

## Csevegőrobotok. Az ITSy-Bitsy modell Chatbots. The ITSy-Bitsy chatbot model

KILIÁN Imre

IT Services Hungary, Pécs  
7621 Pécs, Nagy Lajos király útja 11.  
tel: +49 69 9731792500  
www.itsh.hu  
Imre-Zoltan.Kilian@t-systems.com

### Abstract

*The utilization of chatbots has a significant increase in the software industry in the recent years. They are used sometimes in the field of software support, but sometimes they are also used as a new user interface paradigm. Industry chatbots are usually built using the neural-network/deep learning technology. In the present article however, we write about a rule based solution, that follows the steps of the usual natural language processing pipeline: morphology, syntax, semantics... The model is developed for the English language, and its knowledge base is filled with some example data of A.A.Milne's Winnie-the-Pooh. It can therefore evaluate and answer some questions related to that worldlet.*

### Kivonat

*Csevegőrobotok az elmúlt években robbanásszerűen terjedtek el az ipari szoftverekben, akár az üzemeltetés/hibabejelentés területén, akár egy újszerű kezelői felület elemeként. Az ipari csevegőrobotok általában neurális hálós/mélytanulós technológiával készülnek. A jelen cikkben mégis egy szabályalapon készült csevegőrobot modellről számolunk be, amely híven követi a természetes nyelvek feldolgozásának szóalakelemzés-nyelvtani elemzés-tartalmi elemzés menetrendjét. A modell jelenleg angol nyelven működik, és a Micimackó világa van részben beprogramozva, vagyis erre vonatkozó kérdéseket tud kiértékelni és megválaszolni.*

**Kulcsszavak:** logikai programozás, természetes nyelvek feldolgozása, tudásábrázolás, csevegőrobotok

### 1. Bevezetés

A csevegőrobotok (*chatbotok*) fogalma egyáltalán nem új, még ha az elnevezés az is. A számítástechnikai bölcsőjénél, még Alain Turing tűzte ki célul a róla elnevezett Turing próbát teljesítő gépi eszközök létrehozását, amelyet azóta is a Mesterséges Intelligencia egyik alapdefiníciójának tartanak. Eszerint: ha egy zárt szobában két számítógépes terminál (két csevegőablak) van előttünk, amin keresztül cseveghetünk valakivel, és nem tudjuk megállapítani, hogy melyik ablakban válaszol egy másik hús-vér ember a kérdéseinkre, és melyikben egy gép, akkor bátran mondhatjuk, hogy a gépi eszközök intelligenciája elérte az emberi intelligencia szintjét [1].

A közcélú szoftverekben a csevegőrobot (chatbot) kifejezés mellett használatos még a *hangrobot* (*voicebot*) kifejezés is, amely valamilyen telefonos, vagyis hangátvitellel működő beszélgetés gépi lefolytatására képes. Hangrobotok alkalmazása egyrészt jelenti egyfajta *szövegfelolvasó* (*Text to Speech*), ill. *szövegjegyző* (*Speech to Text*) technológia alkalmazását, ami manapság már annyira kiforrott, hogy még a Windows 10 operációs rendszer részeként is megtalálhatjuk (magyar hangja: Microsoft Szabolcs). Másrészt viszont, legalábbis a felismerő oldalon egy lényegesen rugalmasabb, a beszélt nyelvre alkalmazott nyelvi elemzőt szükséges alkalmazni – és itt igen komolyan felvethető a *nyelv határainak* kérdése, méghozzá nemcsak írott, beszélt, de tematikus, szakterületi, képzettség

szerint, sőt korcsoport szerinti lehatárolás alapján is. (Hangsúlyozottan nem csak szókészletek közötti különbségekről van szó...).

A jelen írásban a hangrobotok felhasználásáról bővebben nem szólunk.



1. ábra  
A Turing-teszt

### 1.1. Ipari csevegőrobotok

Közcélú szoftverekben csevegőrobotokat rengeteg helyen használhatnak – és egyre több helyen használnak is. A megoldásra legtöbbször *alternatív kezelői felületként* gondolunk: ha a felhasználó számára bármi okból nehéz a megszokott GUI elemeken, menükön történő keresztülnavigálás, akkor kézenfekvő, a csevegő technológia bevetése.

