

Kvantifikált kifejezések hatóköri többértelműségének szabályalapú kezelése

Szécsényi Tibor

Szegedi Tudományegyetem
Általános Nyelvészeti Tanszék
szecsényi@hung.u-szeged.hu

A magyar nyelvben az ige előtti kvantifikált kifejezések hatóköre követi a szórendet, az ige utániakra azonban jellemző a hatóköri többértelműség. Ezt a jelenséget a HPSG-ben a kvantortárolás segítségével lehet megmagyarázni. A cikk az elméleti megoldás gyakorlati megvalósítását végzi el. A Prolog-alapú, DCG nyelvtan képes kezelni a szabad szórendű magyar mondatokat, és helyes szűk és tág hatóköri olvasatokat rendel a mondatokhoz.

1 A probléma

A természetes nyelvi kifejezések szemantikai homályosságának az egyik oka a kvantifikált kifejezéseket (*minden kalóz, háromnál több indián* stb.) tartalmazó mondatok hatóköri többértelműsége. A kötött szórendű nyelvekben, mint az angol, ezeknek a kifejezéseknek a mondatbeli pozíciója nem nyújt segítséget a kifejezések által bevezetett logikai kvantorok hatóköri viszonyainak a meghatározásához.

A magyar mint részben kötött szórendű, azaz diskurzuskonfigurációs nyelv [4], részben egyértelműsíti a kvantifikált kifejezések hatóköri viszonyait, ugyanis az ige előtti kifejezések sorrendje megegyezik a hatókörük sorrendjével (a '>' a nagyobb hatókört jelenti):

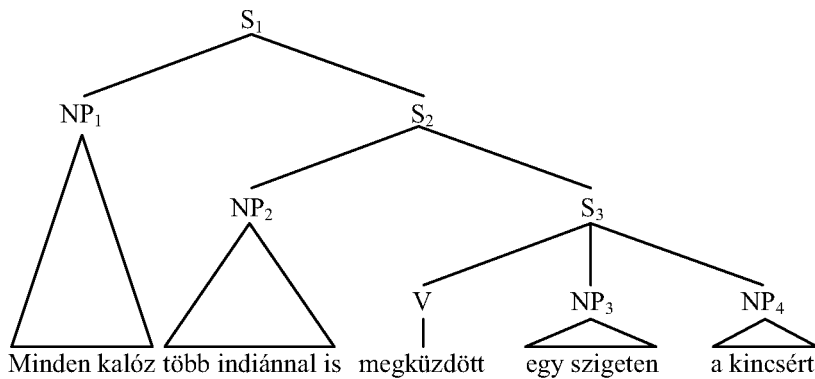
- (1a) *Minden kalóz több indiánnal is megküzdött.*
minden kalóz > több indián
- (1b) *Több indiánnal is minden kalóz megküzdött.*
több indián > minden kalóz

Az igét követő kvantifikált kifejezések hatóköre azonban nem meghatározott, azok hatóköre lehet kisebb is (2a: szűk hatókörű olvasat) vagy nagyobb is (2b: tág hatókörű olvasat), mint az öt megelőző kvantifikált kifejezéséé:

- (2a) *Minden kalóz kibékült néhány indiánnal.*
(2b) minden kalóz > néhány indián, azaz
 $\forall x \exists y (\text{indián}(y) \wedge (\text{kalóz}(x) \rightarrow \text{kibékül}(x,y)))$
- (2c) néhány indián > minden kalóz, azaz
 $\exists y \forall x (\text{indián}(y) \wedge (\text{kalóz}(x) \rightarrow \text{kibékül}(x,y)))$

2 Az elemzés

Korábbi [7], [8] és [9] tanulmányaimban ezt a természetes nyelvi jelenséget próbáltam leírni HPSG ([6]) elméleti keretben. Ezekben a tanulmányokban a klasszikus É. Kiss-féle ([4]) elemzés felszíni szerkezetét tulajdonítottam a magyar mondatoknak, elhagyva ugyanakkor a nála meglevő többi elemzési szintet. A mondat összetevős szerkezete tehát egy igével kezdődő, lapos frázisból és ehhez balról kapcsolódó, hierarchikus bal perifériából áll:



1. ábra: A magyar mondat összetevős szerkezete

A kvantifikált kifejezések hatókörének a meghatározásához a Head-driven Phrase Structure Grammar-ben (HPSG) használatos kvantortárolást használtam ([3]).

