MaSzeKer logikai alapok

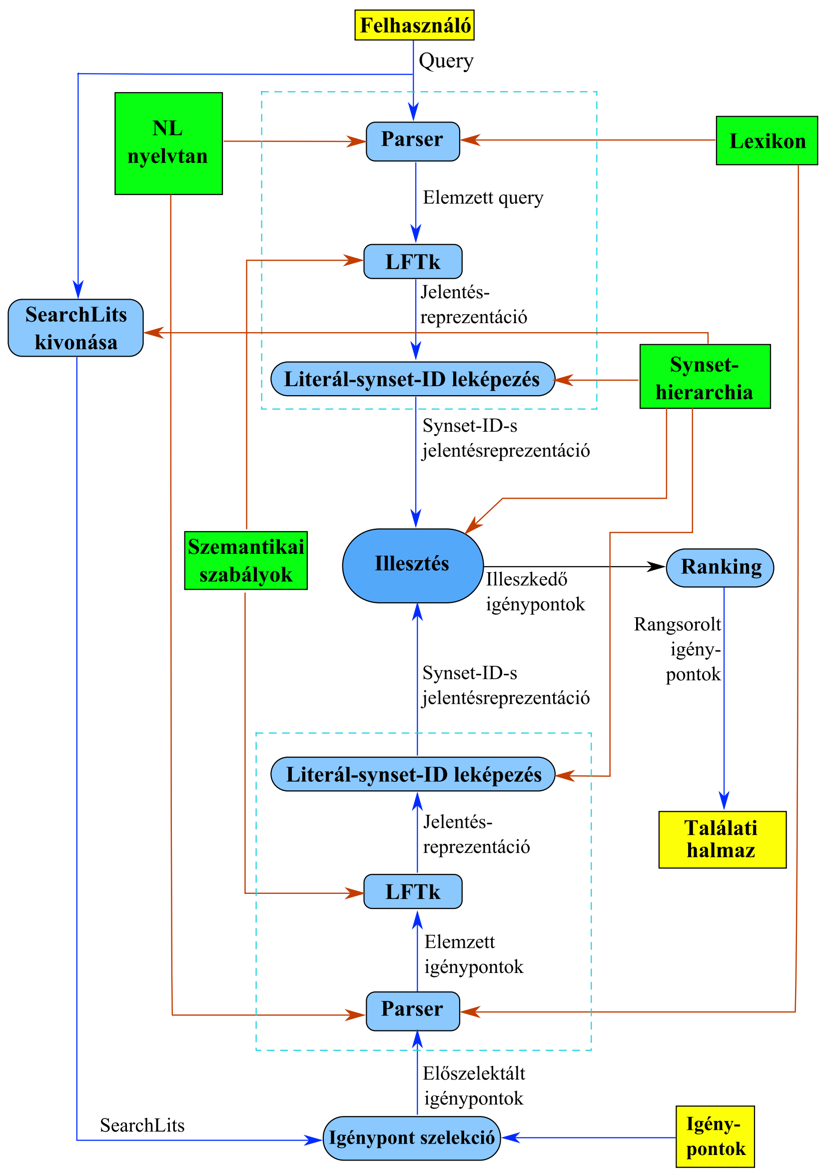
Varasdi Károly Simonyi András

2010. július 14.

Tartalomjegyzék

1. Bevezetés

1.1. Baseline rendszermodulok és adatfolyam



1. ábra. MaSzeKer baseline rendszer: modulok és adatfolyam

A MaSzeKer rendszerben lévő modulokat és a közöttük fennálló kapcsolatokat az 1. ábra mutatja be. Jelen tanulmányban vázoljuk az ábra értelmezéséhez szükséges fogalmi és logikai hátteret.

2. Természetes nyelvi szavak és kifejezések

2.1. Literálok

A feldolgozandó álló természetes (jelenleg: angol) nyelvi szöveg alapegységekből, *literálelőfordulásokból* áll.

1. definíció

Literálelőfordulás

Literálelőfordulás alatt az az igénypontokban valamint a kontrollált természetes nyelven (*CNL*) megfogalmazott keresőkifejezésben (*query*) előforduló olyan sztringeket, illetve olyan több sztringből álló kifejezéseket értünk, amelyek céljainkat tekintve egy fogalom kifejezéseinek tekinthetők. Például: „aspirin", „polyvinyl acetate phthalate".