A feladat jellemző megoldása: nagyobb cégeknek saját csevegőmotorjuk van, ami kiszolgálóként van megvalósítva. Ezek többcélúan tesztre szabhatók, a szókészletük, a fogalomrendszerük, a szokásos adatbázisokba és/vagy szolgáltatásokba történő leképezésük pedig rugalmasan konfigurálható.

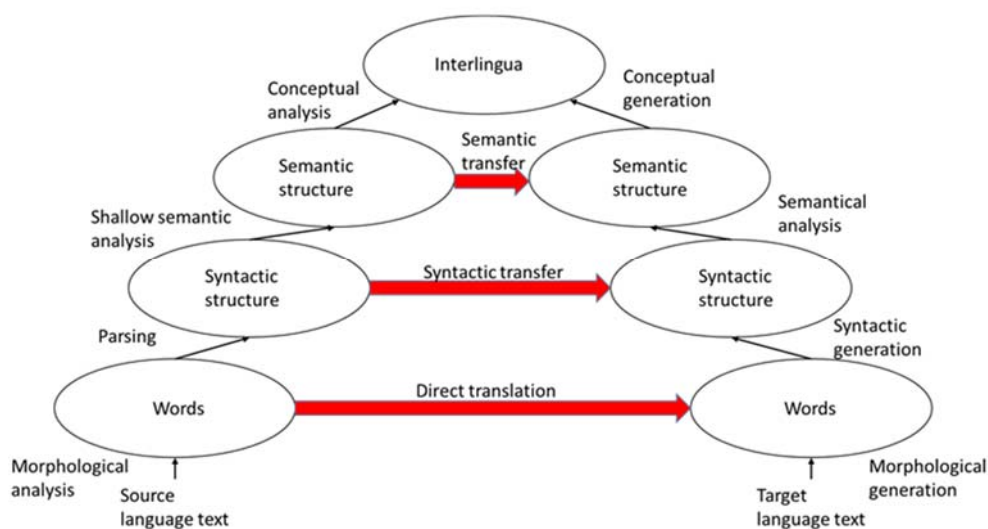
## 2. csevegőrobotok felépítése

A csevegőrobotok felépítése természetesen nagyban függ az elérendő céltól. A legegyszerűbb csevegőrobotok lényegileg egy *párbeszédes hálón* viszik keresztül a felhasználót úgy, hogy ékes (vagy épp éktelen) nyelven megfogalmazott kérdésekre a felhasználó rövid és tömör válaszokat ad, rendszerint valamiféle kötött kifejezéskincsből választ ki, vagy valami párbeszédmezőbe gépel be egy elemet. Úgy gondoljuk, hogy az ennyire leegyszerűsített megoldás ma már igénytelennek számít, (ettől persze még lehet hasznos!), ezért kívül esik a vizsgálódásainkon.

### 2.1. Weizenbaum ELIZA robotja

Az 1966-ban Joseph Weizenbaum által készített ELIZA robot [2] a fentieknél csak kicsit volt bonyolultabb. A robot egy pszichiáter szerepét játszotta, és a páciens begépelte mondataiból szövegszerű átalakításokkal kérdéseket csinált úgy, hogy az efféle átalakítási mintákból véletlenszerűen választott egyet. Az angol nyelv különösen alkalmas az „I am” □ „You are” jellegű átalakításokra. Például ha a páciens a következő mondatot gépelte be: „I am deprived today.”, arra a következő lehetséges választ adta: „Why do you think you are deprived today?” (Ma levert vagyok. Miért gondolja, hogy ma levert?)

Modern csevegőrobotok felépítéséhez célszerű a gépi fordítás során is használni Vauquois háromszögből kiindulni [4]. Eszerint a természetes nyelvek feldolgozása során a következő lépéseket különböztethetjük meg.



2. ábra  
A Vauquois-háromszög

## 2.2. Szóalakelemzés és nyelvi elemzés

A nyelvi feldolgozás első lépése a *szóalakelemzés*. A szóalakelemzés a bemenő mondat szavait (és általában bármilyen, szöként felfogható egységet) annak részeire, előképzőkre (prefix), szótőre, utóképzőkre (suffix), jelekre, ill. ragokra bont fel. Az ELIZA angol nyelven működött, így ez a lépés átgortható volt. Az ITSy-Bitsy elsőként szintén angol nyelvre készült, ugyanezzel a szándékkal: hogy első nekifutásra ne kelljen a szóalakelemzéssel bajlódni.

