

Szintaktikailag elemzett birtokos kifejezések algoritmizált fordítása adott formális nyelvre

Kardkovács Zsolt Tivadar, Tikk Domonkos

BME Távközlési és Médiainformatikai Tanszék, Média Labor,
H-1117 Budapest, Magyar tudósok körútja 2.
{kardkovacs,tikk}@medialab.bme.hu

Kivonat: Számos nemzetközi szakirodalom [5; 7; 10; 17; 20] foglalkozott a birtokos szerkezetek szemantikai modellezésével, szemantikai sajátosságainak bemutatásával, azonban az eddig megalkotott modellek valamely konkrét birtokos szerkezetnek pontosan megfelelő formális mondat automatizált előállítását nem biztosítják. A cikkben megmutatjuk, hogyan lehet a problémát általános formában megoldani, illetve megmutatjuk, hogy az algoritmussal támogatott feldolgozásnak hol vannak a korlátai, melyek a még megoldandó feladatok.

1 Bevezetés

A birtokos jelzős szerkezetek nagyon sokféle szemantikai kapcsolatot fejezhetnek ki [5], ráadásul a birtokos- és birtokszerepek is felcserélődhetnek eltérő szövegkörnyezetben (pl. a könyv szerzője, a szerző könyve), így algoritmizált feldolgozásuk, formalizálásuk korántsem egyszerű feladat. A ma használatos általános célú válaszkereső rendszerek (Question Answering Systems, QAS), illetve internetkereső-motorok [3; 9; 4; 8] egyik jellegzetes hiányossága éppen ebből ered, hiszen természetes nyelvű bemenetek [2], és azon belül a birtokos szerkezetek feldolgozása nélkül, szótövezéssel, valamint a szóeloszlások statisztikai mutatói alapján azonos válaszokat kell kapjunk az alábbi kérdésekre:

- „Mikor látogatott az Egyesült Államok elnöke Oroszországba?”
- „Mikor látogatott a Oroszország elnöke az Egyesült Államokba?”
- „Oroszország melyik elnöke látogatott az Egyesült Államokba?”
- „Kit látogathat meg Oroszország és az Egyesült Államok elnöke?”
- ... (stb.)

A statisztikai modelleken alapuló megoldást azonban el kell vetnünk más okból is, hiszen a birtokos szerkezetnél – néhány idiomatikus kapcsolattól eltekintve – gyakori ismétlődésekre nem készülhetünk fel [5; 21], mélyreható szemantikai analízis nélkül pedig az egyes szerkezetekben rejlő sajátosságokat nem is azonosíthatjuk be. Sőt, a mondatstruktúra általános szintaktikai jellemzői sem feltétlenül segítenek a tájékozódásban, hiszen nagyon hasonló morfológiai, mégis lényegesen különböző mondattani szerkezettel rendelkeznek pl. az alábbi mondatrészletek is:

Józsinnak az Írott-kő megmászása...

Józsának a ColorStar tévéje...

Editnek a váza összetörése... [7]

Editnek az arca visszatükröződése...

(Itt jegyezzük meg, hogy Chisarik állításának, miszerint egy magyar mondatban nem lehet jelen DAT és NOM birtokos is ugyanabban a főnévi frázisban⁷³, a második mondatrészlet azonban ennek ellentmondani látszik.)

A „Szavak hálójában” projekt⁷⁴ keretében arra tettünk kísérletet, hogy a már létező nemzetközi tapasztalatokat összegezve egy olyan magyar nyelvű válaszkereső rendszert hozunk létre, amely – legalábbis magyar nyelven – szemantikai szinten képes feldolgozni a felhasználói kérdésben megjelenő birtokos szerkezeteket.

