Általános célú természetes nyelvi elemzőprogramok létrehozásának tapasztalatai

Experiences with creating natural language parsers

KILIÁN Imre

Deutsche Telekom Systems Solutions Hungary Kft, Pécs
7621 Pécs, Nagy Lajos király útja 11.
tel: +49 69 9731792500
www.deutschetelekomitsolutions.hu
Imre-Zoltan.Kilian@t-systems.com

Abstract

The immediate target of general-purpose natural language parsers is not to answer some question referring to a narrow and fixed topic, but to perform general purpose parses, to produce appropriate parse trees, which are transformed later to an evaluable logical query language. In the background there is an ontology, which is basically a collection of logical knowledge with the definition of common and specific concepts. This requirement demands the runtime configurability of the logical and the lingual information (that is vocabulary), also dynamically, in runtime. On the other hand, testing the parser requires a special set of lingual „benchmark" sentences. Testing also requires the evaluation of the efficiency of parsers, and the availability of appropriate measurement procedures. The article discusses these experiences, having gained by solving these problems in the production of three natural language parsers (American Sign Language, English, German).

Kivonat

Az általános célú elemzőprogramok célja nem valamilyen rögzített témájú közlés megértése és megválaszolása, hanem általános témájú nyelvi elemzés végzése, majd elemzési fa létrehozása, amelyet egy másik programmodul alakít át valamilyen kiértékelhető logikai lekérdezőnyelvvé. A kiértékelés hátterében egy ontológia áll, ami egy logikai tudásállomány, általános és a témába vágó fogalmak definíciójával. Mindez egyrészt a logikai és nyelvi tudásállomány (lexikon, szótár) konfigurálhatóságát követeli meg, lehetőleg futásidejű, dinamikus szinten is. Másrészt az elemzőprogram tesztelése speciális nyelvi „benchmark" mondatkészlet, a kiértékelése pedig megfelelő hatékonyságmérő eljárások létrehozását követeli meg. A cikk ezekről a problémákról ír három természetes nyelvű elemző (amerikai jelnyelv, angol, német) megírásának tapasztalataira építve.

Kulcsszavak: logikai programozás, természetes nyelvek feldolgozása, tudásábrázolás, ontológia

1. BEVEZETÉS

A szerző 2017-ben írt közleménye a „Szabályvezérelt nyelvi elemzők természetrajzáról" címmel jelent meg [1]. A megjelenéskor a tapasztalatokat számtalan tesztprogram és lényegileg egyetlen, az American Sign Language jelnyelvet elemző program sikeres létrehozása alapján gyűjtötte. Az elemző egy angol generátorral együtt az SWI-Prolog nyelven és fejlesztési környezetben készült [6]. Az elemzett nyelvek halmaza azóta bővült: a szerző időközben készített egy angol elemzőt, amelyet beépített az ITSy-Bitsy csevegőprogramba [2], majd egy német elemző elkészítéséhez fogott hozzá, ami a jelen pillanatban előrehaladott állapotban, tesztelési fázisban van.

2. AZ ELEMZETT NYELV

2.1. A természetes nyelvek különlegesek!

Már a korábban közzétett cikkben [1] is érveltünk amellett, hogy egy természetes nyelvi elemzőprogram sosem lehet teljesen kész. Ennek okai a következőkben foglalhatók össze:

- egy természetes nyelv határai *sosem mereven rögzítettek*. A természetes nyelv sosem egy világos kontúrral rendelkező, hanem egy homályosan elhatárolt, fuzzy jellegű halmaz. Az elemzőprogram ennek az elmosódott határvonalnak ad egy világos nyomvonalat – sajnos kellő felhatalmazás híján, és gyakran nem is a legszerencsésebb módon.
- egy természetes nyelv *maga sem statikus, hanem dinamikus* valami. Fejlődik, bizonyos fogalmak, kifejezések kivesznek belőle, mások pedig bekerülnek. Például magyar nyelven a „hadirokkant" vagy „hadiárva" kifejezéseket, ha meg is értik, mégis igen kevesen használják már aktívan. Ezek helyett viszont az elmúlt évben a nyelvbe bekerült egy sor „COVID" előtagú összetett szó is, pl. a „COVID-árva" kifejezésről mindenki tudja, mit is jelent.
- tagolódás területi alapon. A természetes nyelvek kicsit *különbözőek egymástól, ha a nyelvjárásokat tekintjük*. Pl. az irodalmi magyar nyelv – a közös államszervezet okán rengeteg német szót átvett az elmúlt századokban. A csángó nyelvjárás hasonló okból rengeteg román szót vett át, pl. náluk a menyasszony és vőlegény „nyirásza" és „nyírel". Sajnos lényegileg az Árpádok kihalása óta kialakult államhatár véget vetett a közös nyelvfejlődésnek, emiatt több magyar szó jelentése kicsit el is tér az irodalmi magyartól… (pl. az „ünő" szót nem csak nőnemű szarvasra használják, hanem a jelentése egyszerűen „tehén"). Mindamellett az igei szerkezetek nyelvtani alakja is gyakran különbözik. Pl. a „szüvem es megszakadt lenne" mondat egy érthető, de az irodalmi és a magyarországi köznyelvben nem használt feltételes szerkezetet fejez ki. Végeredményben pedig egy átlagos csángó beszélő nemigen érti meg az irodalmi magyart, és ez fordítva is igaz: aki Csángóföldre látogat, bizony jópár hét alkalmazkodás után kezdi csak kapisgálni, mit is akarnak a falubeliek közölni.
- léteznek *társadalmi alapú különbségek* is. Pl. a cigányból származó „csaj", ill. „csávó" szó a hatvanas-hetvenes években a lázadó fiatalság közvetítésével került a magyar nyelvbe, így az első időszakban az öregebb magyar beszélők nemigen érthették. Ma is léteznek azonban érthetetlen csoportnyelvek: gondoljunk csak az alvilág kifejezéseire, vagy egy orvosi (netán számítástechnikai) szakvélemény nyelvezetére.

Mindezek a számítógépes megvalósítással szemben *különleges kihívást* jelentenek. Amíg a különbségek csak a használt szavak szintjén maradnak (pl. románból származó csángó szavak), addig az adattartalom eszközeivel jól kezelhetők – csupán a szótár modult kell valamiféle *konfigurálhatósági képességgel ellátni*.

Szerencsére a nyelv területi és társadalmi tagozódása nagyrészt megragadható különböző szavak és kifejezések használatával, ill. azonos szavak különböző értelmű használatával. Vagyis a konfigurálhatóság nagy része megvalósítható a szótár modulban. Csak kisebb rész vonatkozik a nyelvtani szerkezetek különbözőségeire, ami viszont megvalósítható úgy, hogy a nehezebben változtatható elemző modulba lehetőleg minden nyelvi jelenséget beprogramozunk.

Az elemzett nyelv tekintetében még egy további tagozódás fellelhető, még ha az nagyobbrészt mesterséges is. Ez pedig a nyelv használati szintje, vagyis az az A-B-C tagozódás, amit manapság az idegen nyelvek tanulói alkalmaznak [3].

Egy A, azaz alapszintű nyelvtudás párosul egy alapszintű szókinccsel is. Általában a megkülönböztetik az 1. és a 2. szintet, így az A1 egy kezdő nyelvtudást jelent, míg az A2 már vizsgát is ér, egy alapfokú nyelvvizsgát. Ugyanígy, a B szint a középfokú nyelvtudást jelenti, míg a C szint a felsőfok – egy idegen nyelvet C2 szinten beszélők szinte már anyanyelvi szinten birtokolják a nyelvet.