Egy *nyelvtani elemző (parser)* kimenete az elemzett mondat *elemzési fája*. Az elemzési fa a saját szemszögünkből azért kellemes adatszerkezet, mert egyrészt a Prolog megvalósítási nyelven természetes módon ábrázolható, másrészt az elemzési fából *szöveget generáló* modul nem csinál mást, mint lényegileg az elemzési fát járja körül.

Egy az ELIZÁhoz hasonló képességű robot könnyedén felépíthető a fent említettekre, az elemzőre és a szöveggenerálóra építve, egy harmadik összetevő, a *fa-transzformátor* közbeiktatásával. Ez utóbbi ugyanazt csinálja, mint az ELIZA, de szövegszerű keresés/cseré helyett az elemzési fára illeszkedő fa-minta esetén rögzíti az eredmény-fa szerkezetét.

Pl. az „invertál” szabály leírhatja, hogy a bemenő mondatban felismerhető „sg/1” (egyes szám első) szerkezetet „sg/2”-re cseréljük és fordítva. Ennél csak kicsivel bonyolultabb szabályok szükségesek az ELIZÁhoz hasonló átalakításokhoz, de hasonlóan leírhatók a nyelvtanórákról ismert különböző, igeidők szerinti vagy aktív-passzív stb. átalakítások is.

A leírt technológiát *szintaktikus transzfernek* vagy magyarosítva *nyelvtani átalakításnak* is nevezhetjük, amire építve akár egyszerűbb természetes nyelvi fordítóprogramok is létrehozhatók.

A nyelvtani átalakításos alapon működő robotok használhatósága mindazonáltal csekély, mert az átalakításba nemigen lehet külső adatforrásokat, vagy más, adat- ill. információtartalomra vonatkozó lekérdezést integrálni. Az átalakítás csupán formális, különösebb gyakorlati haszon nélkül.

## 2.3. Az ITSy-Bitsy nyelvi elemzője

Az elemző alulról-felfelé irányban, *nemdeterminisztikusan* működik, és így lényegileg a Cock-Younger-Kasami elemzőalgorithmus Prologra alkalmazott, *mélységi, visszalépéses keresési stratégiával* megvalósított változata. Az elemző megvalósítása komolyan épít a szerző korábbi munkájára, a Con-ralogra, ami a Prologhoz készített, előrehaladó módon következtető előfordító, így a Prologgal az illeszkedés tökéletes [3].

Az alulról felfelé működésmód miatt az elemző a szavak és kifejezések összes lehetséges elemzését előállítja. Ezek közül egyesek a további elemzési lépések során elhálnak, mások viszont részt

vesznek a mondat teljes elemzési fájának felépítésében. Mint minden nemdeterminisztikus elemzőalgoritmusban, itt is két gond merülhet fel. Egyrészt az elhaló elemzési irányok és ágak előállítására felesleges, de az előállítás során ezt korábban még nem mindig tudjuk eldönteni. Másrészt a nem eléggé körültekintően megírt elemző alternatív elemzések nagy tömegét állíthatja elő, ami szintén felesleges lehet, és esetleg nem is tudjuk, hogy melyiket fogadjuk el belőlük érvényesnek. Persze léteznek többféleképpen elemezhető (és értelmezhető) mondatok, amelyek esetében a többszörös eredmény nemigen megtakarítható.

Az elemzések előállításánál az alternatív elemzések nem látnak át egymás részeredményeire, ezért nem tudunk különféle heurisztikus kikötéseket tenni, hogy pl. egy adott fa-mintára illeszkedően a lehető legnagyobb szövegszeletet fogadjuk el. (Másképp fogalmazva: egy adott szövegszeletre illeszkedő lehető legegyszerűbb fa legyen a nyerő.)

#### 2.4. *Állapottal rendelkező és állapotmentes csevegőrobotok. Párbeszédelemzés*

Szoftverobjektumok kezelési és szoftverösszetevők összekapcsolási módjaihoz hasonlóan itt is célszerű különböző megoldásokat megkülönböztetni, ami a *párbeszédelemzés* kérdéskörével is összefügg.