Az így értelmezett literálelőfordulások az angol nyelv szó- és kifejezéskészletéhez tartozó fogalomjelölő elemeinek az instanciái. E fogalomjelölők (*literálok*) között különböző fogalmi–logikai relációk állnak fent, amelyek közül kettőt kiemelten kezelünk: a *szinoníma*, illetve az *alá–fölé rendeltségi* viszonyokat. Az angol nyelvben létező literálok ezen relációk mentén hálózatot, *literálhálót*, alkotnak.

2. definíció

Literálháló

Literálok olyan wordnet- vagy ontológiaszerűen rendezett nemüres halmaza, amelyben literáloknak (legalább) az alá–fölé rendeltségi, valamint a szinoníma viszonyai ki vannak jelölve. Más szóval, egy literálháló egy $\langle W, \preceq, \sim\rangle$ rendezett hármas, ahol W a literálok nemüres halmaza, és $\preceq, \sim \subseteq W \times W$. E relációkra három feltételt követelünk meg:

I. A $\preceq$ „generikus" reláció reflexív és tranzitív előrendezés. Ha $l_1, l_1 \in W$ esetén $l_1 \preceq l_2$, akkor azt mondjuk, hogy l_1 *specifikusabb*, mint l_2 (l_2 *általánosabb*, mint l_1).

II. A $\sim$ szinonima reláció ekvivalenciareláció.

III. Ha $l \sim l'$, akkor $l \preceq l'$ és $l' \preceq l$.

A literálhálóban levő literálok jelentéstani szempontból egyértelműsítettek, és ennek következtében ugyanolyan írásképű literálok a háló különböző pontjain is előfordulhatnak (és ekkor természetesen különböző szinonímákkal illetve alá–fölé rendeltekkel rendelkezhetnek).
Az alábbiakban néhány, a literálhálóhoz kapcsolódó általános fogalmat definiálunk.

3. definíció

Literál synsetje
Az $l \in W$ literál $\sim$ szerinti ekvivalenseinek halmazát (l „synsetjét”) $l^{\sim}$ jelöli:

$$l^{\sim} \stackrel{\text{def}}{=} \{l' \in W \mid l \sim l'\}\,.$$

A $\langle W, \sim, \preceq \rangle$ literálhálóban lévő synsetek halmazát $\textsc{Syns}$ jelöli:

$$\textsc{Syns} \stackrel{\text{def}}{=} \bigcup_{l \in W} \{l^{\sim}\}$$

4. definíció

Synsetek generikus viszonyai
A W literálhalmazon értelmezett $\preceq$ relációt kiterjesztjük magukra a synsetekre is a következő módon:

$$l^{\sim} \preceq l'^{\sim} \stackrel{\text{def}}{\Longleftrightarrow} l \preceq l'.$$

A synsetekre való hivatkozás megkönnyítésének céljából megtehetnénk, hogy minden sysnsetből kiválasztunk egy reprezentáns elemet, amely az egész synsetet képviseli és az ugyanabban a sysnsetben található literálokhoz ezt a reprezentánst rendelhetnénk absztrakt azonosítóként. Ehelyett az alábbi 5. definícióban egy másik utat választunk, és a literálokhoz egy természetes szám azonosítót rendelünk oly módon, hogy a hozzárendelés tartsa tiszteletben a literál szinonímaviszonyait.

5. definíció

Synset-azonosító függvény

Legyen $\textsc{id}_{\#}$ a $\langle W, \sim, \preceq \rangle$ literálháló synsetjeit a természetes számokra leképező függvény:

$$\begin{aligned} \textsc{id}_{\#} &: \textsc{Syns} \to \mathbb{N} \\ l^{\sim} &\mapsto \textsc{id}_{\#}(l^{\sim}) \end{aligned}$$

6. definíció

Literál- és literálhalmaz specializáltjai

Legyen $H \subseteq W$ literálok egy halmaza. H specializáltjainak halmaza, $\downarrow H$, a következő literálhalmaz:

$$\downarrow H \stackrel{\text{def}}{=} \{ l \in W \mid \exists l' \in H : l \preceq l' \} .$$

Ennek részesete az l literál specializáltja:

$$\downarrow l \stackrel{\text{def}}{=} \downarrow \{l\}.$$

A 2. definíció III. kikötése miatt $\downarrow l$ tartalmazza l szinonímáit is.