A feldolgozás két nagy lépésre bontható: egy szintaktikai és egyfajta szemantikai elemzésre. A továbbiakban – terjedelmi okokból – csak az utóbbiról lesz csak szó. Ennek megfelelően, a kiindulási állapotunkban feltételezzük, hogy adottak a frázisban a birtokos és a birtok szerepű tagok – függetlenül attól, hogy azok szemantikailag helyesen vannak-e összerendelve –, és ehhez a struktúrához keressük a megfelelő formális, SQL nyelvű leírást, ha ilyen létezik. Cikkünkben megmutatjuk, hogy hogyan juthatunk az 1. táblázatban látható módon, annak baloldali eleméből kiindulva a jobboldali megfelelőjéig, vagy annak ekvivalenséig.

1. Táblázat: Birtokos és a velük ekvivalens SQL-kifejezések

Birtokos kifejezés	SQL
Bizet Carmenje	SELECT cim FROM operak WHERE szerzo = 'Bizet' AND cim = 'Carmen'
Shakespeare drámái	SELECT cim FROM dramak WHERE szerzo = 'Shakespeare'
Edit címe	SELECT cim FROM cimek WHERE nev = 'Edit'
könyvek szereplői	SELECT szereplo FROM szerepek WHERE darab IN (SELECT cím FROM konyvek)
vállalat vezetői	SELECT fonok FROM vallalat
Petőfi anyjának a neve	SELECT nev FROM személyek WHERE nev IN (SELECT anya FROM csaladfa WHERE gyermek = 'Petőfi')

⁷³ “Hungarian now presents an interesting puzzle: it is impossible to have both a genitive and a dative possessor in the same noun phrase.” (Chisarik, 2001 – p.11)

⁷⁴ A Magyar Köztársaság Kutatás-fejlesztési Hivatala által támogatott, NKFP-0019/2002 jelű projekt

A következő szakaszban bevezetjük a megoldás bemutatásához szükséges alapvető fogalmakat, majd megmutatjuk, hogy a birtokos szerkezetek feldolgozása milyen három jellegzetes típusproblémára redukálható. A harmadik szakaszban bemutatjuk a feldolgozást végző algoritmust. Rövid példákkal és diszkussziókkal illusztráljuk az algoritmus működését és sajátosságait a negyedik szakaszban. Végezetül, az utolsó szakaszban összefoglaljuk az elért eredményeinket.

2 A fogalomhasználatról

Birtokos szerkezetek nagyon változatos szemantikai kapcsolatokat képesek kifejezni a természetes beszédben, ami új kihívásokat jelent a nyelvfeldolgozásban is (lásd 2. táblázat). Ez alatt azt kell érteni, hogy egy szintaktikai elemzés után rá kell ismernünk arra az aggregációra, hivatkozásra vagy valamilyen relációra egy tudásbázisban, amelyek pontosan az adott birtokos kapcsolatot jellemzi.

2. Táblázat: Birtokos szerkezetek típusai

Genitivus típusok	Példakifejezések (angol és magyar)
származás-, forrásleírás	Moszkva küldötte (men of Rome)
anyagleírás	– (ring of gold)
rész-egész viszony	a tanszék vezetője (head of department)
mennyiségi leírás	húsnak kilója (pound of beer)
jellemzés	a jövő embere (man of yesterday)
(állandósult) kapcsolat	Péter felesége (Pam's address)
birtoklás leírása	Sára sapkája (John's coat)
alanyiség	Verdi operája (dramas of Shakespeare)
tárgyiasság	Bocskai portréja (II. Erzsébet arcképe)
cél- és szándékleírás	dolgozók iskolája (girls' school)
láncolás	Ábel apjának barátja (name of Tom's wife)

Jelölje $X \Rightarrow Y$ azt a birtokos szerkezetet, amelyben X a szerkezetben, szintaktikai értelemben a birtokos szerepű tag, míg Y a birtok.

