komplex állítás van \rightarrow kifejezéssel hozzákapcsolva a $\text{farkas}(X)$ állításhoz. Ha a $\text{farkas}(X) \rightarrow \text{kesztyű}(X)$ lett volna zárójellel egységbe foglalva, akkor is jó logikai formulát kapnánk, de az annak a magyar mondatnak felelne meg, hogy *Minden individuum olyan, hogy ha ő farkas, akkor kesztyűt is visel, de nem fekete (függetlenül attól, hogy farkas-e)*, még magyarabban: *Semmi sem fekete, és minden farkas kesztyűs.*

Az interpretáló programot Prolog programozási nyelven dolgozzuk ki.⁷ Ennek több oka is van. Az egyik az, hogy ezt a programozási nyelvet kimondottan ilyen feladatok elvégzésére fejlesztették ki, a neve is a *logika* és a *programozás* szavak első szótagjából tevődik össze. A Prolog nyelven elsősorban nem algoritmusokat és szubrutinokat írunk, hanem tényeket és definíciókat fogalmazunk meg, amik alapján a beépített algoritmus képes következtetni újabb tényekre – tényleg tisztára olyan, mint amit a logikától elvár a hétköznapi ember. Logikusan gondolkozik.

A másik ok, amiért most érdemes a Prologot választani, az az, hogy nekünk már nem kell semmilyen programozást csinálni, megtette helyettünk Patrick Blackburn és Johan Bos, akik a *Representation and Inference for Natural Language: A First Course in Computational Semantics* című könyvükhöz (Blackburn–Bos 2005) már készítették egy elsőrendű predikátumlogikai interpretáló programot. Ismerkedjünk meg vele!

A modellt a következő formában kell megadnunk (Blackburn és Bos példája):

```
model ([d1, d2, d3, d4, d5, d6],
       [f(0, jules, d1),
        f(0, vincent, d2),
        f(0, pumpkin, d3),
        f(0, honey_bunny, d4),
        f(0, yolanda, d4),
        f(1, customer, [d1, d2, d5, d6]),
        f(1, robber, [d3, d4])])
```

A `model` kulcsszó után szögletes zárójelben felsoroljuk a modellben szereplő individuumokat, a szerzők példájában hat ilyen van. Ezután egyesével megadjuk a nyelv által használt neveket és a predikátumokat a modellbeli megfelelőjünkkel. A `jules` név a `d1` individuumot jelöli, a `vincent` a `d2`-t, és így tovább a *Ponyvaregény* többi szereplőjével. A `d5` és `d6` individuumoknak nincs nevük. A nevek előtti 0 jelzi azt, hogy ezek nevek. Az egyargumentumú predikátumok előtt 1 található, és a modellmegadásból láthatjuk például, hogy a `d3` és a `d4` individuumok rendelkeznek a `robber` tulajdonsággal, vagyis ők rablók.

A programban a predikátumlogikai formulákat a következőképpen adhatjuk meg. A nevek és a predikátumok ugyanúgy néznek ki, mint a logikában, csak arra kell ügyelni,

⁷ Én az SWI-Prolog „dialektust” használom. Az SWI-prolog értelmező programot ingyenesen lehet letölteni a <http://www.swi-prolog.org/> címről. A Prolog programozását a Blackburn et al. (2006)-ból lehet megismerni, aminek a <http://www.learnprolognow.org> oldalon ingyenesen elérhető az változata online elérhető.

hogy kis betűvel kezdődjenek: `vincent`, `robber`. A változók a Prologban mindig nagy betűvel kezdődnek, ezért is használtuk már a logika ismertetésében is nagy betűket erre a célra: `X`. A névből vagy változóból és predikátumból álló egyszerű állítások is ugyanúgy festenek, mint ahogy eddig megszokhattuk: `robber(vincent)` vagy `customer(X)`.

A tagadás `~`-ja helyett most a `not` kulcsszót használjuk, utána pedig zárójelben következik a tagadott állítás: `not(robber(vincent))`.

Az `∧`, `∨`, és `→` infix (azaz a két összekapcsolt állítás *között* megjelenő) logikai műveletek helyett az `and`, `or` és `imp` prefix kulcsszavakat használjuk, amiket vesszővel elválasztva a két összekapcsolt állítás követ zárójelben: `imp(robber(vincent), customer(X))` stb.