3. Jelentésreprezentáció

3.1. $\mathcal{L}_{rep}$ és $\mathcal{L}^{\#}_{rep}$

A keresőkifejezés és az igénypont szemantikai tartalmát az elsőrendű logika egy töredékét képező reprezentációs nyelvben ($\mathcal{L}_{rep}$) ábrázoljuk. Az $\mathcal{L}_{rep}$ nyelv ember által jól olvasható formális nyelv, mert a természetes nyelvi literálok és $\mathcal{L}_{rep}$ konstansai között létrehozható egy a literál szófaján alapuló megfeleltetés.

A gépi feldolgozás során azonban nem a konkrét literálok, hanem azok szinonímaosztályai relevánsak. Magát a keresés közbeni mintaillesztést is ennek figyelembe vételével definiáljuk majd. A 3.1.4. pontban bevezetünk $\mathcal{L}_{rep}$ mellett egy olyan $\mathcal{L}^{\#}_{rep}$ nyelvet is, amely logikai értelemben semmi újdonságot nem tartalmaz, de predikátumkonstansai literálok synset-azonosítói (azaz természetes számok).

3.1.1. Az $\mathcal{L}_{rep}$ jelentésreprezentációs nyelv

7. definíció

$\mathcal{L}_{rep}$ terminusai

$$\begin{aligned}
\mathcal{L}_{rep} =& \langle LC, Con, Var, Term, Form\rangle, \text{ahol}\\
LC =& \{(,), \neg, \wedge, \vee, \exists, =\},\\
Con =& Con_{ind} \cup Con_{pred},\\
Con_{ind} =& \bigcup_{i\in\mathbb{N}} \{c_i\},\\
Con_{pred} =& \bigcup_{k\in\mathbb{N}} Con^{(k)}_{pred},\\
Con^{(k)}_{pred} =& \bigcup_{i\in\mathbb{N}} \{F^{(k)}_i\},\\
Var =& \bigcup_{i\in\mathbb{N}} \{x_i\},\\
Term =& Var \cup Con_{ind}.
\end{aligned}$$

y

A példákban, a szokásoknak megfelelően, értelemszerűen lazítjuk ezt a szintaxist, és egyéb betűket is használunk: változónak eszerint az x, y, z, stb., konstansnak az a, b, c, stb., predikátumjelnek pedig az F, G, H, stb. betűk és indexelt változataik is, sőt, konkrét literálok is használhatóak, amíg teljesen világos, hogy ezek milyen kategóriába tartoznak.

A literálok használatát az teszi lehetővé, hogy létezik egy *invertálható* syncon függvény, amely a SYNS halmazt átviszi a Con halmazba.

8. definíció

Synsethez tartozó $\mathcal{L}_{rep}$ konstans

$$\begin{aligned}
&\mathsf{syncon}\colon \textsc{Syns} \to Con\\
&l^{\sim} \mapsto \mathsf{syncon}(l^{\sim})\\
&\text{ha } l^{\sim} \neq l'^{\sim} \text{ akkor } \mathsf{syncon}(l^{\sim}) \neq \mathsf{syncon}(l'^{\sim})
\end{aligned}$$

A $\mathcal{L}_{rep}$ nyelv formulái az alábbi 9. definícióban leírtak szerint épülnek fel.

9. definíció

$\mathcal{L}_{rep}$ formulái

$\mathcal{L}_{rep}$ formuláinak halmaza a legkisebb olyan *Form* halmaz, amely kielégíti az alábbi induktív klózokat:

$$\text{ha } t_0, \ldots, t_{k-1} \in \mathit{Term}, F^{(k)} \in \mathit{Con}_{pred}, \text{ akkor}$$
$$F^{(k)}(t_0, \ldots, t_{k-1}) \in \mathit{Form} \tag{i}$$
$$\text{ha } t_1, t_2 \in \mathit{Term}, \text{ akkor } (t_1 = t_2) \in \mathit{Form} \tag{ii}$$
$$\text{ha } \varphi \in \mathit{Form}, \text{ akkor } \neg(\varphi) \in \mathit{Form} \tag{iii}$$
$$\text{ha } \varphi, \psi \in \mathit{Form}, \text{ akkor } (\varphi \wedge \psi) \in \mathit{Form} \tag{iv}$$
$$\text{ha } \varphi, \psi \in \mathit{Form}, \text{ akkor } (\varphi \vee \psi) \in \mathit{Form} \tag{v}$$
$$\text{ha } \varphi \in \mathit{Form}, x \in \mathit{Var}, \text{ akkor } \exists x(\varphi) \in \mathit{Form} \tag{vi}$$

A felesleges zárójelek elhagyhatósága céljából precedenciát rendelünk az egyes operátorokhoz.