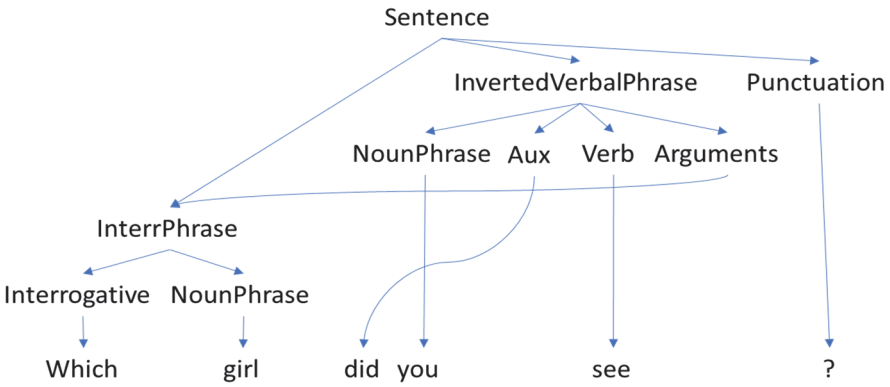
Az egyes szintekhez (legalábbis az A-hoz és B-hez) rögzített szókincs is tartozik, és ez az Interneten meg is található. (Kis bonyodalom, hogy gyakran példákkal is kísérik az egyes szavakat, ám a példákban már nem feltétlenül szorítkoznak a rögzített szókincsre). Ha tehát lényegileg B2 mondatszerkezeteket programozunk be, akkor a különbség az A és B szint között csupán a szókészletben van, ami viszont a szótár szintjén megkülönböztethető. Választhatjuk a statikus utat – ekkor az A2 szintre

„rátölthetjük” a B2-A2 szintek különbségének szavait, vagy a dinamikus utat is, amikor az egyes szavak egyik tárolt jellemzője, hogy melyik szinten jelennek meg először, és a szintek így akár futásidőben is be- vagy kikapcsolhatók.

2.2. Szakadásos nyelvtanok (diszkontinuitások) elemzése

A nyelvben sokszor olyan nyelvi szerkezetek találhatók, amelyekben szorosan összetartozó szavak egymástól távolra kerülnek. Az efféle szerkezeteket *szakadásosnak, nem-folytonosnak vagy diszkontinuitásnak* nevezik. Kezelésük lényege, hogy őket szétválasztva elemezzük, de együtt tároljuk. Grafikusan ezt az *elemzési fa egymást keresztező ágaival* jelezzük.

A legtipikusabb nyelvtani szakadás a kiegészítendő kérdések kérdőszóinál és -kifejezéseinél áll elő. Pl. a „Which girl did you see?” angol kérdésben a „which girl” kérdő kifejezés kétségkívül a „see” ige tárgya – a kifejezés mégsem az ige után, hanem a mondat elején áll. A mondatot az alább látható fával lehetséges ábrázolni.



1. ábra *A „Which girl did you see?” angol mondat szakadásos nyelvtani szerkezete*

Szakadásos szerkezet a legtöbb angolszász nyelvben van – az igekötőket bizonyos állító és kérdő mondatokban is a nyelv a mondat végére dobja. Például:

„Which city have you been living in?” – angolul inkább csak kérdő mondatok esetén…

„Du machst einen Termin mit dem Arzt aus.” – németül állító mondatokban is, de csak egyszerű (jelen, ill. múlt) igeidőkben.

„Ich kann diesen Satz nicht verstehen.” – Németül hasonló jelenség pl. a segédigék mellett a főigék hátradobása is, amit „igekeretnek” is neveznek.