A kvantortárolás alapötletét az adja, hogy az olyan predikátumlogikai kifejezéseket, mint ami a (2b)-ben is látható, szétszedhetjük egy magjelentést kifejező részre ('kibékül(x,y)') és a kvantifikált kifejezések jelentését leíró részekre: ' $\forall x(\text{kalóz}(x) \rightarrow P(x))$ ', illetve ' $\exists y(\text{indián}(y) \wedge Q(y))$ '. A kvantorokban található P és Q egy-egy predikátumváltozó, lekötésükre egy-egy halmazképző lambda operátor szolgál: 'minden_kalóz' = ' $\lambda P.\forall x(\text{kalóz}(x) \rightarrow P(x))$ ', illetve 'néhány_indián' = ' $\lambda Q.\exists y(\text{indián}(y) \wedge Q(y))$ '. Az így kapott tulajdonsághalmazokat (általánosított kvantorokat, kvantorokat) mint predikátumokat sorban alkalmazhatjuk a magpredikátumra, így megkaphatjuk 'minden_kalóz(néhány_indián(kibékül))' logikai szerkezetű szűk hatókörű állítást. Ha a kvantorokat fordított sorrendben alkalmazzuk, akkor a 'néhány_indián(minden_kalóz(kibékül))' tág hatókörű olvasatot. Ahhoz, hogy a kvantorokat tetszőleges sorrendben alkalmazhassuk a magra, először össze kell gyűjteni őket. A mondatban szereplő kvantorok összegyűjtése, majd sorbarendezése adja a kvantortárolási elemzést.

A HPSG-ben a kvantorok a kvantifikált kifejezésekből (pontosabban azok determinánsából) származnak, ott a kvantortárolóban (QSTORE) helyezkednek el. Az 1. ábrán minden NP bevezet egy-egy kvantort. Az NP kategóriák fölötti S kifejezések összegyűjtik az összetevőikben jelen levő kvantorokat. S₃ kvantortárolójában megtalálható NP₃ és NP₄ kvantorai: Q₃ és Q₄; S₂-ben NP₂ kvantora, Q₂, valamint S₃ összegyűjtött kvantorai, {Q₃, Q₄}; S₁ QSTORE-ja pedig a következő: {Q₁, Q₂, Q₃, Q₄}.

A logikai kifejezés magja az igei fejből származik, valamint az ige és az őt domináló kifejezéseken jelöljük, hogy a kifejezésekben szereplő kvantorok milyen sorrendben alkalmazandóak a magra. Ez az igei fejú frázisok QUANTS listáján van megadva, tehát:

- (3) Egy igei fejú S frázis esetén az összetevők QSTORE halmazában meglévő kvantorok vagy az S QSTORE-jában jelennek meg, vagy az S QUANTS listájának az elején (a QUANTS lista további része az S által közvetlenül domináló igei fejú összetevő QUANTS listájával azonos).

Az így kialakult mondat szerkezet esetén tehát – üres QSTORE halmazt feltételezve – a QUANTS lista megadja a kvantorok hatóköri sorrendjét.

A fent leírt módszer a HPSG általános kvantorértelmezési módszere, segítségével a kötött szórendű, konfigurációs nyelvek esetében is meg tudjuk magyarázni a hatóköri többértelműséget. A magyarban azonban, mint azt az (2) példák is mutatják, csak az ige utáni kvantifikált kifejezések hatóköre lehet szabad, az ige előtti kvantifikált kifejezések hatóköre egymáshoz képest kötött, az (1) példák szerint a kifejezések sorrendje meghatározza a hatóköri sorrendet. [7], [8], és [9] szerint a magyarban csak az ige utáni, komplementumpozícióból származó kvantorokra vonatkozik a (3) szabály, az igit megelőző, azaz filler-pozíciókból származó kvantorokra a (4) kiegészítő szabály is vonatkozik:

- (4) Ha egy igei fejú S frázisnak van ige előtti, azaz filler-összetevője, akkor annak a QSTORE-jában megtalálható kvantorok nem jelenhetnek meg az S QSTORE-jában.