Precedenciaviszonyok: A $\neg$ és $\exists$ operátorok erősebben kötnek, mint az $\wedge$, amely erősebben köt, mint a $\vee$.

3.1.2. $\mathcal{L}_{rep}$ formuláinak egyszerűsített alakja

$\mathcal{L}_{rep}$ egy φ formuláját egyszerűsített alakban is ábrázolhatjuk, a következő szabályok szerint.

I. φ-ből elhagyjuk az egzisztenciális kvantorokat,

II. φ-ből elhagyjuk a konjunkció jelét,

III. a kapott atomi formulákat egy halmazba gyűjtjük.

Ha a φ formula egzisztenciálisan kötött változói között nem volt átfedés (azaz φ-ben nem volt $\exists x F(x) \wedge \ldots \wedge \exists x G(x)$ alakú rész), akkor az egyszerűsített reprezentáció ekvivalens az eredeti φ-vel. Ezt mindig el lehet érni a változók megfelelő átbetűzésével.

3.1.3. Keresőkifejezés és az igénypont $\mathcal{L}_{rep}$-be fordítása

A morfoszintaktikailag feldolgozott szöveget (keresőkifejezést vagy igénypontot) egy megfelelő modul („LFTk") lefordítja az $\mathcal{L}_{rep}$ reprezentációs nyelvbe. Az alább részletezendő okoknál fogva némi különbség van aközött, ahogy LFTk a keresőkifejezésen és az igényponton működik.

Az $\mathcal{L}_{rep}$ nyelv formuláira való fordítás célja a keresés során történő mintaillesztés, amelyben a szokásos unifikáció műveletét használjuk. Ez azonban felvet egy problémát, ha *mind* a keresőkifejezésből, *mind* pedig az igénypontból származó reprezentációt változókat tartalmazó formulákkal szeretnénk ábrázolni. A következő példa világossá teszi a problémát. Tegyük fel, hogy a keresőkifejezésből származó formula a már egyszerűsített ábrázolás szerint $\{F(x), G(x)\}$, az igénypontból származó pedig $\{F(y), G(z)\}$. Az illesztés során a keresőkifejezéshez tartozó formula első tagja, $F(x)$, unifikálódik az igénypont $F(y)$ formulájával, és eközben x és y kötése egybeejtődik. A második lépésben $G(x)$ unifikálódik $G(z)$-vel, és a változók egyesítése itt is sikeresen megtörténik ($x{=}y{=}z$). Ez azonban hibás találat, mert az igénypontból származó formula potenciálisan *különböző* objektumokról állította az F illetve G tulajdonságot, míg a keresőkifejezés azon objektumokra vonatkozik, amelyek *egyszerre* rendelkeznek ezekkel. Ezen a hagyományos unifikációfogalom keretein belül nem lehet segíteni, hiszen, ha a formulákban a megfelelő pozícióban változók vannak, ezek unifikálódását nem lehet megakadályozni.

A fenti probléma egyik lehetséges megoldása az, hogy az igénypontokat *változómentes* formulákká alakítjuk, azaz „helyfenntartó” konstansokat illesztünk a predikátumaik argumentumhelyeire, és változókat csak a keresőkifejezés fordításában engedünk meg. Ezzel megmarad az argumentumhelyek összeindexeltségi szerkezete, és hagyományos maradhat az unifikáció is.

3.1.4. Az $\mathcal{L}^{\#}_{rep}$ jelentésreprezentációs nyelv

A keresést megelőzően az $\mathcal{L}_{rep}$ nyelv formuláit lefordítjuk a szinonímaosztályokat is tükröző $\mathcal{L}^{\#}_{rep}$ nyelvbe. Az unifikáció művelete erre nyelvre vonatkoztatva lesz értelmezve. Az $\mathcal{L}_{rep}$ és $\mathcal{L}^{\#}_{rep}$ közötti fordítást egy **trans** függvény végzi, a következők szerint.[1]

[1] A syncon függvény definícióját ld. a 8 . definíciónál.

10. definíció

$\mathcal{L}_{rep}$-beli formulahalmaz $\mathcal{L}_{rep}^{\#}$-beli képe

I. Ha $F \in Con_{pred}$, akkor $\mathbf{trans}(F) = \mathrm{ID}_{\#}(\mathsf{syncon}^{-1}(F))$.

II. Ha $x \in Var$, akkor $\mathbf{trans}(x) \stackrel{\text{def}}{=} x$.

III. Ha $c \in Con_{ind}$, akkor $\mathbf{trans}(c) \stackrel{\text{def}}{=} \mathrm{ID}_{\#}(\mathsf{syncon}^{-1}(c))$.

IV. Ha $t_1, t_2 \in Term$, akkor $\mathbf{trans}(t_1 = t_2) \stackrel{\text{def}}{=} \mathbf{trans}(t_1) = \mathbf{trans}(t_2)$.