Prolog nyelven az elemzési fát legcélszerűbb Prolog fával ábrázolni, ami viszont nem ad lehetőséget hasonló összekuszálódások ábrázolására. A dolog a hiányos nemterminálisokat tartalmazó változók *kései lekötésével* kezelhető a legegyszerűbben. A hiányos nemterminális megköveteli az elemzés során később a hiány betöltését, vagyis a specifikációjának megfelelő nemterminális megtalálását. Maga a hiány egy Prolog változóval van ábrázolva, ami az elemzés során *lusta (lazy, kései) módon kap értéket*, mihelyt a tényleges elemzés elkészül. Az elemzési fában ugyanaz a kifejezés részfája *két helyen is ábrázolva van*: egyfelől ott, ahol tényleg megjelenik, másrészt ott, ahol lennie kellene (pl. az ige hatókörében). A generátort az ilyen helyeken valamiféle külön funktorral (pl. `ref/1`) értesítjük arról, hogy itt a generálás megáll.

```
sentence(interrPhrase(interr(which),nounPhrase(girl),
         invertedVerbalPhrase(nounPhrase(you,sg/2)
                              aux(did), verb(see),
                              ref(interrPhrase(interr(which),
                                  nounPhrase(girl,sg/3))),
                              punct(?)).
```

2. ábra *A mondat ábrázolása Prolog fával*

2.3. Az elemző tesztelése

A legvégső teszt az elemző ráengedése lehet valamiféle *nagyobb szövegtestre*. Efféle ma már a legtöbb nyelven létezik. Ezen a módon azonban leginkább egy közel kész elemző képességeit értékeljük ki. Az elemző képességeinek fejlesztése egy valamiféle *benchmark-mondatkészlet* segítségével történhet. Ez egyrészt feltételezi, hogy a szóalaktani réteg már tesztelve lett, így az előforduló szóalakokat mindig tökéletesen elemzi a rendszer. Másrészt akkor hasznos egy ilyen mondatkészlet, ha lehetőleg minden fajta mondatszerkezetből tartalmaz néhány tesztpéldát. Efféle kész teszt-mondatsort sajnos nem sikerült szerezni, az informatikus szemmel elkészített mondatsorok bizonyára igényelnék még nyelvészek közreműködését, esetleges módosítását vagy kiegészítését.

2.4. Kiértékelés

Az elemző *kiértékelésére* többféle mérőszám használható.

A legegyszerűbb a sikertelen/sikeres elemzések aránya az elemzett szövegtestre, ill. mondatkészletre vonatkozólag. Itt nyilván a 100%-hoz közeli értékeket szeretnénk megkapni.

Ezt a legegyszerűbb mérési megoldást nemdeterminisztikus elemző esetében bonyolíthatjuk egy *elemzési szám hisztogrammal* is, ami megmondja, hogy hány mondatnak hány elemzése volt. Itt viszont az 1 közeli értékek a siker jelzői.

A bonyolultabb elemzési módszerek legelterjedtebbje a *BLEU* (Bilingual Evaluation Unterstudy), amit elsősorban fordítók esetében használnak [7]. Egyetlen nyelv esetében a BLEU úgy használható, hogy nem a fordítási példamondat-párokra hanem a szövegre magára használja az ember egy kigeneráló modul igénybevétele után. Vagyis az a kérdés, hogy egy elemzés-kigenerálás lépéspár után mennyire pontosan kaphatjuk meg az eredeti forrásmondatot. Megjegyzendő, hogy sem az angol, sem a német elemző-kigeneráló párosra a BLEU vizsgálat nem történt meg. A korábbi siketnyelv-elemző munka esetében a kapott BLEU érték 40%-50% körül mozgott. Ez nem egy kimagasló eredmény (ami a 90-100% lenne), de nem is csapnivaló... A siketnyelv-elemző további fejlesztésével és pontosításával bizonnyal lehetett volna komolyabb értékeket is kapnunk.

3. METASZINTEK ÉS LOGIKAI GUBANCOK

3.1. Ábrázolási kérdések

A *tiszta elsőrendű logika mindenképpen elégtelen* a TNY gondolatok ábrázolásához, ill. mondatok leképezéséhez. Ehelyett legalább *kétszintű modellt* kell használnunk, ami úgy az egyes osztályokra, mint azok példányaira különböző összefüggéseket tárol. Az osztályokat és relációkat, valamint az összefüggéseiket *modellszintnek,* a példányokra vonatkozó információkat pedig *adatszintne*k nevezzük.