- A legegyszerűbb, *állapotmentes* megoldás csupán természetes nyelvű kérdések megválaszolására alkalmas, amelyek egymástól függetlenek, és ugyanazon kérdésre mindig ugyanazt a választ kapjuk, azt a rendszer előlétele nem befolyásolja. Ez a megoldás nem tartalmaz semmiféle párbeszédelemzést, és nem képes a nyelvi hivatkozások (pl. névmások) feloldására sem.
- Az ennél igényesebb megoldások már *állításokat (kijelentő mondatokat)* is feldolgoznak, és ha az új információ még nem volt ismert, akkor azzal bővítik a saját ismereteiket (tanulnak). Ez már mindenképpen állapotallal rendelkező megoldás, az új ismerethalmazt pl. az aktuális felhasználóhoz lehet kötni, és később külön felhatalmazás alapján lehet az állandó ismeretek tárában elhelyezni, ill. esetleg törölni. Ez az adatbázisok commit-rollback mechanizmusához hasonló működésmódot jelent.
- Az előző megoldásban még csak a felhasználó az egyetlen állapotinformáció. Ha a párbeszédfeldolgozással kapcsolatban további – esetleg rugalmasan meghatározható állapotinformációk tárolása is szükséges, amelyek esetleg mondatról mondatra is változhatnak, akkor lényegileg a *párbeszédelemzés* területére érkezünk. Itt már muszáj a korábban hallott információk tárolását, és a rájuk történő hivatkozások (pl. névmások) kezelését is megvalósítani. Ez a mondatok felett egy a környezetfüggetlen nyelvekéhez hasonló, egymásba ágyazott dobozrendszer felépítését javasolja, amelyben egyes nyelvi elemek újabb, beágyazott párbeszédkörnyezetet nyithatnak meg. Mások azonos szinten nyitnak újabb környezetet, megint mások pedig lezárják a nyitott környezetet [5].
- Az eddigi stratégiai szintek csak az ismeretek bővítéséről beszéltek, vagyis monoton növekvő tudásállományt feltételeztek. A további bonyolultsági szint már a tudáselemek *törlését* is megengedi. Egy efféle megoldás az összes olyan gonddal szembeesül (pl. nem ismert távoli hivatkozások a törölendő elemre), amelyek általában az ipari szoftverek terén a törlési művelet megvalósításakor felmerülhetnek.

Az ITSy-Bitsy a jelenlegi megvalósítási szinten állapotfüggetlen, a lekérdezéseken túl képes primitív állítások feldolgozására is. A moduláris felépítése természetesen lehetővé teszi később párbeszédelemző réteg beiktatását is.

#### 2.5. *Átalakítás logikai nyelvvé*

Az átalakítás egy lehetséges iránya, ha az elemzési fából egy *logikai nyelv* fáját hozzuk létre. A logikai nyelv kiválasztásakor egyfajta alkat kell kötnünk: nagyon magas szintű logikai nyelvnek nagyon nagy a kifejező ereje, vagyis igen tömören tudjuk a mondanivalót megfogalmazni. Ám ezek a nyelvek *eldönthetetlenek*, vagyis nem létezik rá véges idő alatt garantáltan lefutó eldöntő algoritmus. Alacsonyabb szintű nyelvek (pl. az ítéletkalkulusé) eldönthetők lehetnek, de rendkívül szöszátyárak, és a gyakorlatban használhatatlanok is.

A logikai alakba történő átalakítás nagymértékben függ a mondatban használt *igemódtól*, ami a mondat módjának is tekinthető.

A *feltételes módtól* ezúttal eltekintünk, mert az az angol nyelven csupán feltételes kötőszóval összekapcsolt, és időbeli viszonyba állított kijelentő tagmondatokkal van megvalósítva.

*Felszólító mondatokat* a csevegőrobotok valami azonnali tevékenységre utaló parancsként. Közvetlen tevékenység végrehajtására egy számítógépnek különböző ügyfélszolgálatok esetén lehet sok lehetősége. Egyébként a szokásos irodai munkán belül pl. „töröld ki” (ezt a fájlt, ezt az üzenetet), „fizesd ki” (például egy számlát), netán „hívd fel” (pl. feltárcsáztathatjuk valamelyik ismerősünket vagy partnerünket, vagy „nyomtasd ki” (valamelyik dokumentumot), és hasonlókon túl esetleg a program bezárása lehet egy megvalósítandó hétköznapi parancsfunkció.

Igazán érdekes a kijelentő és a kérdő mondatok megvalósítása. Ezeket ún. *konjunktív kérdésekké*, vagyis elemi relációk konjunkciójává alakíthatjuk át, amelyet a háttérben levő tudáskezelő rendszer értékel ki és/vagy válaszol meg.