V. $\mathbf{trans}(F(t_1, t_2, \ldots, t_n)) \stackrel{\text{def}}{=} \mathbf{trans}(F)(\mathbf{trans}(t_1), \mathbf{trans}(t_2), \ldots, \mathbf{trans}(t_n))$.

VI. $\mathbf{trans}(\neg\varphi) \stackrel{\text{def}}{=} \neg\mathbf{trans}(\varphi)$.

VII. $\mathbf{trans}(\varphi \otimes \psi) \stackrel{\text{def}}{=} \mathbf{trans}(\varphi) \otimes \mathbf{trans}(\psi)$, ahol $\otimes \in \{\wedge, \vee\}$.

VIII. $\mathbf{trans}(\exists x\varphi) \stackrel{\text{def}}{=} \exists x\mathbf{trans}(\varphi)$.

IX. Ha $\Phi \subseteq Form$, akkor $\mathbf{trans}(\Phi) \stackrel{\text{def}}{=} \bigcup_{\varphi \in \Phi} \{\mathbf{trans}(\varphi)\}$.

4. Keresés

4.1. Kulcsszavas előszűrés

A keresés folyamata két fázisból áll. Az első, felszíni fázisban kulcsszavak segítségével leszűkítjük a szóba jöhető igénypontok halmazát egy kezelhető méretű halmazra. A kulcsszavas előszűrésről részletesen egy másik tanulmány szól, így annak mikéntjébe itt nem megyünk bele. Jelen tanulmányban csak a kulcsszavas előszűréshez használt literálok halmazának meghatározásáról ejtünk szót.

11. definíció

<u>*Előszűrő literálhalmaz*</u>

Legyen a $Q \in Form$ keresőkifejezésben szereplő individuum- és predikátumkonstansok halmaza $\mathsf{Consts}(Q)$. A Q konstansaihoz tartozó literálősök halmaza, $\mathsf{Lits}(Q)$, a következő:[a]

$$\mathsf{Lits}(Q) \stackrel{\text{def}}{=} \bigcup_{C \in \mathsf{Consts}(Q)} \mathsf{syncon}^{-1}(C).$$

Ekkor a Q-hoz tartozó előszűrő literálhalmaz, $\mathsf{SearchLits}(Q)$, a következőképpen definiálható:[b]

$$\mathsf{SearchLits}(Q) \stackrel{\text{def}}{=} \downarrow \mathsf{Lits}(Q).$$

[a] A syncon függvény definícióját ld. a 8. definíciónál.
[b] A $\downarrow$ definícióját ld. a 6. definíciónál.

4.2. Mintaillesztés

A keresési folyamat második, mélyszemantikai fázisát a keresőkifejezés jelenésreprezentációjának az igénypont jelentésreprezentációjával való illesztése jelenti, annak érdekében, hogy a keresőkifejezés által tartalmilag lefedett (azaz ahhoz adekvát) találatok megjeleníthetőek legyenek.

Az általunk használt illesztés fogalma néhány segédfogalmat feltételez; először ezeket definiáljuk.

4.2.1. Helyettesítés

12. definíció

<u>*Helyettesítés (szubsztitúció)*</u>

Helyettesítésen változó/konstans párok egy véges halmazát értjük:

$$\sigma = \{x_1/c_1, x_2/c_2, \ldots, x_n/c_n\},$$

ahol kikötjük, hogy ugyanahhoz a változóhoz nem tartozhat két különböző konstans párként σ-ban. A σ-ban szereplő x_i/c_i párra mint az x_i/c_i *változókötésre* hivatkozunk.

Vegyük észre, hogy a 12. definíció a logikai programozásból ismert helyettesítésfogalom egy egyszerűsített változata: ezt az egyszerűsítést az teszi lehetővé, amit a keresőkifejezés és az igénypont eltérő reprezentálásáról mondtunk a 3.1.3. pontban.