Tekintsük pl. az 'enni' igét! „A medvék szeretik a mézet" mondat például nyilvánvalóan csak az osztályok szintjén értelmezhető. A „Micimackó szereti a mézet" mondat alanya már egy példány, a tárgya viszont még mindig egy osztály. A logikai formában nem akarván metaszint átvágással összekuszálódást elérni, a mondat tárgyát logikai kvantorral célszerű ábrázolni.

```
∀y(méz(y)→szereti(Micimackó,y))
```

… illetve Prolog-közelibb formában:

```
szereti(micimackó,X):-méz(X).
```

Ez az ábrázolási mód eléggé kézenfekvő, és garantálja a konkrét példányok korrekt kezelését is.

Már a „szeretni" ige is *többértelmű*: az értelmezésekor is gondolhatjuk, hogy az ige valójában más jelent akkor, ha valami szerelmi szálról, netán baráti vagy rokoni szeretetről, ill. élelmiszerek vagy netán egyes irodalmi vagy zenei műfajok, esetleg sport szeretetéről van szó. Ez a különbségtétel különösen kirívó pl. a „have" (birtokolni) angol ige esetében. Nyilván más jelent valami tárgyat, netán pénzt birtokolni, mint valamit részegységként birtokolni („A car has four wheels"), vagy valamilyen személyt valamilyen szerepben vagy rokonsági fokban számon tartani („I have three daughters", „I have a boss").

A problémát a „have” igéből képződő reláció típusosságával lehet megoldani, ami lényegileg a paraméterek megduplázását jelenti a típusparaméterekkel. Vagyis az ige nyelvi szintű ábrázolása a logikai leképezés során felhasad pl. egy `havePrivateProperty`, `havePiece`, netán `haveRelation` logikai relációkra.

Az ilyen módon *gazdagított paraméterezésű relációkat* is valamiképpen meg kell valósítani, ill. egy ontológiára le kell képezni, legtöbbször az ontológia különböző relációiba. Feltételezve egy Prologban megvalósított ontológiát, az egész egy egyszerű csatolómodullal pillekönnyen megvalósítható.

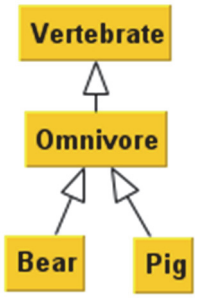
3.2. Statikus és dinamikus következtetés

Az ábrázolt és tárolt ontológia felett *különféle következtetések* hajthatók végre. Természetesen a legegyszerűbb következtető művelet az ontológiában tárolt információk lekérdezése.

Dinamikus következtetés alatt azokat a következtetési lépéseket értjük, amelyek csak futási időben, a következtetéseket leíró szabályok értelmezésével végezhetők el, ill. amelyeket így célszerűbb vagy hatékonyabb elvégezni. *Statikus a következtetés* akkor, ha valamilyen magasabb rendű szabályból előfordítás útján betöltésidőben újabb, alacsonyszintű szabályokat hozunk létre, amelyeket immár futásidőben értékelünk ki.

Erre a legkézenfekvőbb példa az *öröklési hálón végzett következtetés*. A modellszinten egyszerű tényrelációk tárolják az egyes osztályok és részosztályok viszonyát, pl.

```
subClassOf(medve,mindenevő).
subClassOf(mindenevő,gerinces).
```



3. ábra *Osztályszerkezet részlet grafikus és Prolog formában*

Az adatszinten az ismert osztálypéldányok és osztályok viszonya van ábrázolva, pl.

```
medve(micimackó).
```

Azt, hogy egy medve mindenevő, gerinces, netán állat is, futásidőben nem a `subClassOf/2` modellreláció valamiféle értelmezésével döntjük el, hanem a modellszintű relációból a betöltés készít egy adatszintű következtetést, pl.

```
mindenevő(X):-medve(X).
gerinces(X):-mindenevő(X).
```

Ha ezek után bármilyen olyan kérdés merül fel, ami a következményoldalon levő osztályokra vonatkozik, (pl. hány állat lakik a Százholdas Pagonyban?), akkor a reláció segít az összes konkretizált állatot, medvét, disznót, kengurut stb. összeszámolni.