A szintaktikai feldolgozást követően, azaz ha adott $X \Rightarrow Y$, valamiféle jelentéstani feldolgozást kell végezni, majd ennek segítségével kell egy formális megfelelőt, pl. SQL lekérdezést belőle előállítani. Természetesen, a kifejezés pontos jelentése függ a benne található szavak értelmétől, ahogyan ezt a 2. táblázatban is láthattuk, ráadásul ezeknek a feldolgozása egyáltalán nem magától értetődő (lásd 1. táblázat). Az SQL nyelven lekérdezhető relációs adatmodellek nem rendelkeznek átfogó szemantikai leírással, így két lehetőség áll előttünk:

1.) ontológiai modelleket építünk, amely kölcsönösen egyértelműen leképezi az adatbázis struktúráját a modellre és ebből kiindulva próbáljuk meg a nyelvtani kifejezésben szereplő kapcsolat jelentését értelmezni,

2.) vagy olyan sémastruktúrát alakítunk ki, amelyből kinyerhető a számunkra szükséges szemantikai információ.

Az ontológiaépítés problémáiból, nehézségeiből tanulva felmerül a kérdés, hogy nem lehetséges-e pusztán a második megfontolás alapján megoldani a problémát?

A kérdés megválaszolásához először tekintsük át, milyen elemekkel dolgozhatunk egy általános (relációs) adatbázis esetében. Az adatbázis adattartalma leírható egy négyes segítségével, azaz legyen $DB = \langle V, I, S, A \rangle$, ahol V az adatbázisban található konkrét értékek, kitöltések, I a példányok, rekordok, S a sémák vagy osztályok és A az attribútum, leírók halmazai.

Ha megvizsgáljuk a birtokos szerkezetek adatbázisokra vetített viselkedését, akkor végeredményben hat típust különböztethetünk meg: a birtokos minden esetben séma (osztály) vagy rekord (példány, individuum), míg a birtok szerepű kifejezés egyaránt lehet séma, rekord vagy attribútum (6db aleset).

A ma ismert implementációk (pl. [16; 12; 19; 1; 22; 18; 15]) nem használnak fel ontológiai tudást, kizárólag az adatbázis-tartalomból építkeznek. Rögtön hozzáteszszük, hogy az implementációkhoz tartozó adatmodellek nagyobb hányada az $\langle \text{objektum, tulajdonság, érték} \rangle$ hármásra illeszthető [13; 14], amely csak a példány és a tulajdonság, valamint a séma (objektumosztály) és az individuum közötti birtokos szerkezettel leírható viszonyokat ($S \Rightarrow A$ és $I \Rightarrow A$ típus) képes megragadni.

Az általánosabb megoldáshoz további fogalmakat kell bevezetnünk.

1. Definíció (Természetes kulcs). *Legyen adva egy $DB = \langle V, I, S, A \rangle$ adatbázis, ahol rendre V az értékek, I az individuumok, S a sémák és A az attribútumok nem üres halmaza. Legyen $\kappa: S \rightarrow A$ egy függvény, amely minden $s \in S$ elemet valamely $\alpha \in A$ elemre képez le úgy, hogy s minden i individuumát a természetes nyelvben éppen az i α attribútumának a $v \in V$ értékével nevezzük meg. Ebben az esetben azt mondjuk, hogy $\kappa(s)$ az s séma természetes kulcsa.*

Más szavakkal, a természetes kulcsok a mondatban szereplő (esetleg összetett) névszók, amelyek a nyelvben a való világ egy-egy entitását leíró, konkrét, adott nyelvű, és az adott környezetben egyértelmű megnevezései. Például $\kappa(\text{könyv}) = \text{cím}$, vagy $\kappa(\text{személy}) = \text{személynév}$. Vegyük észre, hogy a természetes kulcsok a közismert adatbáziskulcsoktól abban tér el, hogy a természetes kulcsokra nincs rögzítve, hogy szükségszerűen egyedinek kellene lennie. Azaz, a természetes kulcs lehetővé teszi a adatbázisban a nyelvi többértelműség megjelenítését.

2. Definíció (Hivatkozási függvény). *Legyen adva egy $DB = \langle V, I, S, A \rangle$ adatbázis, és legyen $\alpha \in A$ az $s \in S$ séma egy attribútuma. Tegyük fel továbbá, hogy egy attribútumnév legfeljebb egy sémában fordul elő, és DB -n értelmezett a természetes kulcs fogalma. A $\varphi: A \rightarrow A$ függvényt hivatkozási függvénynek nevezzük, ha az adatbázisban*

1.) minden attribútumra $\varphi(\alpha) \in A$,

2.) $\varphi(\kappa(s)) = \kappa(s)$,

3.) $\varphi(\alpha) = \kappa(s')$, ahol $s \neq s'$.