13. definíció

Behelyettesítés

Legyen $\aleph \in \mathit{Term} \cup \mathit{Form}\ \mathcal{L}_{rep}$ egy terminusa vagy formulája, és $\sigma = \{x_1/c_1, x_2/c_2, \ldots, x_n/c_n\}$ egy szubsztitúció. Azt a $\aleph'$ terminust (formulát), amelyet akkor kapunk, ha az $x_1, x_2, \ldots, x_n$ változókat $\aleph$-ben mindenütt egyidejűleg behelyettesítjük rendre $c_1, c_2, \ldots, c_n$-nel, $\aleph$ σ szerinti *behelyettesítésének* (instanciájának) nevezzük és $\aleph\{x_1/c_1, x_2/c_2, \ldots, x_n/c_n\}$-vel vagy egyszerűen $\aleph\sigma$-val jelöljük.

14. definíció

Üres helyettesítés

Az $\{\}$ *üres helyettesítés* olyan speciális helyettesítés, amely minden változót változatlanul hagy; tehát bármely $\aleph \in \mathcal{L}_{rep}$ formula vagy terminus esetén igaz, hogy $\aleph\{\} = \aleph$.

4.2.2. Unifikáció

Az alábbiakban definiáljuk a terminusok unifikálhatóságát. Ez nem tér el attól a szokásos unifikációfogalomtól, amely szerint két konstans csak akkor unifikálható, ha azonos, változó pedig akkor unifikálható konstanssal, ha a változó aktuális kötése alapján azonossá tehetők. A 3.1.3. pontban említett okoknál fogva nincs szükségünk a változók egymással való unifikációjának definiálására.

Minden unifikációs művelet valamilyen változókötési kontextushoz, bemeneti helyettesítéshez (σ_{in}) viszonyítva értelmeződik, értéke lehet siker (**success**) vagy sikertelenség (**fail**), és mellékhatásként megváltoztathatja a bemeneti helyettesítések halmazát (σ_{out}).

A literálháló rendezési viszonyait dinamikusan kívánjuk kihasználni. Ebből a célból az unifikáció fogalmát minimálisan kiterjesztjük, és bevezetjük az individuum- és predikátumkonstansokra vonatkozóan a *domináló unifikáció* műveletét.

15. definíció

Individuumkonstansok domináló egyesítése (d-unifikációja)

Legyen $\sigma_{\text{in}} = \{x_1/c_1, x_2/c_2, \ldots, x_n/c_n\}$ az aktuális bemeneti helyettesítés, és legyen $c, c' \in Con_{ind}$. Azt mondjuk, hogy c és c' a σ_{in} helyettesítés mellett *ebben a sorrendben* dominálóan egyesíthető, $c \stackrel{\succeq}{=} c'$, ha c literál ősei általánosabbak, mint c'-é:[a]

$$c \stackrel{\succeq}{=} c' = \begin{cases} \textbf{success} \text{ és } \sigma_{\text{out}} := \sigma_{\text{in}} & \text{ha } \mathsf{syncon}^{-1}(c') \preceq \mathsf{syncon}^{-1}(c) \\ \textbf{fail} \text{ és } \sigma_{\text{out}} := \sigma_{\text{in}} & \text{egyébként} \end{cases}$$

[a]Ld. a 4. definíciót.

16. definíció

Predikátumkonstansok domináló egyesítése (d-unifikációja)

Legyen $\sigma_{\text{in}} = \{x_1/c_1, x_2/c_2, \ldots, x_n/c_n\}$ az aktuális bemeneti helyettesítés. Legyen $F, F' \in Con^{(k)}_{pred}$. Azt mondjuk, hogy F és F' a σ_{in} helyettesítés mellett *ebben a sorrendben* dominálóan egyesíthető, $F \stackrel{\succeq}{=} F'$, ha F literál ősei általánosabbak, mint F'-é:

$$F \stackrel{\succeq}{=} F' = \begin{cases} \textbf{success} \text{ és } \sigma_{\text{out}} := \sigma_{\text{in}} & \text{ha } \mathsf{syncon}^{-1}(F') \preceq \mathsf{syncon}^{-1}(F) \\ \textbf{fail} \text{ és } \sigma_{\text{out}} := \sigma_{\text{in}} & \text{egyébként} \end{cases}$$

17. definíció

Változók és individuumkonstansok egyesítése (unifikációja)