A dinamikus következtetési képesség bemutatására több példát is hozunk.

Az egyszerűbb esetben valamiféle *kvantifikációt* kezelünk, mint az előző példában. A nyelvi szinten felismert kvantifikáció ennek megfelelő logikai kifejezésre képződik le, amit hozzájuk hasonló Prolog kvantifikációs predikátumok valósítanak meg, akár beépített predikátumokkal, akár esetleg azokra

épülő egyszerű csatolómodul predikátumaival. Pl. a számosság kezelése a „hány állat él a Százholdas Pagonyban?” kérdés esetében a következő logikai formulával történik:

```
count(X,X^(állat(X), lakik(X,százholdasPagony),NR).
```

…ami a nagyon hasonló a Prologban használatos `setof/bagof` predikátumokhoz.

3.3 Következtetés a modális világszerkezeten

A korábban leírt *modális világszerkezeten többféle következtetési lehetőség* is adódik [4,5]. Az egyének feletti (szuperindividuális) régió topológiája a (szellemi) közösségek egymásba ágyazási relációját követve az egyes csomópontokat nevezi meg, pl. a *KeresztényKultúra* része a *RómaiKeresztény-Kultúra*, annak része a *MagyarPolgár* kultúrája, aminek része a *MagyarÉrtelmiségi* kultúra stb. Ezen közösségbeágyazásos szerkezet legfontosabb kitétele, hogy az egyes kultúrák az *ős kultúrából örökölnek*, és ezért minden kultúrát jelképező csomópontból örökölnek a leszármazottak, ő maga pedig örököl az őstől. A *világocskafa gyökere* az a tudásanyag, amely minden ágens/egyén számára egyformán, *kétségbevonhatatlanul igaz*. Figyelembe véve, hogy a modális világocska azonosítót első paraméterként beszúrjuk, és hogy a topológiát a `subWorldletOf(SUB,SUPER)` bináris Prolog relációban tároljuk, a modális öröklés Prologhoz hasonló megfogalmazása a következő:

```
RELATION(WSUB,PARMS):-
  subWorldletOf(WSUB,WSUP), RELATION(WSUP,PARMS).
```

Ez a szabály ráadásul konkrét *relációról relációra ismételten megvalósítandó* (ezért a RELATION jelölésmód), tehát ez is statikus, amit a legcélszerűbb betöltésidőben megvalósítani. Említsük meg, hogy a modalitás mind az adatszinten, mind a modellszinten érvényes, ezért mindez a modellszinten (másodrendű szabályként) megfogalmazható a következőképpen is:

```
relation(WSUB,REL/ARITY):-
  subWorldletOf(WSUB,WSUP), relation(WSUP,REL/ARITY).
```

Más fogalmazásban: a *statikus következtetések nem mások, mint a másodrendben leírható következtetések lefordítása betöltésidőben elsőrendű következtetésekké*.

Az egyének alatti (szupraindividuális) régióban a fenti öröklődés már nem érvényes. Minthogy itt az egyes ügynökök (ágensek, egyének) *elmetartalmát* tároljuk, itt legfeljebb azon lehet elgondolkodni, hogy milyen viszonyban vannak egymással az A ügynökre vonatkozó objektív tartalmak, és az A ügynök által valamilyen szinten ismert tartalmak. (pl. igaz-e az, amit A tud, ill. A tudja-e vajon, hogy mit tud, vagy éppenséggel mit nem tud?)