Az alábbiakban állítások és kérdések konjunktív logikai lekérdezőnyelvvé történő átalakítását mutatjuk be részletesen egy kiterjesztett elsőrendű logika felett.

### 2.6. *Modális logikai modell*

Logikai alapú következtető rendszerek legalapvetőbb kezelési gondja a logikai rendszerek *ellentmondás-mentessége*. Márpedig a valóságos élet tele van ellentmondásokkal: ami az egyik társaság számára örömnapp, az a másikkak számára gyásznap. Az ellentmondások kezelésének egyik útja a *modális logikák* bevezetése, ahol *világocskákat* (*környezeteket*) határozzuk meg. Az ellentmondás-mentesség csak egy világocskán belül követelmény, a világok között már nem.

A világocskák persze lehetnek *világocskahalmazok* is, amit más terminológiával *operátoroknak* is neveznek. A jogi normatívák pl. az ún. *deontikus* világocska-halmazban értelmezhetők, amely a TILOS, MEGENGEDETT, VÁLASZTHATÓ, KÖTELEZŐ világocskákat (operátorokat) tartalmazza. Az egyes világocskák között a következtetések végzéséhez különféle logikai axiómákat szokásos megállapítani.

Világocskákat az *episztemikus-doxasztikus* logika alapján *hozzárendelhetünk gondolkodó ügynökökhöz* (*ágensekhez*) is, pl. TUD (Micimackó), HISZ (Malacka). A világocskák egymásba is ágyazhatók, pl. HISZ (Malacka, TUD (Elefánt)). Az egymásba ágyazott világocskák között gyakori az *öröklés*, mint logikai axióma használata, a világocskák globális gyökéreleme ilyenkor az általánosan elfogadott/megdönthetetlen állításokat tartalmazó világ [5], [6].

A mondatok időparaméterét – mint a mondat egészére vonatkozó információt – szintén felfoghatjuk modális környezetként, amit külön dimenzióként együtt alkalmazhatunk az egyéb világocskákkal. Személetes ábrázolásmódban az egyéb világocskák „szőreiként”, vagyis beágyazott apró *levél-világocskákként* érdemes őket ábrázolni, és a következtetések elvégzésére itt is rögzíthetők különféle axiómák [7].

### 2.7. *Kétszintű logikai modell*

A hétköznapi élet mondatai és kérdései sajnos nem mindig elsőrendűek. Az efféle problémák kezelését ún. *kétszintű tudásábrázolással* oldhatjuk meg (reifikáció). Ez azt jelenti, hogy a *példányszintű / adatszintű* tudáselemeken (pl. Gipsz Jakab mikor hol volt, mit csinált) kívül *modellszintű* tudáselemeket (tudásszegmenst) is tartalmaz. Ez a példányszinten használt fogalmakra: osztályokra, tulajdonságokra és relációkra vonatkozó általános információkat, pl. az értelmezési tartományait, ill. azok egyéb összefüggéseit tárolja.

A kétszintű tudásábrázolás haszna, hogy a másodrendű kérdésekre is választ tud adni. Pl. a „Mi a különbség Malacka és Zsebibaba között?” kérdés esetén egy olyan logikai kifejezés generálható, amely egyrészt modellszinten megkeresheti a két állatkára csak külön-külön vonatkozó tulajdonságokat vagy relációkat (pl. a *vanErszénye/hasPouch* reláció értelmezési tartománya Malackát biztosan nem tartalmazza). Másrészt a közös tulajdonságokra és relációkra vonatkozólag megkereshetők azok, amelyeknek az értéke különbözik. Például a *hasKeeper/vanGazdája* tulajdonság értéke mindkettőjükre „Christopher Robin”, míg a *hasResidence* reláció értéke különböző. (Mint tudjuk, Malacka a saját házában lakik, „Tilos az Á” név alatt, míg Zsebibaba az édesanyja, Kanga házában lakik.) A leírt megoldás a cikk írásának időpontjában (2020 nyara) még nem működött.

A modelladatbázis ezen túl lehetőséget adhat az elemzett mondatok *statikus szemantikus* ellenőrzésére, és esetleg a felesleges elemzések szűrésére is. (pl. „Színtelen zöld eszmék dühödten alsznak.” Chomsky)

### 2.8. Kijelentő és kérdő mondatok logikai modellje

Igei kijelentő mondat modellje egyszerű (a Prologhoz hasonló logikai nyelven):

alany(...), argumentumok(...) []állítmány(...)