Legyen $\sigma_{\text{in}} = \{x_1/c_1, x_2/c_2, \ldots, x_n/c_n\}$ az aktuális bemeneti helyettesítés. Ha $x \in Var$, és $c \in Con_{ind}$, akkor

$$x \doteq c = \begin{cases} \textbf{success} \text{ és } \sigma_{\text{out}} := \sigma_{\text{in}} & \text{ha } x\sigma_{\text{in}} = c \\ \textbf{fail} \text{ és } \sigma_{\text{out}} := \sigma_{\text{in}} & \text{ha } x\sigma_{\text{in}} \neq c \\ \textbf{success} \text{ és } \sigma_{\text{out}} := \sigma_{\text{in}} \cup \{x/c\} & \text{ha minden } c' \in Con_{ind}\text{-re: } x/c' \notin \sigma_{\text{in}} \end{cases}$$

18. definíció

Atomi formulák domináló egyesítése (d-unifikációja)

Legyen $\sigma = \{x_1/c_1, x_2/c_2, \ldots, x_n/c_n\}$ az aktuális helyettesítés. A $F(t_1, t_2, \ldots, t_n)$ atomi formula *dominálóan egyesíthető* az $F'(t'_1, t'_2, \ldots, t'_n)$ atomi formulával, jelekben: $F(t_1, t_2, \ldots, t_n) \stackrel{\succeq}{=} F'(t'_1, t'_2, \ldots, t'_n)$, ha predikátumkonstansaik *ebben a sorrendben* dominálóan egyesíthetőek és terminusaik egyesíthetőek, azaz $F \stackrel{\succeq}{=} F'$ és minden $1 \leq i \leq n$-re, $t_i \doteq t'_i$ értéke a σ helyettesítés mellett **success**.

19. definíció

Formulák tartalmi lefedése

Legyen $\sigma = \{x_1/c_1, x_2/c_2, \ldots, x_n/c_n\}$ az aktuális helyettesítés, és legyen adott egy $Q \in \mathit{Form}$ keresőkifejezés. A Q formula *tartalmilag le van fedve* a C igénypontból nyert $\varphi \in \mathit{Form}$ formulával, ha a σ helyettesítés mellett Q *minden* α atomi formulájához található φ-nek olyan β atomi formulája, hogy $\alpha \stackrel{\succeq}{=} \beta$ értéke **success**.

A keresési folyamat illesztési fázisának célja mindazon igénypontok azonosítása, amelyek a keresőkifejezést tartalmilag lefedik. Az egyes unifikációs kísérletek során létrejövő változókötések esetleges *backtrack*eléskor felbomlanak, és a rendszer újabb kötés kialakítására tesz kísérletet mindaddig, amíg erre lehetőség van. Az illesztési kísérlet végleg megbukik, ha ezek a lehetőségek mind sikertelennek bizonyulnak.

5. Példák

1. példa

$Q := \{F(x), G(x)\}$
$C := \{F(a), H(a), K(b)\}$
$G \succeq H$
$\sigma := \{\}$

$Q\sigma$-BÓL ILLESZTENDŐ	UNIFIKÁLANDÓ	SIKER?	σ
$\{F(x), G(x)\}$	$F(x) \stackrel{\succeq}{=} F(a)$	**success**	$\{x/a\}$
$\{G(a)\}$	$G(a) \stackrel{\succeq}{=} H(a)$	**success**	$\{x/a\}$
$\{\}$	—	**success**	$\{x/a\}$

2. példa

$Q := \{F(x), G(x)\}$
$C := \{F(a), G(b), H(a)\}$
$\sigma := \{\}$

$Q\sigma$-BÓL ILLESZTENDŐ	UNIFIKÁLANDÓ	SIKER?	σ
$\{F(x), G(x)\}$	$F(x) \doteq F(a)$	**success**	$\{x/a\}$
$\{G(a)\}$	$G(a) \doteq G(b)$	**fail**	$\{\}$
$\{F(x), G(x)\}$	$G(x) \doteq G(b)$	**success**	$\{x/b\}$
$\{F(b)\}$	$F(b) \doteq F(a)$	**fail**	$\{\}$
$\{F(x), G(x)\}$	—	**fail**	$\{\}$

6. A negáció kezelése

A baseline rendszerben az (implicit) egzisztenciális kvantorokkal lekötött, kizárólag atomi mondatokat tartalmazó konjunkciókkal reprezentálható keresőkifejezés-, illetve igénypont-reprezentációk mellett megengedünk olyan formulákat is, amelyek az előbbiekből *egyetlen* negáció hozzáadásával nyerhetők.