Szintén definiáljuk az inverz függvényt, amelyet $\varphi^{-1}(\alpha)$ -val jelölünk.

Könnyen igazolható, hogy a hivatkozási függvény az attribútumokat ekvivalencia-osztályokba sorolja. A továbbiakban jelöljük α attribútum ekvivalencia-osztályát $\|\alpha\|$ -val, amelyet az α lezárásának nevezünk. A hivatkozási függvény az adatok használatát leíró egyszerű matematikai konstrukció, hiszen pontosan meghatározza, hogy szemantikailag helyesen mely értékeket milyen másik értékekkel lehet illeszteni.

Vegyük észre, hogy a hivatkozási függvénnyel rendelkező adatbázisban az attribútumértékek és az individuumok halmaza nem válik el egymástól, azaz $V = I$. Az ilyen adatbázisokat a továbbiakban *(V)ISA-modell*nek fogjuk nevezni. Könnyen belátható, hogy minden adatbázishoz létezik egy *(V)ISA-modell*, amely pontosan az adatbázis jellemző használatát írja le. Az egyedi jellemzőket a transzformáció során külön sémába érdemes felvenni, hiszen ennek és csak ennek természetes kulcsa a megfelelő azonosítókód.

3. Definíció (Általános birtokos szerkezet). *Legyen adva egy $DB = \langle V, I, S, A \rangle$ $(V)ISA$ -modell egy φ hivatkozási függvénnyel. Az $X \Rightarrow Y$ birtokos szerkezetet általánosnak nevezük akkor és csak akkor, ha az alábbi állítások egyikét igazá teszi:*

- 1.) $X \subseteq I$ és $Y \subseteq I$ (pl. Bizet Carmenje)
- 2.) $X \subseteq I$ és $Y \subseteq S$ (pl. Shakespeare drámái)
- 3.) $X \subseteq I$ és $Y \subseteq A$ (pl. Edit címe)
- 4.) $X \subseteq S$ és $Y \subseteq I$ (pl. kazánok Rolls Royce-a)
- 5.) $X \subseteq S$ és $Y \subseteq S$ (pl. könyvek szereplői)
- 6.) $X \subseteq S$ és $Y \subseteq A$ (pl. vállalatok címei)

Az általános birtokos szerkezetek a leggyakoribbak a természetes nyelvekben. Általában elmondható, hogy attribútum, jellemzőleírás nem szerepel birtokosként a mondatokban – és éppen emiatt ki is merítettük az összes kombinációs lehetőséget, már ami a relációs adatbázisok kifejezőerejét illeti.

Vizsgáljuk meg alaposabban a 3. definíció 4-6. kritériumait! A birtokos szerepben levő séma megnevezése az adott fogalom általános értelmére utal. Bár a séma lényegében a benne található individuumok, példányok összességével matematikailag azonosan kezelhető, a két „leírás” azonban nem feltétlenül azonos. Amikor például könyvek szereplőiről beszélünk, nem konkrétan egy, de nem is az összes könyvről jelentünk ki valamit. Ráadásul, hogy mi legyen séma és mi individuum azt sokszor a szemlélődő, a modellező nézőpontja határozza meg, nem lehet minden esetben éles különbséget tenni e kettő között. Az azonban elmondható, hogy a séma bizonyos individuumok valamilyen absztrakcióját jelenti, de alapvetően maga is egy individuum egy másik nézőpontból, más viszonyokat tekintve, azaz $S \subseteq I$. Tehát az absztrakciós szinteknek megfelelően hierarchiát definiálhatunk individuumok között, amelyben bár jellemzően csak a hierarchia alján levő elemeket (leveleket) szokás individuumnak, a felettes rétegek elemeit sémának nevezni, de a hierarchiát lényegében ugyanolyan elemek alkotják.