...vagyis minden olyan logikai változóértékre, ami megfelel az alany megkötéseinek, ÉS minden olyan változóértékre, amely az egyéb igei vonzatoknak (argumentum) is megfelel, ugyanezen változóértékekre teljesül az igei állítmánynak megfelelő logikai kifejezés is.

Névszói kijelentő mondatok ettől csak annyiban különböznek, hogy az állítmány nem ige, hanem névszó, tehát megfelelő logikai kifejezést is a névszói szerkezetekhez hasonlóan képezzük.

Névszói szerkezetek átalakítási mintája az ún. *összegző (kumulatív)* értelmezés szerint:

jelző<sub>1</sub>(X), ..., jelző<sub>N</sub>(X), köznévfő(X)

*Tulajdonnevek* egy egyedi azonosítóba képződnek le, és semmilyen egyéb feltételre nincs szükség. Az azonosító az őt tartalmazó legszűkebb osztályon belül egyértelmű, vagyis ha konkrét osztálypéldányról van szó, akkor az osztálynévvel is címkézni kell őket.

Pl. a „Malacka a Százholdas Pagonyban lakik.” állításból az alábbi logikai (tény) -állítás keletkezik.

live(piglet#pig, hundredAcreWood) .

...ahol a Malacka címkézését láthatjuk. A Százholdas Pagony a jelen modellben azért nincsen címkézve, mert nem a wood osztály egy eleme, hanem csupán egy postai címhez hasonló adatelem.

Az általános modellhez képest természetesen kivételek is megfogalmazhatók, pl. a „szétszórt bölcsészlány” kifejezés inkább önmagában is egy következtetés (szétszórt(X) :- lány(X), bölcsész(X)) [8]

*Kijelentő mondatok* átalakítása egyszerű esetben (ha csak példányokra hivatkozunk) tényállításokat eredményez. Ha az alany és az esetleges igei argumentumok leírása bonyolultabb (pl. kvantorokat, birtokviszonyt, stb. is tartalmaz), akkor a logikai alak egy Horn-klóz, vagyis egy Prolog szabály lesz. Annak vizsgálata után, hogy az új tényállítás vagy szabály már létezik-e, dönthetünk annak felvételéről. További érdekes kérdés, hogy *melyik világocskába* vegyük fel az új ismeretet. Ezt csak stratégiai megfontolások dönthetik el. Egy *bizalmatlan* stratégia (effélet játszik pl. egy bíró a bíróságon) minden információt a kijelentést tevő egyén világocskájába helyez, míg egy *hiszékeny* stratégia mindent a gyökérvilágba.

*Eldöntendő (yes/no) kérdő mondatok* esetében a mondat logikai alakját kiértékeljük, és a logikai eredményt visszaadjuk, ill. generálunk egy rövid helyeslést (yes, it is) vagy tagadást (no, she doesn't).

### 2.9. Kiegészítendő mondatok logikai modellje

*Kiegészítendő kérdésekben* már nyelvtani diszkontinuitás is megfigyelhető: a kérdőszó gyakran egy igei vonzat szerepét játssza, mégis rögzítetten a mondat elején találjuk. A kérdőszóból logikai változó lesz, az egész mondat pedig olyan logikai kifejezéssé fordul le, amely a változót is tartalmazza. A kérdés kiértékelése során a változó értéket kap, ez lesz a kérdésre adandó válasz. A kiegészítendő kérdés modellje tehát:

X^mondat(X) .

...ahol X a kérdőszónak megfelelő változó, mondat(X) pedig a teljes mondat logikai alakja, amely a változót biztosan tartalmazza.

### 2.10. Kvantorok és gyűjtőfüggvények kezelése

*Kvantorok* az elemzett nyelvben előfordulhatnak explicit módon is, de több ösztönös vagy rejtett nyelvi kifejezés is gyakran csak kvantorokkal vagy általánosabban *gyűjtőfüggvényekkel* értelmezhető. A logikai alakba alakító modul jelenlegi állapotában a következő gyűjtőfüggvények kezelése van megvalósítva.

- Egyes számú, de a nem konkrét főnévi csoport esetén az alábbi egyszerű kérdés kerül végrehajtásra, a Prolog kiértékelési módszerének megfelelően. Vagyis, ha az ismeretlen változónak több lehetséges értéke is volna, akkor visszalépéssel előállítja őket, pl. a „Who is Baby Roo’s male friend?” kérdésre válaszolva.