Ezen a szinten az ismerettartalmakra szintén lehetséges bizonyos öröklési relációkat felállítani. Kézenfekvő, hogy *ha valaki tud valamit, akkor azt hiszi is, és netán sejti is*. Ezek szerint, ha csak kettő fokozatot (knows, believes) ábrázolunk, akkor a…

```
relation(believes(WHO),PARMS):- relation(knows(WHO),PARMS).
```

…következtetés is igaz lesz. Ugyanez másodrendű szabályként megfogalmazva:

```
relation(believes(W),REL/ARITY):- relation(knows(W,REL/ARITY).
```

3.4. Külön következtetések

Bizonyos esetekben egyszerű következtetések végrehajtása történik dinamikusan. Például, ha az állatok mind szeretik a gazdájukat és viszont, akkor azt a következő Prolog szabályokkal lehet leírni:

```
szereti(X,Y):- gazdája(X,Y).
szereti(X,Y):- gazdája(Y,X).
```

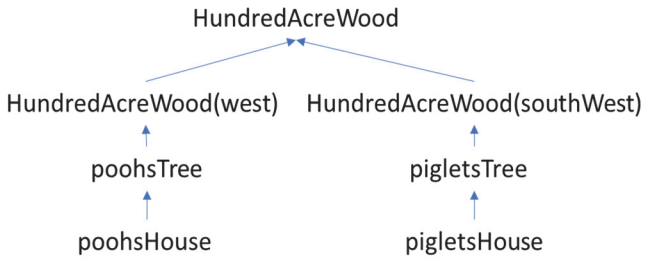
Ha ugyanez a következtetés csak a Micimackó-féle világban volna igaz, akkor első paraméterként a világocskaazonosítót is be kell szúrnunk (pl. milneWinniepooh).

Ez a példa a `gazdája` reláció *szimmetrikus lezárását*, és a `szereti` relációra következtetést mutatja be. Ez azért is érdekes, mert pl. az OWL ontológialeíró nyelv tartalmaz is efféle metarelációkat megadó elemeket (pl. `transitive, symmetric`, stb.). Ha ezeket is alkalmazzuk, akkor a fenti dinamikus megoldás statikussá is tehető, vagyis betöltés (előfordítás)- időben is létrehozhatók a következtetést végző szabályok.

3.5. Legkisebb közös ősre (LCA) következtetés

További érdekes példa egy különleges eset. Ha valami olyan kérdést akarunk értelmezni, amiben *több, különnemű elemre vonatkozó közös tulajdonságra* kérdezünk, pl. „Hol lakik Micimackó és Malacka?", erre az ontológia csak közvetett választ tud adni. Micimackó a „Kovács János" (angolul Sanders úr) házában, Malacka pedig a „Tilos az á" (angolul Trespassers W.) házában lakik, amit a nagyapjától örökölt. Egy válaszadási stratégia lehetne a két lakhely felsorolása, a jelen megvalósítás azonban egy érdekesebb javaslatot ad. A közös és egyetlen lakhely nem más, mint a két egyedi lakhely legkisebb közös őse a `hasGeopart/2` reláció mentén (LCA, Least Common Ancestor). A legkisebb közös ős pedig esetünkben nem más, mint a Százholdas Pagony (Hundred Acre Wood). A megfelelő logikai szabály (amit egy Prolog kiterjesztő-predikátum értékel ki) a következőképpen néz ki:

```
RESULT = lca(Z^(member(X,[Malacka, Micimackó),
                hasResidencelocation(X,Y)),
            hasGeopart(Z,Y)).
```



4. ábra *A Százholdas Pagony földrajzi tartalmazási fájának részlete*

Megjegyezzük, hogy a legkisebb közös ősre történő következtetést más esetekben is használjuk, pl: a „Kicsoda Malacka és Micimackó?" kérdésre válaszolva a `subClassOf` fán – vagyis lényegileg a Linné-féle taxonómiai reláció mentén végzünk LCA műveletet.

3.6. Másodrendű következtetések

A hétköznapi élet mondatai és kérdései sajnos nem mindig elsőrendűek. Az efféle problémák kezelését ún. *kétszintű tudásábrázolással* oldhatjuk meg, amit *reifikációnak* is neveznek. A *példányszintű / adatszintű* tudáselemeken (pl. Kala Pál mikor hol, mit csinált) kívül *modellszintű* tudáselemeket (tudásszegmenst) is tartalmaz, ami a példányszinten használt modelleszközökre: osztályokra, tulajdonságokra és relációkra vonatkozó általános ismereteket, pl. az értelmezési tartományaikat, ill. azok egyéb összefüggéseit tárolja.