Az 1. ábrán látható szerkezetben így az ige előtti NP_1 és NP_2 összetevőkből származó Q_1 , illetve Q_2 kvantorok nem az őket domináló S_1 , illetve S_2 frázisok QSTORE halmazában jelennek meg (4 szabály), hanem a megfelelő QUANTS listák élén (3 szabály). Mivel azonban S_1 QUANTS listájának a további része S_2 QUANTS listájával egyezik meg, amelynek viszont Q_2 volt az első eleme, a Q_1 kvantor mindig nagyobb hatókörű lesz, mint a Q_2 kvantor, vagyis az ige előtti kvantifikált összetevők sorrendje megegyezik a hatóköri sorrenddel. Az ige utáni kifejezésekből származó Q_3 és Q_4 kvantorokra viszont nem vonatkozik a (4) kiegészítő szabály, azok bármely S kifejezésnél átkerülhetnek a QUANTS listára, vagy tovább másolódhatnak a QSTORE kvantor-tárolóba.

3 Az implementáció

Az előző fejezetben ismertetett elméleti elemzés ellenőrzéseként szükséges a gyakorlatba is átültetni a megoldási javaslatot. Az elemzés nagyban épít a HPSG elméleti keretre. Létezik ugyan, és el is érhető a HPSG-nek számítógépes implementációja ([5]), azonban az egy fontos szempontból nem bizonyul kielégítőnek: nem tudja kezelni a magyar nyelvre jellemző szabad szórendűséget. Ezért arra vállalkoztam, hogy egy alapjaitól újra felépített elemző megalkotására teszek kísérletet. Ez, bár nem telje-

sen követi híven a HPSG formalizmusát, szellemében megfelel annak, és lehetőséget nyújt arra, hogy egy jobban, pontosabban kidolgozott implementáció része, alapja legyen.

Mivel a jelenség elemzése frázisstruktúra-nyelvtannal történt, az alkalmazás Prolog nyelven történt, ahol a beépített DCG formalizmus nagy segítséget nyújt a frázisstruktúra nyelvtanok megfogalmazására.

Az alkalmazás több modulból álló nyelvtant feltételez. Az első modul a lexikai egységek lexikaiegység-specifikus tulajdonságait adja meg, úgymint hangalak, jelentés, ragozási paradigma stb. Ezekből építi fel a következő modul a tényleges alap lexikai egységeket, specifikálva az előző egység által csak jelzett tulajdonságokat – itt derül ki például, hogy egy tranzitív igenek pontosan milyen vonzatszerkezete van. A harmadik modul a lexikai szabályokat tartalmazza, amelyek egy alap lexikai egység variánsait adják meg. A negyedik modulban találhatóak a tényleges szintaktikai/grammatikai szabályok, amelyekkel összeállíthatjuk a frázisokat, az összeállítással párhuzamosan azok szemantikai leírását is megadva. Ezzel a nyelvten nemcsak elemezni képes magyar nyelvű mondatokat, hanem a mondatok jelentésrepresentációja is előáll. Ennek a jelentésrepresentációnak az olvashatóbb, predikátumlogikai formájúra átalakítását egy további modul végzi. Ez a modul teljes egészében a [1]-ben ismertetett megoldással azonos, amely elérhető [2]-n. A lexikaiegység-specifikus tulajdonságokat tartalmazó modul szintén [1] szellemében épült fel, bár nyelvspecifikussága miatt nyilvánvalóan nem változatlan átvétele annak.