4. Definíció (Speciális birtokos szerkezet). *Legyen adva egy $DB = \langle V, I, S, A \rangle$ $(V)ISA$ -modell egy φ hivatkozási függvénnyel. Az $X \Rightarrow Y$ birtokos szerkezetet speciálisnak nevezük akkor és csak akkor, ha $X \subseteq I$.*

3 Birtokos szerkezetek szemantikájáról

Az általános és a speciális birtokos szerkezetek csak a kifejezésben szereplő elemekre vonatkozóan tesznek bizonyos fokú megszorítást, ugyanakkor nem jelennek meg bennük a szemantikára vonatkozó leírások. A szemantika meghatározásához azonban

először szükségünk van az érvényesség kritériumára is, hiszen nem minden birtokos szerkezet mondható helyesnek [5; 17; 21].

5. Definíció (Birtokos szerkezet érvényessége). Legyen $\Pi : A \times A$ egy reláció a $DB = \langle V, I, S, A \rangle$ (V)ISA-modell felett. Ha $\alpha, \beta \in A$ attribútumokhoz létezik olyan $s \in S$ séma, amely tartalmazza $\|a\| \cap \|b\|$ valamely nem üres részhalmazát, akkor $\Pi(\alpha, \beta)$ akkor és csak igaz, ha α minden kitöltésére értelmes speciális birtokos kifejezést kapunk.

Például a $\Pi(\text{személynév}, \text{születési idő})$ érvényes kifejezés, de az ellentéte nem az. Meg kell jegyeznünk azt is, hogy az érvényesség nyelvfüggő reláció, hiszen vannak olyan kapcsolatok, amelyek egyes nyelvekben, birtokos szerkezetben előfordulhatnak, más nyelvekben viszont nem [6; 21]. Gondoljunk csak arra, hogy az angolban használatos anyagleírások (pl. *ring of gold*) a magyarban birtokos szerkezetben nem léteznek (lásd még [7]).

Megvalósítást tekintve az érvényességi relációnak az adatbázisban egy két attribútummal rendelkező sémával ábrázolhatjuk, és azt mondhatjuk, hogy a reláció akkor és csak akkor érvényes, ha az érvényességi relációnak megfelelő sémában megtalálható az adott attribútumpáros.

6. Definíció (Birtokos szerkezet szemantikája). Legyen $DB = \langle V, I, S, A \rangle$ egy (V)ISA-modell, amelynek a hivatkozási függvénye \emptyset , és legyen $X \Rightarrow Y$ egy tetszőleges speciális birtokos kifejezés. Vezessük be továbbá az alábbi jelöléseket:

- Az $\alpha \mid s$ jelölje azt, hogy $s \in S$ séma rendelkezik az $\alpha \in A$ attribútummal.
- A $\Sigma : 2^A \rightarrow 2^S$ egy olyan leképezés, amely az attribútumhalmazokhoz hozzárendeli azt a maximális sémahalmazt, amelynek elemeire igaz az, hogy az attribútumhalmaz legalább egy elemét tartalmazza.
- A $\sigma : I \rightarrow 2^A$ egy olyan leképezés, amely minden individuumhoz meghatározza azt a legnagyobb attribútumhalmazt, amelynek elemeire igaz az, hogy ott az individuum értéként szerepelhet. Azaz σ meghatározza azon attribútumok lezártjainak unióját, ahol egy adott individuum előfordulhat.
- Jelölje $\gamma \alpha$ $\sigma(Y)$, a $\varphi(\kappa(Y))$, illetve a $\|Y\|$ kifejezéseket, ha Y rendre individuum, séma, illetve attribútum.

Az adott jelölések és feltételek mellett, ha léteznek $\alpha \in \sigma(X)$, $\beta \in \gamma$ attribútumok úgy, hogy $\exists s \in \Sigma(X) \cap \Sigma(\gamma)$ és $\alpha, \beta \mid s$, továbbá $\Pi(\alpha, \beta)$ igaz, akkor $X \Rightarrow Y$ nem más, mint a β attribútum értékei (individuumai), feltéve, hogy X egyetlen elemből álló halmaz. Máskülönben $X \Rightarrow Y$ az összes $\chi \Rightarrow Y$, $\chi \in X$, egyelemű birtokos szerkezet uniója.