Az univerzális és az egzisztenciális kvantor `∀` és `∃` jele helyett az `all` és a `some` kulcsszavak használandóak, amiket zárójelben a változó és a változót tartalmazó állítás követ: `some(X, robber(X))` például azt fejezi ki, hogy létezik olyan individuum, aki rabló.

A rejtvényben szereplő négy állítás prologizálva a következő lesz:

- ```
(1") and(sapka(bruce), not(barna(bruce)))
(2") all(X, imp(nyuszi(X), piros(X)))
(3") and(not(roka(arnold)), not(sal(arnold)))
(4") all(X, imp(farkas(X), and(kesztyu(X),
 not(fekete(X)))))
```

#### 4. Miért nem elég ez a feladat megoldásához?

Most, hogy a rejtvény állításait számítógépes formára alakítottuk, a következő lépésünk a rejtvény megoldása felé az, hogy megadjuk a rejtvény világát, azaz a modellt is, amiben az állításokat értelmezzük.

A modellben szereplő individuumokat könnyű megadni, hiszen már eldöntöttük, hogy három lesz belőlük, legyenek tehát *a*, *b* és *c*. A nevekhez tartozó jelöleteket is könnyen elintézhethetjük, *arnold* jelölje az *a* individuumot, *bruce* a *b*-t, *sylvester* pedig a *c*-t.<sup>8</sup> A predikátumok jelölétéül szolgáló individuumhalmazok megadásánál azonban már problémába ütközünk, hiszen nem tudjuk például megmondani (most még), hogy a *nyuszi* tulajdonság mely individuumokra jellemző. A rejtvény nem úgy van ugyanis

---

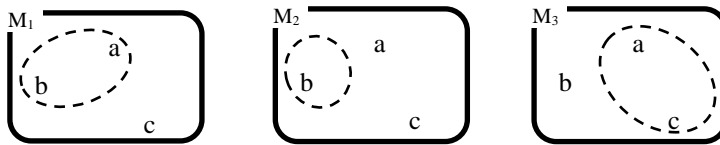
<sup>8</sup> A továbbiakban nem a számítógépes reprezentáció *Courier* betűtípusát használom, hanem a logikai formulák *Calibri*-jét, a predikátumlogikai kifejezések számítógépes megfelelőinek a megalkotását az olvasóra hagyom.

megfogalmazva, hogy „Ez itt a nyúl, a neve Arnold, és egy piros sapkát visel, ő pedig Bruce... És most mond meg, hogy van-e sapkája a nyúlnak!” Így túl unalmas lenne, nem is lennének kíváncsiak a megoldásra. Amikor arra vagyunk kíváncsiak, mi van a hegyen túl, akkor sem tudjuk a választ előre: lehet, hogy egy sárkány, de lehet, hogy Kazincbarcika.

## 5. Dinamikus szemantika

Nézzük meg akkor, hogy hogyan működik valójában a rejtvény, vagy általánosabban: hogyan ismerjük meg a világot, amikor információkat szerzünk (hallunk) róla!

Kezdetben, mármint a játék vagy a megismerés kezdetén, nem tudunk semmit a világról, vagy legalábbis hiányosak az információink. Nem tudjuk, hogy Bruce sapkát visel-e, lehet, hogy sálat. Több lehetőség is van, több lehetséges világ, az egyikben van sapkája van Bruce-nak, a másikban nincs. Ezek a lehetséges világok különböznek egymástól, de az egyikük olyan, mint a valódi (rejtvénybeli) világ. A következő ábrákon három lehetséges világot (modellt) ábrázolok, a modellekben a bekarikázott individuumok sapkásak:



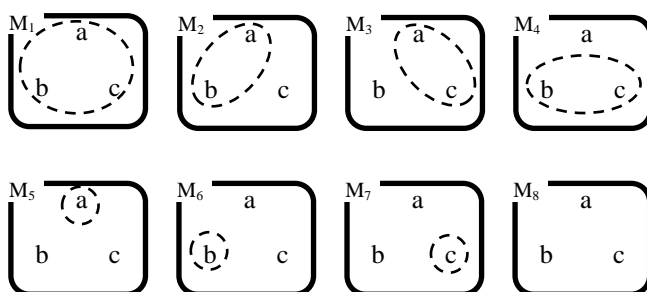
Ha most azt az információt kapjuk, hogy *Bruce-on sapka van*, akkor először is feltételezzük, hogy ez az információ igaz, vagyis az általunk megismerni kívánt világban igaz. Ha megnézzük, hogy a lehetséges világaink közül melyikben igaz ez az állítás akkor azt láthatjuk, hogy az első kettőben igaz, a harmadikban viszont hamis. Tovább haladva a megkezdett úton, minél több információt dolgozunk fel, minél többet tudunk meg a világról, a kezdeti sok-sok lehetséges világból egyre kevesebb olyan marad, amelyikben az eddigi összes állítás igaz. Végül egy olyan lehetségesvilág-halmazt kapunk, aminek az elemei többé-kevésbé rásimulnak a tényleges világra, vagyis az információk feldolgozásával egyre jobban megismerjük a tényleges világot.

A predikátumlogika korábbi használata során egy statikus modellt használtunk: ez itt a modell, ebben lehet értelmezni az állításokat, az állítások pedig vagy igazak, vagy nem. Most áttérünk egy dinamikus állításértelmezési stratégiára. Az állításainkat ugyanabban a predikátumlogikában fogalmazzuk meg, de nem csak egy modellben értelmezzük, hanem többen, a lehetséges világok mindegyikében.

Kezdetben adott a lehetséges világoknak egy  $\omega_0$  halmaza. Ezek közül az  $\alpha$  állítás igaz néhányban, ezek lesznek a lehetséges világok  $\omega_1$  halmaza ( $\omega_1 \subseteq \omega_0$ ). Egy következő

állítás feldolgozva ebből egy újabb  $\omega_2$  lehetségesvilág-halmazt kapunk. Egy állítás jelentését úgy írhatjuk le, mint azt a képességet, ami egy lehetségesvilág-halmazt megváltoztat.

A predikátumlogikában az állítások igazságértéke egy modellben végső soron csak a modellben levő individuumoktól és az állításban szereplő nevek és predikátumok modellbeli értelmezésétől függ. Ez azt jelenti, hogy ha az individuumokat adottnak vesszük, a nevek jelölését pedig állandónak, akkor a teljesen információhiányos  $\omega_0$  lehetségesvilág-halmazt a logikában használt predikátumok segítségével teljes mértékben meghatározhatjuk. Egy háromindividuumos modell esetében egy predikátumot feltételezve összesen nyolc lehetséges világ van, hiszen egy háromelemű halmaznak nyolc részhalmaza van, vagyis az adott egyargumentumú predikátumot nyolcféleképpen lehet interpretálni:



Minden egyes újabb predikátum a jelenlegi lehetséges világoknak újabb variánsait hozza létre, de ezek az újabb világok is egyértelműen meghatározhatóak.

Bár az eredeti Blackburn–Bos-féle prolog program nem volt képes megoldani nekünk a nyuszikás rejtvényt, de kis (nem olyan rettenetesen nagy) kiegészítéssel a célunknak megfelelővé tudjuk alakítani. A technikai részletekbe nem szeretnék belemenni, de mindenki kipróbálhatja a kész programot a <http://ling.bibl.u-szeged.hu/inqsem> oldalon, ahol egy webfelületen keresztül használhatjuk.<sup>9</sup>

A programban először is meg kell adnunk az individuumokat, vesszővel elválasztva:

<sup>9</sup> Eredetileg nem a nyuszis feladat megoldására készítettem a programot és a weboldalt, hanem az inkvizitív szemantika (Inquisitive Semantics, l. Ciardelli (2010), bővebben <https://www.illc.uva.nl/inquisitivesemantics/Home>) kipróbálására és illusztrálására.

**Individuums of the Universe**

a, b, c

Next

Ezután tetszőleges (Prolog!) logikai formulát írhatunk be, de mielőtt ezt megtennénk, először meg kell adnunk a formulá(k)ban szereplő neveket (a hozzájuk tartozó individuummal) és a predikátumokat. Egyenként, egymás után.