3.1 A lexikaiegység-specifikus tulajdonságok

A lexikaiegység-specifikus tulajdonságokat tartalmazó modulban a lexikai egységeknek azon tulajdonságai, amelyek tipikusnak mondhatóak, csak jelzésszerűen vannak megadva, ilyen például a következő *minden* determináns esetében a szófaj: *det*. Azok a tulajdonságaik, azonban, amelyek egyediek teljes részletességükben, ahogyan ez a szemantikai leírásnál is látható.

```
lexentry(
  det,
  [def(indef), word([minden]), index(I),
   sem(lam(S, lam(Q, all(I, imp(app(S, I), app(Q, I))))))]).
```

Ugyanez a *kibékül* tranzitív igenél a következőképpen alakul. A szó igei kategóriájú, azon belül is tranzitív, mégpedig olyan, amelyiknek a második argumentuma *-val/vel* esetű kell hogy legyen (*tv2*), csakúgy, mint például a *találkozik* vagy a *megismerkedik* ige. A jelentésleírásában osztozik a tranzitív igeikkel, mindegyik ugyanolyan séma alapján épül fel, csak a predikátum változik benne (*symbol(kibékül)*).

```
lexentry(
  tv2,
  [fin(fin), word([kibékül]), symbol(kibékül),
   agr(sg, 3, indef)]).
```

3.2 Az alap lexikai egységek

A lexikai egységek a `lexentry` definíciók adatainak a felhasználásával állnak össze:

```
lex(
  synsem(
    cat(...),
    content(...) ),
  qStore([bo(app(SemDet, SemN), I)]),
  slash([]))
-->
{lexentry(det,
  [def(Def), word(Word), index(I), sem(SemDet)]),
  Word.
```

A determinánsok (amelyeknek a szintaktikai (`cat`) és szemantikai (`content`) tulajdonságainak a részletezésétől eltekintek) `qStore` listáján egyetlen elem található, a determinánssal kezdődő főnévi csoport kvantorának a leírása. A `slash` lista leírása a lexikai szabályoknál lesz megtalálható.

A hatókör-értelmezés szempontjából érdekes még az igék szerkezete:

```
lex(
  synsem(
    cat(
      head(v(Fin)),
      comps([
        synsem(
          cat(head(n(nom)), args(_), deps(_), comps([], _),
            content(agr(Num, Per, _), index(I1), restr(_))),
          synsem(
            cat(head(n(ins)), args(_), deps(_), comps([], _),
              content(agr(_, _, _), index(I2), restr(_))))),
        content(
          agr(Num, Per, Def),
          quants([],
            nucleus(Sem))),
      ]),
    qStore([],
      slash([]))
  -->
  {lexentry(tv2, [fin(Fin), word(Word), symbol(Sym),
    agr(Num, Per, Def)]),
    Word.
```

Itt a `comps` lista tartalmazza a `tv2` típusú igék argumentumszerkezetét, ezen lista alapján tudjuk majd ellenőrizni a szintaktikai szabályoknál, hogy a mondatban megjelenő komplementumok megfelelőek-e az őket vonzó ige számára. A `nucleus` adja meg az ige jelentését, ami egyúttal a mondat magjelentése. A `quants` lista a magra alkalmazandó kvantorok sorrendjét, vagyis a kvantorok hatóköri sorrendjét tartalmaz-

za. Ez a lista üres az igék lexikai leírásánál, csakúgy, mint a `qStore` és a `slash` lista is.

3.3 Lexikai szabályok

A nyelvtan jelen pillanatban csak egyetlen lexikai szabályt tartalmaz.

A HPSG-ben az összetevős szerkezetek kialakításának két módja van. Az egyik az, amikor a szerkezet egyik összetevője, a szerkezet feje meghatározza, hogy milyen más összetevők, azaz komplementumok lehetnek még a szerkezetben. A fej lexikai leírásában szerepel a `comps` lista, amely a komplementumokat sorolja fel. Amikor egy komplementum összecsatlakozik a fejjel, akkor a komplementum unifikálódik a `comps` lista egyik elemével. A lista tehát azoknak az összetevőknek a leírását tartalmazza, amelyek még hiányoznak a fej mellől ahhoz, hogy teljes frázist – mondatot, főnévi csoportot stb. – kapjunk. Ha egy frázis tehát ilyen fej-komplementum szerkezetű, akkor a fej `comps` listája tartalmazza a komplementumot, a frázis `comps` listájáról azonban már hiányzik.