A definíció alapján megállapítható, hogy a szemantika nem függ közvetlenül az adatbázis szerkezetétől, azaz az α, β és s paraméterek szabad változók, azok kizárólag a konkrét helyi adatbázis tartalmának függvényében lehet, illetve kell meghatározni. A megoldás tehát elég hatékony abban az értelemben, hogy a szemantikai feloldáshoz elegendő a megfelelő hármast megkeresni. Egy ilyen kényszerfeloldás könnyen megvalósítható például korlátos logikai programozással. Másfelől azonban szükség van az univerzális szerkezetű (V)ISA-modell leképezésére a helyi, keresni kívánt adatbázisra. Magát a leképezést azonban mindig a keresendő adatbázis adminisztrátorai valósítják meg és felügyelik.

A megközelítés újdonságereje éppen ezen alapszik: figyelembe veszi, hogy az Interneten található adatbázisok heterogén szerkezetűek, és csak arra van szükségünk a szemantikai információk kinyerésakor, hogy a megfelelő, értelmes, lekérdezésekben használt navigációkat, útkifejezéseket (path expressions) ábrázoljuk a lehető legtermészetesebb formában. Ez a leírás nem elhagyható, ugyanis ha az adatok összekapcsolásának mikéntje, módja nem adott vagy alulhatározott, akkor még ember számára sem egyértelmű, hogy hogyan kellene használni őket.

Hozzáteesszük azt is, hogy a természetes kulcsok bármikor helyettesíthetők a hagyományos, adatbázisokban használt (elsődleges) kulcsokkal, hiszen a definíciók és a szemantikai leírás nem használta a természetes kulcsnak egyetlen speciális tulajdonságát sem. Azonban azt is látni kell, hogy ebben az esetben a nyelvi többértelműségek idejekorán egyértelműsödhetnek, és nincs ilyenkor garancia a helyes döntésre. Ráadásul az olyan kifejezések esetében, mint a számok vagy a dátumok sokkal bonyolultabb szerkezetet kellene bevezetni.

1. Algoritmus ((V)ISA-algoritmus). *Legyen adva egy $DB = \langle V, I, S, A \rangle$ (V)ISA-modell egy φ hivatkozási függvénnyel. Az alábbi algoritmus, amelyet (V)ISA-algoritmusnak nevezünk, a speciális $X \Rightarrow Y$ birtokos szerkezetet dolgozza fel.*

```

1. function VISA( X, Y ) returns SQL
2. begin
   if ( isAttrib(Y) ) $gamma = //Y//
3.   else if ( isSchema(Y) ) $gamma =  $\varphi(\kappa(Y))$ 
   else $gamma =  $\sigma(Y)$ ;
4. find(  $\alpha$ ,  $\beta$ , s );
5. $add = isAttrib(Y) ? ' $\beta = Y$  AND ' : '';
6. if ( isSet(X) )
   return 'SELECT  $\beta$  FROM s WHERE $add  $\alpha = X$ '
   else {
7.   $Z = ( X is a V  $\Rightarrow$  W ) ? VISA(V, W) : X;
8.   if ( isSchema(X) )
     return 'SELECT  $\beta$  FROM s
           WHERE $add  $\alpha$  IN ( SELECT  $\kappa(X)$  FROM X )'
   else
     return 'SELECT  $\beta$  FROM s WHERE $add  $\alpha$  IN ( $Z )';
   }
9. end;
```

Az algoritmusban a 4. sor azt jelenti, hogy a 6. definíciónak megfelelő hármassokat kell megkeresni, azaz rejtve itt jelenik meg a birtokos szerkezet érvényességi relációja.