**Enter a formula!**

OK

**Add a new predicate (with the number of its argument(s))!**

nyuszi 1 ▼ OK

**Add a new name (with its denotation)!**

arnold a ▼ OK

Ekkor a program visszaadja a lehetséges világokat (indexnek nevezi őket):

There are 8 indices, in which 8 are valid.

|           |                |
|-----------|----------------|
| index: i7 | nyuszi=[c]     |
| index: i6 | nyuszi=[b,c]   |
| index: i5 | nyuszi=[b]     |
| index: i4 | nyuszi=[a,c]   |
| index: i3 | nyuszi=[a,b,c] |
| index: i2 | nyuszi=[a,b]   |
| index: i1 | nyuszi=[a]     |
| index: i0 | nyuszi=[]      |

Ha most beírunk egy Prolog logikai formulát, például hogy `nyuszi(arnold)`, akkor a visszaadott lehetséges világok halmaza ugyanaz lesz, mint korábban, de közülük már csak négy lesz olyan, ami kompatibilis az elhangzott információval: az i1, az i2, az i3 és az i4 jelű:

There are 8 indices, in which 4 are valid.

|           |                |
|-----------|----------------|
| index: i7 | nyuszi=[c]     |
| index: i6 | nyuszi=[b,c]   |
| index: i5 | nyuszi=[b]     |
| index: i4 | nyuszi=[a,c]   |
| index: i3 | nyuszi=[a,b,c] |
| index: i2 | nyuszi=[a,b]   |
| index: i1 | nyuszi=[a]     |
| index: i0 | nyuszi=[]      |

További állítások beírásával tovább szűkíthetjük a ténylegesen lehetséges világok körét, például a *Nem mindenki nyuszi* magyar mondatnak megfelelő  $\text{not}(\text{all}(X, \text{nyuszi}(X)))$  beírásával az i3-mal jelölt világ is kiesik.

## 6. Az elszabadult lehetséges világok megzabolázása

Most végre nekiállhatunk az eredeti rejtvényünk megoldásának!

Először is írjuk be, hogy három individuummal akarunk dolgozni, *a*-val, *b*-vel és *c*-vel, majd nevezzük el őket *arnold*-nak, *bruce*-nak és *sylvester*-nek. Ezután egyenként adjuk meg a rejtvényben szereplő tulajdonságokat, azaz az egyargumentumú predikátumokat: *nyuszi*, *farkas*, *roka*, *sapka*, *sal*, *kesztyu*, *piros*, *barna* és *fekete*.

Na most azok az olvasók, akik ezt a bekezdést olvassák, szóljanak azoknak, akik nem voltak restek kipróbálni magát a programot is, és addig nem akartak továbbolvasni, amíg be nem írták a kilenc predikátumot, hogy hagyják abba a várakozást, inkább csatlakozzanak hozzánk. Ugyanis a program így nem fog működni, valahol a sapka és a sál között elakad. Miért is?

Amikor a predikátumokat egyenként beírjuk, a program újraszámolja a lehetséges világok halmazát: a korábbi lehetséges világok mindegyikét megtöbbszörözi, hogy az új predikátum által megkívánt lehetőségeket is reprezentálhassa. Amíg egy predikátumot sem használtunk, csak egyetlenegy világ volt, amiben az individuumokat nem jellemeztük semmilyen tulajdonsággal sem. Az első predikátum megadásával  $2^3=8$  lehetséges világ jött létre, majd minden egyes új predikátum után megnyolcszorozódott a számuk. Mind a kilenc predikátum megadásával már  $8^9 = 134\,217\,728$  lehetséges világgal kell számolnunk. Százharmincnégymillióval! És ez nem egy bonyolult világ, csak három szereplője van és kilenc tulajdonsága. Ha négy individuummal kezdtünk volna, akkor minden új predikátum  $2^4=16$ -szorosára növelte volna a lehetőségeket. És

hol van ez még a *Trónok harca* szereplőgárdájához képest? Hányféleképpen lehetne elképzelni a *Trónok harca* világát? (Lenne olyan is, ahol a szereplők fele sárkány, a másik fele pedig törpe.)

Látható, hogy ezen a módon nem lehet megoldani a mi kis nyulas rejtvényünket, mivel bár a számítógépek mára már hatalmas számítási kapacitással rendelkeznek, nem várhatjuk el, hogy a vacak kis feladat miatt bekapcsolják az egyetemi számítóközpontban a szuperszámítógépeket. Azonban egy kis csalással – na jó, nem csalással, nevezzük inkább logisztikai optimalizálásnak – mégiscsak megtudhatjuk, van-e sapkája ennek a nyúlnak.