A másik frázisalkotási mód az olyan hiányos kifejezéseknek a hiányait szünteti meg, mint amilyen az elliptikus mondat, a kérdőszó-kiemeléses mondat vagy a datívuszi birtokos kimozgatásával hátra maradt hiányos főnévi csoport. Az ilyen jellegű hiányokat a kifejezések `slash` listái tárolják. Akkor jelenik meg egy kifejezés valaminek a `slash` listáján, ha az a kifejezés az elvárt komplementumpozíciójától távol kerül majd elő. A mondat szerkezet alapján a listán megjelenő kifejezések a mondat szerkezetben fölfelé összegyűlnek, majd egy bizonyos ponton *filler* összetevőkként jelennek meg. A mi elemzésünk szempontjából ilyen filler összetevők az igét megelőző pozícióban található kvantifikált kifejezések.

Mivel egy kifejezés nem lehet egyszerre komplementum és filler is, a következő lexikai szabály az alap lexikai leírásban szereplő `comps` listát kettéválasztja valóban komplementumként megjelenő elemekre és filler összetevőként megjelenő elemekre, így egy új lexikai egységet hoz létre, ami az eredetinek egy argumentumszerkezeti variánsa:

```
sign(
  synsem(
    cat(head(v(fin)), comps(Comps)),
    Content),
  qStore,
  slash(Slash))
-->
lex(
  synsem(
    cat(head(v(fin)), comps(CompsHead)),
    Content),
  qStore,
  slash([])),
  {shuffle(Slash, Comps, CompsHead)}).
```

A szabályban szereplő `shuffle` predikátum a `Slash` és a `Comps` lista elemeit csúsztatja össze oly módon, hogy az eredeti listák elemeinek egymáshoz viszonyított sorrendje ne változzon – mint amikor két pakli kártyát csúsztatunk össze.

3.4 Szintaktikai szabályok

A kvantifikált kifejezések hatókörének a meghatározásához szükséges a kifejezések mondatban elfoglalt pozíciójának meghatározása, úgyhogy elsődlegesen a tényleges mondatelemzéshez szükséges szabályokat vizsgáljuk meg, a megfelelő pontokon rámutatva, hogy a kvantorok hatókör-értelmezésénél az adott ponton milyen részletek játszanak szerepet.

A magyar mondatok szerkezete az 1. ábrán bemutatottak szerint két fő részből áll. Az egyik az ígét és az öt követő mondatszakasz összetevőit tartalmazza, és mindegyik összetevő a lexikai ige testvére.

Az ígét követő összetevők az ige komplementumai. Ebben a mondatszakaszban az összetevők sorrendje szabad, jelentéskülönbséget (és hatóköri különbséget) nem okoz az összetevők felcserélése. Az igei fejtű, lapos, szabad komplementumsorrendű szerkezetet a `sign2` kategória generálásával hozzuk létre:

```
sign2(
  synsem(
    cat(head(v(fin)), comps(CompsVP)),
    Content),
  qStore(QStoreVP),
  Slash)
-->
{shuffle([SynsemArg], CompsVP, CompsHead)},
sign2(
  synsem(
    cat(head(v(fin)), comps(CompsHead)),
    Content),
  qStore(QStoreV),
  Slash),
sign(SynsemArg, qStore(QStoreArg), _),
{append(QStoreArg, QStoreV, QStoreVP)}).
```

`sign2` rekurzívan előállítható egy igei fejből és az igei fej egy véletlenül kiválasztott komplementumából, és az eredményül kapott kifejezés `comps` listája eggyel rövidebb, mint az ő igei fejéé: `shuffle([SynsemArg], CompsVP, CompsHead)`. Az így létrehozott kvázi lapos szerkezet generálásakor semmi más nem történik, csak a `comps` lista kiürül, és összegyűlnek a komplementumok `qStore` listáján tárolt kvantorai: `append(QStoreArg, QStoreV, QStoreVP)`.