4 A (V)ISA-algoritmus működés közben

Nézzük meg, hogy az algoritmus hogyan működik 1. táblázat elemeire! Vegyük például a „Bizet Carmenje” kifejezést. Tételezzük fel, hogy van egy $\alpha =$ szerző, $\beta =$ műcím attribútum az opera sémában, azaz $opera \in \Sigma(\sigma(\|Bizet\|)) \cap$

$\Sigma(\sigma(\text{llCarmenll}))$. Mivel Carmen egy individuum, így az $\$add$ változó értéke műcím = 'Carmen'. Vagyis az algoritmus az alábbi kimenetet állítja elő:

```
SELECT műcím FROM opera
WHERE műcím = 'Carmen' AND szerző = 'Bizet'
```

Az eredmény megegyezik az 1. táblázatban látottakkal. Hasonlóan, a „vállalat vezető”-t illetően:

```
SELECT vezető FROM vállalat
WHERE vezető IN ( SELECT vezető FROM vállalat )
```

Az algoritmus nem a legegyszerűbb választ találja meg, mindazonáltal ekvivalens az 1. táblázattal szereplő lekérdezéssel.

Összetett vagy láncolt birtokos szerkezeteket, mint amilyen a „*Petőfi anyjának a neve*” kifejezés volt, az algoritmus az alábbi módon fejt ki. Először is megpróbálja feldolgozni a teljes (*Petőfi* \Rightarrow *anya*) \Rightarrow *személynév* láncolatot. A fentebb már bemutatott lépések után az algoritmus a (*Petőfi* \Rightarrow *anya*) kifejezést dolgozza fel (7. sor). Ennek eredményeképpen létrejön a

```
SELECT anya FROM leszármazás
WHERE gyermek = 'Petőfi'
```

amelyet a teljes kifejezés megoldásába illeszt:

```
SELECT nőszemélynév FROM nőszemély
WHERE nőszemélynév IN ( SELECT anya FROM leszármazás
                        WHERE gyermek IN ( 'Petőfi' ) )
```

Felmerülhet a kérdés, hogy az algoritmus hogyan következteti ki, hogy a leszármazás sémában Petőfinek gyermeknek kell lennie és nem anyának? Az algoritmus leírásában látszólag nincs akadálya annak, hogy ilyen alternatív megoldások is szülessenek, azonban a 4. sorban elrejtett *II*-relációnak köszönhetően ez nem fordulhat elő, hiszen az kizárja a *II(anya, anya)* lehetőségét.

5 Összefoglalás

A birtokos szerkezetek feldolgozását lehetővé tevő algoritmusok és matematikai modellek hiányoznak a számítógépes nyelvészeti irodalomból – külön tekintettel a magyar nyelvet illetően. A cikkben az alapvető problémákat és egy lehetséges megoldást mutattunk be, amely a természetes nyelvi birtokos szerkezeteket – ha adva vannak a szintaktikus szerepek – formális nyelvre, SQL-re képes fordítani.

Algoritmusunk univerzális abban az értelemben, hogy a magyar birtokos szerkeztípusokra jól működik, sőt, még a láncolásokat, az összetett birtokos szerkezeteket is képes feldolgozni, ugyanakkor vannak korlátai is. Mivel a megoldás során mindig adatbázisból, rögzített sémaszervezet és relációk mentén nyerjük ki az információkat, így a (V)ISA-algoritmus nem tudja kezelni a metaforikus (pl. filmek Mekkája), az idiomatikus (pl. a jövő embere), valamint az olyan komplex, jelentéstömörítő kijelentéseket, amelyeket csak mélyebb emberi tudással dolgozhatóak fel (pl. mérkőzés győztese).

Beláttuk, hogy a megoldáshoz nem feltétlenül szükséges külön ontológiai tudás, a sémaszerkezetből, az elnevezési konvenciókból, valamint a nyelvfüggő érvényességi relációból a legfontosabb tudáselemek kinyerhetőek. Az algoritmus működését példákkal is illusztráltuk.