Amikor az egymással összefüggő nyuszi, farkas és róka predikátumokat bevezettük,  $8^3=512$  lehetséges világ keletkezett, azonban ezek többsége valójában nem is lehetséges: az 510-es indexű világban a  $c$  individuum egyszerre róka, farkas és nyuszi is, de róka még a  $b$  individuum is, az  $a$  pedig egyik sem.

|             |                                        |
|-------------|----------------------------------------|
| index: i510 | roka=[b,c]<br>farkas=[c]<br>nyuszi=[c] |
|-------------|----------------------------------------|

Ez ugye abszurdum. Hiszen *tudjuk*, hogy nem lehet valami egyszerre róka is és farkas is. Honnan is tudjuk?

Amikor az ilyen típusú rejtvények megfejtésének nekiállunk, feltételezzük, hogy a rejtvényben szereplő tulajdonságok mindegyike csak az egyik individuumra jellemző, és az összetartozó tulajdonságok kizárják egymást, vagyis hogy pontosan egy róka van, és a róka nem lehet egyúttal nyuszi is. Persze sálja attól még lehet neki, hiszen az egy más jellegű tulajdonság. Az ilyen jellegű tudásunkat, ami a feladatban nincsen explicit módon megfogalmazva, háttértudásnak nevezzük. Ha az ebben a rejtvényben impliciten megjelenő háttértudást is figyelembe vesszük, akkor a farkas–róka–nyuszi viszonylatban csak  $3 \times 2 \times 1 = 6$  logikailag lehetséges világgal kellene számolnunk, ha pedig mind a kilenc predikátumot vesszük, akkor  $6^3=216$  lehetséges világ marad a korábbi százharmincvalamennyi millióval szemben. De hogyan választjuk ki ezt a 216 világot, hogyan hozzuk létre?

Amikor egy lehetségesvilág-halmazt leszűkítünk egy állítás értelmezésével, akkor ezzel a világhalmazt két részre osztjuk: az egyik részükben igaz az állítás, ezek továbbra is lehetséges világok, a másik részükben viszont hamis. Amikor egy új predikátumot bevezetünk, akkor valamennyi világot megsokszoroztuk, azokat is, amelyek már csak elméletileg lehetségesek, az állításainkkal viszont nem kompatibilisek. Kordában tarthatjuk a figyelembe veendő lehetséges világok számát, ha az ilyen nem lehetséges lehetségesvilágoktól megszabadulunk. Ezt meg is tehetjük a webfelület *Drop impossible indices* gombja segítségével. Megnyomásával csak azok a világok maradnak,

amelyekben az eddigi állításaink mind igazak voltak. Az optimalizálás tehát annyit jelent, hogy még mielőtt az összes predikátumot bevezetnénk, közben állításokat teszünk a már meglevő predikátumainkat használva, és a nemkívánatos világokat eldobjuk. A lehetséges világok között szelektáló közbülső állításaink pedig a korábban említett háttértudásunk logikai megfelelői lesznek.

Az eddig kimondatlan háttértudásunkat legegyszerűbben ezekkel a szavakkal fogalmazhatjuk meg: *Pontosan egy nyuszi van. Pontosan egy róka van. Pontosan egy farkas van.* A gond ezzel csak az, hogy a pontosan egy kifejezést nehéz formalizálni. A három-individuumos világunkban e helyett a három mondat helyett a velük logikailag egyenértékű *Van legalább egy nyuszi/róka/farkas* és az *Ami nyuszi, az nem farkas* stb. mondatokat (6 darab) fogjuk formalizálni.

A *Van legalább egy nyuszi* mondat logikai megfelelőjével nincs különösebb gond:

(5')  $\exists X(\text{nyuszi}(X))$

(5'')  $\text{some}(X, \text{nyuszi}(X))$

Az *Ami nyuszi az nem farkas* mondatot továbbgyúrjuk, úgy parafrázáljuk, hogy *Nincs olyan, ami nyuszi is és farkas is:*

(6')  $\sim\exists X(\text{nyuszi}(X) \wedge \text{farkas}(X))$

(6'')  $\text{not}(\text{some}(X, \text{and}(\text{nyuszi}(X), \text{farkas}(X))))$

és így tovább, a farkasok és a rókák létezését is állítjuk, valamint a nyulak és a rókák, illetve a rókák és a farkasok transzgenetikus mutációit is megtiltjuk, és ugyanígy járunk el a ruhadarabok és a színek dimenziójában is.