Az így kapott, üres `comps` listájú igei kifejezés már megfelel az 1. ábra legalsó `S` kategóriájának:

```

sign(
  synsem(
    cat(head(v(fin)), comps([])),
    content(Agr, quants(QuantsVP), Nucleus)),
  qStore(QStoreVP),
  slash(SlashVP))
-->

sign2(
  synsem(
    cat(head(v(fin)), comps([])),
    content(Agr, quants(QuantsV), Nucleus)),
  qStore(QStoreV),
  slash(SlashVP)),
{quantorRule(QStoreVP, QStoreV, [], QuantsVP, QuantsV)}).

```

Ezen a ponton történhet meg először az eltárolt kvantorok bármelyikének a hatókörének a meghatározása, azaz itt kerülhetnek át elemek a qStore halmazból a quants listára. Ezt a (3) szabályban leírtaknak megfelelően a quantorRule predikátum végzi el:

```

quantorRule(QStoreMother, QStoreHead, QStoreSister,
            QuantsMother, QuantsHead):-
  append(QStoreSister, QStoreHead, Temp1),
  deleteSubList(Temp2, Temp1, QStoreMother),
  append(Temp2, QuantsHead, QuantsMother).

```

A definícióban szereplő deleteSubList az első argumentum elemeit törli a második argumentumról, és a maradékot a harmadik argumentumba teszi.

A magyar mondat szerkezet másik fő részében az igét megelőző összetevők egyenként csatlakoznak az előzőekben kialakított, komplementumaival már teljes mértékben kiegészített kifejezéshez:

```

sign(
  synsem(
    cat(head(v(fin)), comps([])),
    content(Agr, quants(QuantsS), Nucleus)),
  qStore(QStoreS),
  slash(SlashMother))
-->
{shuffle([SynsemFiller], SlashMother, SlashHead)},
sign(SynsemFiller, qStore(QStoreFiller), _SlashFiller),
sign(
  synsem(
    cat(head(v(fin)), comps([])),
    content(Agr, quants(QuantsVP), Nucleus)),
  qStore(QStoreVP),
  slash(SlashHead)),
{quantorRule(QStoreS, QStoreVP, QStoreFiller,
            QuantsS, QuantsVP),
 subSet(QStoreFiller, QuantsS)}).

```


A balról csatlakozó filler összetevők a fej slash listájáról kerülnek ki egyenként, tetszőleges sorrendben. Az összetevők kvantorai, csakúgy, mint az előző újrairó szabály esetében is, választhatóan kerülhetnek a szülőcsomópontnak a qStore halma-zába vagy a quants listájára. Pontosabban ez az opció csak az igei fejről származó kvantorok számára nyitott, a filler összetevő kvantora kizárólag a quants listára ke-rülhet: `subSet(QStoreFiller,QuantsS)`. Ez a (4) szabály Prolog-megfelelője.

3.5 A mondat szemantikai tartalmának predikátumlogikai formulává alakítása

A tényleges mondatelemzési folyamat ezzel készen is van, a nyelvtan képes generálni és elemezni a feltételeknek megfelelő magyar mondatokat: szintaktikailag azokat a nyelvi jeleket (`sign`) tekinti mondatnak, amelyeknek a kategóriája ige (`cat(head(v(fin)))`), komplementumai mind szerepelnek a kifejezésben (`comps([])`), és a filler összetevői is megjelentek a bal periférián (`slash([])`). A mondat szemantikai értelmezhetőségéhez még az is szükséges, hogy valamennyi kvantornak meg legyen határozva a hatóköre (`qStore([])`).

A kvantorok hatóköreinek az erőviszonyait, mint azt a 2. szakaszban láthattuk, a kvantorok `quants` listán elfoglalt helye egyértelműen meghatározza. Hogy ezt szemléletesen is belássuk, alakítsuk át a kapott kvantorlistát könnyebben olvasható, predikátumlogikai formulává!