Bibliográfia

1. Androutsopoulos, I., Ritchie, G.D., Thanisch, P.: Masque/SQL – an efficient and portable natural language query interface for relational databases, in: P. Chung, G. Lovegrove, M. Ali (Eds.), Proc. of IEA/AIE 93 Conference, IEA/AIE Conferences, Edinburgh, Gordon and Breach Publishing (1993) 327–330.
2. Androutsopoulos, I., Ritchie, G. D., Thanisch P.: Natural language interfaces to databases – an introduction, *Journal of Natural Language Engineering* **1**(1) (1995) 29–81.
3. Answerbus. <http://www.answerbus.com/>
4. Askjeeves. <http://www.ask.com/>
5. Barker, C.: Possessive descriptions, PhD thesis, University of Carolina, Santa Cruz, Department of Linguistics (1995)
6. Barker, C., Dowty, D.R.: Non-verbal thematic proto-roles. In Schafer, A., ed.: Proc. of NELS 23 Conference. North-Eastern Linguistics Conferences, Amherst, Massachusetts, GLSA Publications (1992) 49–62
7. Chisarik, E., Payne, J.: Modelling possessor constructions in lfg: English and hungarian. In Butt, M., King, T., eds.: Proc. of the LFG01 Conference. International Lexical-Functional Grammar Conferences, Hongkong, Stanford CSLI Publications (2001) 49–62
8. Google. <http://www.google.com/>
9. Ionaut. <http://www.ionaut.com:8400/>
10. Jensen, P.A., Vikner, C.: The english prenominal genitive and lexical semantics. Workshop on the Semantics/Syntax of Possessive Constructions, Amherst, Massachusetts (2002) Invited paper.
11. Kardkovács, Zs.T.: On the Transformation of Sentences with Genitive Relations to SQL Queries. In Montoyo, A., ed.: Proc. of 10th NLDB Conference. Alicante, Spain, Springer-Verlag Publishing (2005) 10–20
12. Katz, B.: Using English for Indexing and Retrieving, Vol. 1 of Artificial Intelligence at MIT: Expanding Frontiers, MIT Press (1990) 134–165
13. Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J.J., Marton, G., McFarland, A.J., Temelkuran, B.: Omnibase: A uniform access to heterogeneous data for question answering, in: B. Andersson, M. Bergholtz, P. Johannesson (Eds.), Proc. of NLDB 2002, Vol. 2553 of Lecture Notes in Computer Science, Stockholm, Springer-Verlag (2002) 230–234
14. Katz, B., Lin, J.L.: Start and beyond, in: Proc. of SCI 2002, Vol. XVI of World Multiconference on Systemics, Cybernetics and Informatics (2002)
15. Meng, X., Wang, S.: Overview of a chinese natural language interface to databases: Nchiql. *International Journal of Computer Processing of Oriental Languages* **14** (3) (2001) 213–232
16. Popescu, A.-M., Etzioni, O., Kautz, H.: Towards a theory of natural language interfaces to databases, in: W. Johnson, E. Andre, J. Domingue (Eds.), Proc. of the 8th IUI 2003, International Conferences on Intelligent User Interfaces, Miami, ACM Press (2003) 149–157
17. Rappaport, G. C.: The syntax of possessors in the nominal phrase: Drawing the lines and deriving the forms, Possessives and Beyond: Semantics and Syntax, Amherst, Massachusetts, GLSA Publications (2005) 223–262

18. Reis, P., Matias, J., Mamede, N.: Edite: A natural language interface to databases – a new perspective for an old approach, in: Proc. of ENTER'97, Information and Communication Technologies in Tourism, Edinburgh, Springer-Verlag (1997) 317–326
19. Start. <http://www.ai.mit.edu/projects/infolab/>
20. Storto, G.: Possessives in context. Possessives and Beyond: Semantics and Syntax, Amherst, Massachusetts, GLSA Publications (2005) 59–86
21. Storto, G.: Possessives in context – issues in the semantics of possessive constructions, PhD thesis, University of California, Los Angeles, Linguistics (2003)
22. Stratica, N., Kosseim, L., Desai, B. C.: A natural language processor for querying cindi, in: V. Milutinovic (Ed.), Proc. of SSGRR 2002, International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet, L'Aquila, Italy (2002)