Most már csak meg kell etetni a programunkat a megfelelő információkkal a megfelelő sorrendben! Az individuumok és a nevek rögzítése után először bevezetjük a *nyuszi* predikátumot (8 index), majd beírjuk az (5'') állítást, és eldobjuk a nem megfelelő világokat (marad 7 index). Folytatjuk ugyanezt a farkassal is (56 illetve 49 index), majd kijelentjük (azaz beírjuk) a (6'') állítást (12 index, jó úton járunk). Végül bevezetjük a *róka* predikátumot (96 index), eldobjuk azokat a világokat, amelyekben nincsenek rókák (84 index), majd megszabadulunk azoktól is, amelyekben valami róka is és nyuszi is (30 index), és azoktól is, amelyekben valami róka is és farkas is. Így végül megmarad hat lehetséges világ, ahogyan az előbb ki is számoltuk.

Az így megmaradt lehetséges világokból kiindulva ugyanezt a folyamatot végigcsináljuk a ruhadarabokkal (36 túlélő index) és a színekkel is (216 index). Igaz, a végén már egy kicsit sokat kellett várni, de ne legyünk telhetetlenek. Az elemzés közben a lehetséges világok számának a maximuma 3456.

## 7. Hogy is van ezzel a sapkával?

És most neki is kezdhetünk a rejtvény tényleges megoldásának, a feladványban szereplő négy állítás bevitelének. Az eddigiek után ez már rutinmunkának számít, először az (1"), majd a (2"), a (3"), végül a (4") állítást írjuk be, és mivel az minket nem érdekel, hogy milyen *nem* a rejtvény világa, a nem lehetségeseket eldobjuk.<sup>10</sup>

És ekkor láthatjuk, hogy mi a megoldása rejtvénynek!

|              |                                                                                                                     |
|--------------|---------------------------------------------------------------------------------------------------------------------|
| index: i583  | fekete=[c]<br>barna=[a]<br>piros=[b]<br>kesztyu=[a]<br>sal=[c]<br>sapka=[b]<br>roka=[c]<br>farkas=[a]<br>nyuszi=[b] |
| index: i3189 | fekete=[b]<br>barna=[a]<br>piros=[c]<br>kesztyu=[a]<br>sal=[c]<br>sapka=[b]<br>roka=[b]<br>farkas=[a]<br>nyuszi=[c] |

És ekkor azt látjuk, hogy két lehetséges világ maradt. Az egyikben, ami nálam az 583-as indexű, a *b* individuum, azaz Bruce a nyuszi, akinek piros sapkája van, a 3189-es

<sup>10</sup> A kevésbé türelmes olvasóknak: ha az individuumok megadása után a következő kódot írják be az *Enter a formula!* mezőbe, akkor egy lépésben letudhatják az egész elemzést – bár egy kicsit sokat kell írni:

```

history:addNewName(arnold,a)+addNewName(bruce,b)+addNewName(sylvester,c)+
addNewPredicate(nyuszi,1)+and(some(X,nyuszi(X)))+forgetImpossibleIndices+
addNewPredicate(farkas,1)+and(some(X,farkas(X)))+forgetImpossibleIndices+
and(not(some(X,and(nyuszi(X),farkas(X)))))+forgetImpossibleIndices+
addNewPredicate(roka,1)+and(some(X,roka(X)))+forgetImpossibleIndices+
and(not(some(X,and(nyuszi(X),roka(X)))))+forgetImpossibleIndices+and(not(some(X,
and(roka(X),farkas(X)))))+forgetImpossibleIndices+addNewPredicate(sapka,1)+
and(some(X,sapka(X)))+forgetImpossibleIndices+addNewPredicate(sal,1)+
and(some(X,sal(X)))+forgetImpossibleIndices+and(not(some(X,and(sapka(X),sal(X)))))+
forgetImpossibleIndices+addNewPredicate(kesztyu,1)+and(some(X,kesztyu(X)))+
forgetImpossibleIndices+and(not(some(X,and(sapka(X),kesztyu(X)))))+
forgetImpossibleIndices+and(not(some(X,and(kesztyu(X),sal(X)))))+
forgetImpossibleIndices+addNewPredicate(piros,1)+and(some(X,piros(X)))+
forgetImpossibleIndices+addNewPredicate(barna,1)+and(some(X,barna(X)))+
forgetImpossibleIndices+and(not(some(X,and(piros(X),barna(X)))))+
forgetImpossibleIndices+addNewPredicate(fekete,1)+and(some(X,fekete(X)))+
forgetImpossibleIndices+and(not(some(X,and(piros(X),fekete(X)))))+
forgetImpossibleIndices+and(not(some(X,and(fekete(X),barna(X)))))+
forgetImpossibleIndices+and(and(sapka(bruce),not(barna(bruce))))+and(all(X,
imp(nyuszi(X),piros(X))))+and(and(not(roka(arnold)),not(sal(arnold))))+
and(all(X,imp(farkas(X),and(kesztyu(X),not(fekete(X)))))+forgetImpossibleIndices