`X:=X^COND`

- Többes számú főnévi csoport esetén az eredmény egy gyűjtemény/egy halmaz. Az ilyen kérdésből létrehozott lekérdezés (ld. alább) a COND feltételnek megfelelő X változókat gyűjti össze a LIST változóba, pl. a „Who are the male friends of Piglet?” feltevése esetén.

`LIST:=each(X^COND)`

- Ha határozott névelőt használunk egyes számban, akkor arra utalunk, hogy a körülírt dologból egyetlenegy létezik. Az ilyen kérdésből létrehozott lekérdezés az *egyetlen, a feltételeknek megfelelő változóértéket* adja vissza. Ha több ilyen volna, vagy egy sem volna, a kiértékelés sikertelen lesz. Ilyen pl. a „Who is the male kangaroo?” kérdés (hiszen csupán Kangáról tudjuk, hogy nőnemű, hímnemű kenguru a világocskában nincs).

`VALUE:=only(X^COND)`

- Ha valakiknek vagy valamiknek a számára kérdezzük rá, akkor a feltételnek megfelelő *gyűjtemény számossága* érdekel. Ez az alábbi lekérdezéssel lehetséges, pl. a „How many male friends does Piglet have?” kérdés eredményeképpen.

`NR:=count(X^COND)`

- *Helyhatározói kérdésekre* van egy további érdekes beépített feldolgozás. Ha ugyanis több szereplőre vonatkozólag valami helyhatározós kérdés tennénk fel, (pl. hogy hol laknak), akkor a földrajzi egész-rész reláción (`hasGeopart/2`) keresztül a legközelebbi közös lakóhelyet kapjuk válaszul (`lca`: least common ancestor). A kérdés általános alakjában paraméterül adjuk a relációt magát is, ld. alább.

`X:=lca(X^CALL, RELATION)`

Például, a „Where do Piglet and Owl live?” kérdésre a „HundredAcreWood” választ kapjuk, míg a „Where do Kanga and Baby Roo live?” kérdésre a „kangasHouse” a helyes válasz.

### 2.11. Kapcsolódás konkrét adatbázisokhoz vagy szolgáltatásokhoz

A szoftver tartalmaz egy csatolómodult, amelyben a konkrét Prolog ontológiához történő kapcsolódás van leírva: vagyis esetleges Prolog átalakító-szabályokat lehet/kell itt megadni. Külső adatbázisok vagy szolgáltatások esetén meg kell valósítani az ontológiát kiterjesztő műveleteket is (vagyis ami lekérdez, bővít, töröl egy olyan tudáselemet, amely külső eszközön vagy szoftveren áll rendelkezésre).

## 3. Értékelés és további munkák

Az ITSy Bitsy a jelen állapotában egy állapotmentes csevegőrobot, amely angol nyelven működik. A moduláris felépítésből következőleg tetszőleges más elemző, netán más logikai alakra hozó modul is bekapcsolható, és más logikában másféle kiértékelés is használható. Annak sincs akadálya, hogy valamelyik modult másféle technológiára (pl. mélytanulásosra) cseréljük. [9]

A robot Prolog nyelven íródott, az SWI-Prolog rendszer környezetében [10]. A kiszolgáló-szerű működés úgy valósítható meg a legegyszerűbben, hogy az SWI-Prolog egyben webserverként is konfigurálható, vagyis igen könnyű a segítségével pl. REST kiszolgálót megvalósítani. Ebben az irányban eddig egyetlen sikeres kísérlet történt.

A rendszerrel lényegileg végigjártuk a csevegőrobot készítés főutcáját. Létrejött egy műszakilag teljes értékű, és minden fontos technológiai lépésre kiterjedő feldolgozási lánc, ami jelenleg a kiegészítendő kérdések feldolgozását tudja a legtökéletesebben elvégezni (és megválaszolni). Ezt konkrét helyzetekhez azonban többféle irányból is illeszteni kell.

- ha az angoltól eltérő nyelvről van szó, akkor a jelenlegi programfelületek tiszteletben tartásával meg kell írni egy elemzőt, (esetleg még szóalakelemzőt is)...