A *Minden kalóz kibékült néhány indiánnal* mondat elemzése után a `nucleus` és a `comps` tartalmazzák a logikai kifejezés magját és a kvantorok listáját, a tág hatókörű olvasat esetén például ez a lista a két elemű, a lista első tagja a *néhány indián* kvantora, a második eleme pedig a *minden kalóz* kvantora. Először egyetlen formulává ala-kítjuk a magjelentést és a kvantorokat úgy, hogy a kvantorokat a legkisebb hatókörű-től a legnagyobb hatókörű felé haladva egymás után alkalmazzuk a magjelentésre. Ekkor kapunk egy λ -formulát:

```
app(app(lam(_G298,lam(_G301,exist(_G304,
and(app(_G298,_G304),app(_G301,_G304))))),lam(_G276,
indián(_G276))),lam(_G276,app(app(lam(_G116,lam(_G119,
all(_G122,imp(app(_G116,_G122),app(_G119,_G122))))),
lam(_G72,kalóz(_G72))),lam(_G72,kibékül(_G72,_G276))))))
```

Ugyanez konvencionális formában (a @ a függvényalkalmazás jele):

$$(5) \quad ((\lambda R.\lambda S.\exists v(R@v \wedge S@v) @ \lambda y.indián(y)) @ \lambda y.((\lambda P.\lambda Q.\forall w((P@w) \rightarrow (Q@w)) @ \lambda x.kalóz(x)) @ \lambda x.kibékül(x,y)))$$

Ezen végrehajtva az [1]-ben használt, [2]-ben elérhető β -konverziót, megkapjuk a szokásos elsőrendű formulát:

```
exist(_G304,and(indián(_G304),all(_G999,
imp(kalóz(_G999),kibékül(_G999,_G304))))
```

Ugyanez konvencionális formában:

$$(6) \quad \exists y (\text{indián}(y) \wedge \forall x (\text{kalóz}(x) \rightarrow \text{kibékül}(x,y)))$$

(6) logikailag ekvivalens (2c)-vel. A Prolog-implementáció megadja a szűk hatókörű olvasatot is, amely a szükséges konverziókkal (2b)-vel ekvivalens formulává alakítható. Az elméleti megoldás számítógépes implementációja tehát helyesen működik, képes megadni az elvárt hatóköri többértelműséget.

4 További lehetőségek

Az implementáció, mivel egy kidolgozott elméletre, a HPSG-re alapul, kibővíthető további grammatikai szabályokkal, amelyek például szabályozhatják, hogy az ige előtt pontosan milyen elemek és hol jelenhetnek meg, gondolva itt a fókuszértelmezésre és a topicalizációra. A már meglévő implementációrészek azonban ebben a kibővített elemzőben is megfelelően működnek.

További bővíthetősége az implementációnak, hogy a rendszer az [1]-ben bemutatott elemekkel kiegészítve az elsőrendű logikai kifejezések alapján képes egy mondatot interpretálni egy megadott világmodellben, vagyis egy olyan lekérdező rendszert készíthetünk, amelyben a kérdések természetes nyelven vannak megfogalmazva.

Bibliográfia

1. Blackburn, P., Bos, J.: Representation and Inference for Natural Language: A First Course in Computational Semantics. CSLI Press (2005)
2. Blackburn, P., Bos, J.: Representation and Inference for Natural Language: Software Requirements and Downloads: <http://homepages.inf.ed.ac.uk/jbos/comsem/software1.html>
3. Cooper, R.: Quantification and Syntactic Theory. Reidel, Dordrecht (1983)
4. É. Kiss, K.: Configurationality in Hungarian. Akadémiai Kiadó, Budapest (1987)
5. Penn, G.: The ALE Homepage: <http://www.cs.toronto.edu/~gpenn/ale.html>
6. Pollard, C., Sag, I A.: Head-Driven Phrase Structure Grammar. CSLI – University of Chicago Press, Stanford – Chicago (1994)
7. Szécsényi T.: Sorrend és hatókör a magyarban: HPSG elemzés. Nyelvtudomány Vol.1 (2005) 171–205
8. Szécsényi T.: Lokális és argumentumöröklés. A magyar infinitívuszi szerkezetek leírása HPSG keretben. Doktori értekezés. Szeged, SZTE (2009)
9. Szécsényi T.: Magyar mondatszerkezeti jelenségek elemzése HPSG-ben. In: Bartos Huba (szerk.): Általános Nyelvészeti Tanulmányok XXIII (2011) 99–138