```



indexú világban pedig Sylvester a nyuszi, és nem sapkát visel, hanem egy piros sálat. Micsoda átverés! Nem is tudjuk meg, hogy van-e sapkája a nyuszinak?

Hát igen, a rejtvényből nem derül ki. Pedig a rejtvények nem ilyenek szoktak lenni. Rejtvényfejtéskor feltételezzük, hogy a rejtvényeknek mindig van megoldásuk, és mindig csak egy van, és hogy a megadott eszközök segítségével meg lehet oldani, mint ahogy a matematikai feladatgyűjteményekben levő feladatoknak is mindig van megoldásuk, általában valamilyen szép egész szám. A valódi világban azonban ez nem mindig van így.

## 8. További kérdések

Ez a rejtvény, illetve a szokásos megoldása helyett végigjárt út megmutatja nekünk, hogy ideális esetben milyenek képzelhetjük el az emberi kommunikációt, ami során a beszélgetőtársunk információkat közöl velünk arról a világról, amit ő ismer. A rejtvény, a megoldhatatlanságától eltekintve, egy ideális szituációt írt le. A megfogalmazott állítások egyértelműek voltak. Ismertek a játék szabályai, vagyis a háttértudásunk, és ezeket aránylag könnyen meg tudtuk fogalmazni. Előre meghatározott szereplők voltak, meghatározott nevekkal. És ami a legfontosabb: a rejtvényben elhangzó állítások mind igazak voltak.

A hétköznapi kommunikáció nem ilyen ideális körülmények között zajlik. Hogyan értjük meg a másikat, ha a hallott információ nem egyértelmű, homályos? Megfelelő-e a háttértudásunk, ugyanazt tudjuk-e, mint a beszélgetőtársunk, és tudja-e a társam, hogy én tudom, hogy ő mit tud...? Mi van akkor, ha egy új szereplő jelenik meg a társalgás világában, vagy ha egy szereplőről kiderül, hogy ő tulajdonképpen két személy, esetleg különbözőnek hitt személyekről kiderül, hogy nem is különbözőek? Hogyan kell a világot leírni, ha a beszélő véletlenül nem igaz állítást mond, esetleg szándékosan hazudik? Ha nem mond el mindent, amit tud, vagy épp ellenkezőleg, fölösleges információkkal áraszt el bennünket? Ha ellentmondásba keveredik, vagy visszavonja egy korábbi állítását?

Megannyi megválaszolendő kérdés. Sok hegy van még előttünk.

## Hivatkozások

Alberti Gábor 2006. *Matematika a természetes nyelvek leírásában*. (Segédkönyvek a nyelvészet tanulmányozásához 52.) Budapest: Tinta Könyvkiadó.

Blackburn, Patrick - Johan Bos 2005. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. (CSLI Studies in Computational Linguistics 1.) Stanford: CSLI.

- Blackburn, Patrick – Johan Bos – Kristina Striegnitz 2006. *Learn Prolog Now!* College Publications.
- Ciardelli, Ivano 2010. A first-order inquisitive semantics. In Maria Aloni – Harald Bastiaanse – Tiki de Jager – Katrin Schulz (szerk.) *Logic, Language, and Meaning: Selected Papers from the 17th Amsterdam Colloquium*. Berlin – Heidelberg: Springer-Verlag. 234–243.
- Partee, Barbara E. – Alice ter Meulen – Robert E. Wall 1990. *Mathematical methods in Linguistics*. (Studies in Linguistics and Philosophy 30.) Kluwer Academic publishers.