- ha a jelenlegi, kb. A2 nyelvi szint nem elégséges, akkor elsősorban a szótár fejlesztésével, másodsorban esetleg az elemző további fejlesztésével el kell érni a megcélzott nyelvi szintet. A szótár rugalmasan bővíthető: sikeres kísérlet történt a B2 szintet elérő szótár alkalmazására
- az általános célú szótárakat bővíteni kell a megcélzott szakterület kifejezéseivel
- létre kell hozni egy ontológiát, amely a megcélzott szakterület különleges fogalmait tartalmazza, és össze kell kapcsolni egy általános célú, ún. „csúcs-” ontológiával, hogy a hétköznapi fogalmak felett is tudjon következtetni (pl. tudja azt, hogy az egyes hónapok milyen évszaknak felelnek meg, vagy a hét napjai hogyan következnek egymás után...).

Mindazonáltal még a jelenlegi helyzetben is egy sor hasznos továbbfejlesztési lehetőség nyílik:

- „csúcsontológia” integrációja hiányzik. Jelenleg csupán, a Micimackó világoeska legfontosabb osztályai, példányai és szereplői vannak modellezve.
- a logikai alakra hozó modul félkész, mert csak a „főutcat” ismeri: ezernyi apró nyelvi fordulat van, amit nyelvfüggetlenül vagy akár az adott nyelv környezetében is az általánostól eltérően kell logikailag értelmezni
- a logikai alakra hozó modul átalakítása jegyszerkezetűvé
- valószínűleg eléggé egyszerű feladat volna egy adott szöveglejegyző vagy felolvasó szoftver illesztése. Ez a hangrobotok (voicebot) irányába visz el, de ahhoz a használt nyelvi elemző már biztosan nem elégséges. Egy beszélt nyelv megértése lényegesen bonyolultabb feladat az írotténál.

A cikkben leírt kezdeményezés lehet ígéretes, de a további életét teljesen biztosan befolyásolja a *külső érdeklődés*: az a segítség, amivel legalább egy életszerű és életszagú mintaalkalmazás összehozható. Természetesen – magyar környezetben – eléggé fontos volna a magyar elemző létrehozása és beillesztése is...

#### 4. Hivatkozások

- [1] Corydon Ireland: Alain Turing at 100, *The Harvard Gazette, Harvard University, September, 2012*, <https://news.harvard.edu/gazette/story/2012/09/alan-turing-at-100>, elérés: 21-07-2020)
- [2] Weizenbaum, Joseph (January 1966): ELIZA--A Computer Program for the Study of Natural Language Communication Between Man and Machine ([http://www.universelle-automation.de/1966\\_Boston.pdf](http://www.universelle-automation.de/1966_Boston.pdf), elérés: 21-07-2020).
- [3] Kilián, I.: Conralog: a Prolog conform forward-chaining environment and its application for dynamic programming and natural language parsing, *Acta Universitatis Sapientiae, 8-1, 2016. (pp.41-62)*.
- [4] Vauquois, B. (1968). A survey of formal grammars and algorithms for recognition and transformation in mechanical translation. *IFIP Congress (2)* (pp. 1114-1122)
- [5] Alberti, G. 2009. ReALIS: An Interpretation System which is Reciprocal and Lifelong. *Workshop 'Focus on Discourse and Context-Dependence' (16.09.2009, 13.30-14.30 ÚvA, Amsterdam Center for Language and Comm.)*,
- [6] Kilián, I.: ReALIS: egy többszereplős, episztemikus rendszer Prolog modellje *Erdélyi Magyar Műszaki Tudományos Társaság, SzámOkt 2012. konferencia kiadványa, Kolozsvár, pp. 276–281, 2012*
- [7] Ruzsa Imre: Klasszikus, modális és intenzionális logika *Akadémiai Kiadó, Budapest, 1984*
- [8] Alberti, G.: ReALIS: Interpretálók a világban, világok az interpretálóban. *Akadémiai Kiadó, Budapest, 2011.*
- [9] Wei Wu-Rui Yan: Deep Chit-Chat: Deep Learning for Chatbots, *Conference on Empirical Methods in Natural Language Processing, Brussels, 2018.* <http://www.ruiyan.me/pubs/tutorial-emnlp18.pdf>, elérés: 05-12-2020)
- [10] J. Wielemaker: An overview of the SWI-Prolog programming environment, Proc. 13-th International Workshop on Logic Programming Environments, pp.1-16. ed: F. Mesnard, A. Serebenik, Katholieke Universiteit, Leuven, Belgium, 2003.