A kétszintű tudásábrázolás már *másodrendű kérdésekre* is választ adhat. Pl. a „Mi a különbség Malacka és Zsebibaba között?" kérdésből egy olyan logikai kifejezést hozhatunk létre, amely egyrészt modellszinten megkeresheti a két állatkára csak külön-külön vonatkozó relációkat, de a közös tulajdonságok és relációk különböző értékeit is megkeresheti. Például a `hasKeeper (vanGazdája)` tulajdonság értéke mindkettőjükre „Christopher Robin", míg a `hasResidence` (`lakása`) reláció értéke különböző.

4. ÉRTÉKELÉS ÉS TOVÁBBI MUNKÁK

Az ITSy-Bitsy csevegőrobot körében pillanatnyilag a német elemzőkészítés utolsó simításai történnek – ami nem jelenti még azt, hogy egy terjedelmes tesztelési és kiértékelési fázis elhagyható volna. A jelen írás éppen az említett tevékenységek végzéséhez ad pár szempontot, pár adalékot.

Egy elemző minőségét és hatékonyságát semmiképpen sem lehet bináris skálán értékelni (működik vagy nem), hanem minimum valós skála szükséges hozzá, de lehet, hogy abból is többdimenziós. Ez is egy, még előttünk álló feladat.

Ha mégis az elemzőprogramok terén kilátásba vehető munkákról beszélünk, akkor a következő fontos lépések áll(hat)nak előttünk:

- magyar elemző beépítése, egy létező elemzőprogram rendelkezésre bocsátására ígéretet kaptunk.
- az angol és a német elemzőre hatékonyságmérő eszközök kifejlesztése
- az angol és a német elemző továbbfejlesztése úgy, hogy komolyabb szövegtestekre ráengedhető legyen, legalább elfogadható hatékonysággal
- angol és/vagy német nyelvre szintaktikus indexelőalgoritmus kidolgozása, ami konkrét szövegtesteken (pl. parlamenti felszólalások jegyzőkönyvei) használható lehet

Igen fontos kérdés az is, hogy a munka hovatovább céltalanná válhat, ha nem sikerül köré *együttműködő társakat* szerezni – vagyis meg kellene szerezni valamilyen nyelvtechnológiai műhely együttműködését.

Ugyanilyen fontos lenne valami *alkalmazási példát/helyzetet* találni, ami netán még a források teremtésében is fontos tényező lehet.

5. HIVATKOZÁSOK

[1] Kilián, I.: Szabályvezérelt természetes nyelvi elemzők természetrajzáról *Erdélyi Magyar Műszaki Tudományos Társaság, SzámOkt 2017. konferencia kiadványa,* Kolozsvár, *pp. 159–167,* 2017.

[2] Kilián I.: Csevegőrobotok. Az ITSy-Bitsy modell. *Erdélyi Magyar Műszaki Tudományos Társaság, SzámOkt 2020. konferencia kiadványa,* Kolozsvár, *pp. 110–117,* 2020.

[3] Oktatási Hivatal, Nyelvvizsgáztatási Akkreditációs Osztály: Közös Európai Referenciakeret. Budapest, 2002 (https://nyak.oh.gov.hu/nyat/doc/ker_2002.asp, elérés 2021.szept.13.)

[4] 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 UvA, Amsterdam Center for Language and Comm.),*

[5] Kilián, I.: ℜeALIS: egy többszereplős, episztemikus rendszer Prolog modellje *Erdélyi Magyar Műszaki Tudományos Társaság, SzámOkt 2012. konferencia kiadványa,* Kolozsvár, *pp. 276–281,* 2012

[6] 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*.

[7] Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics. pp. 311–318. 2002.*