A negációmentes formulák

$$\exists x_1 \ldots \exists x_n (P_1^{(a_1)}(x_1^1, \ldots, x_{a_1}^1) \wedge \ldots \wedge P_k^{(a_k)}(x_1^k, \ldots, x_{a_k}^k))$$

logikai formája egy negáció beiktatásával a következőképpen változik:

$$\begin{aligned} \exists x_1 \ldots \exists x_r (P_1^{(a_1)}(x_1^1, \ldots, x_{a_1}^1) \wedge \ldots \wedge P_i^{(a_i)}(x_1^i, \ldots, x_{a_i}^i) \wedge \\ \neg \exists x_{r+1} ... \exists x_n (P_{i+1}^{(a_{i+1})}(x_1^{i+1}, \ldots, x_{a_{i+1}}^{i+1}) \wedge \ldots \wedge P_k^{(a_k)}(x_1^k, \ldots, x_{a_k}^k))) \end{aligned}$$

A negáció hatókörébe kerülő konjunkció atomi tagjai két típusba sorolhatók:

(i) $P^{(m)}(x_1, \ldots, x_m)$, ahol az $x_1, \ldots x_m$ változókat lekötő kvantorok *mindegyike* a negáció hatókörén *kívül* található,

(ii) $P^{(m)}(x_1, \ldots, x_m)$, ahol az $x_1, \ldots x_m$ változókat lekötő kvantorok *legalább egyike* a negáció hatókörén *belül* található.

Az (i) típusba tartozó tagok kihozhatók a negációt követő kvantorok hatóköréből. Ha ezt megtesszük, akkor a

$$\begin{aligned} \exists x_1 \ldots \exists x_r (P_1^{(a_1)}(x_1^1, \ldots, x_{a_1}^1) \wedge \ldots \wedge P_i^{(a_j)}(x_1^j, \ldots, x_{a_j}^j) \wedge \\ (\neg P_{j+1}^{(a_{j+1})}(x_1^1, \ldots, x_{a_{j+1}}^{j+1}) \vee \ldots \vee \neg P_l^{(a_l)}(x_1^l, \ldots, x_{a_l}^l) \vee \\ \neg \exists x_{r+1} ... \exists x_n (P_{l+1}^{(a_{l+1})}(x_1^{l+1}, \ldots, x_{a_{l+1}}^{l+1}) \wedge \ldots \wedge P_k^{(a_k)}(x_1^k, \ldots, x_{a_k}^k))) \end{aligned}$$

„kanonikus” logikai formához jutunk, amelyet erősen leegyszerűsítve (a maximális hatókörű egzisztenciális kvantorokat impliciten hagyva, és kikötve, hogy a Φ_i $(i \in \mathbb{N})$ metaváltozók atomi mondatokra utalnak) a

$$\Phi_1 \wedge \ldots \Phi_i \wedge (\neg\Phi_{i+1} \vee \ldots \vee \neg\Phi_j \vee \neg\exists x_1 \ldots \exists x_n(\Phi_{j+1} \wedge \ldots \wedge \Phi_k)) \tag{N}$$

formulával rövidíthetünk, ahol a $\Phi_1, \ldots, \Phi_k$ atomi formulák közül pontosan $\Phi_{j+1}, \ldots, \Phi_k$ tartalmazzák az $x_1, \ldots, x_n$ változók legalább egyikét.

A negációt is tartalmazó lekérdezések, illetve igénypontrészletek alapszintű kezelésének érdekében a tartalmi lefedés (19) relációt ki kell terjesztenünk az (N) formájú formulákra is, ezért némileg általánosítjuk azt.

(N)-t megvizsgálva látható, hogy olyan konjunkcióról van szó, amelynek egy kivétellel minden tagja atomi mondat, és az egyetlen nem atomi tag egy diszjunkció, amely negált atomi mondatokból, és egyetlen

$$\neg\exists x_1 \ldots \exists x_n(\Phi_1 \wedge \ldots \wedge \Phi_m) \tag{N'}$$

alakú tagból áll. Ennek megfelelően olyan lefedésdefiníciót próbálunk találni, amely az atomi, negált atomi, illetve (N') alakú formulák párjaira definiált, és tisztázza az ilyen elemekből diszjunkció és konjunkció segítségével felépülő formulák lefedési viszonyait is.

Az említett párokra vonatkozó definíciók megalkotásánál formálisan ugyan nem, de a motiváció céljára felhasználjuk azt az intuíciót, hogy egy Q keresőkifejezést akkor fed le tartalmilag egy φ formula, ha φ-ből és bizonyos (a rendszer számára hozzáférhető) „jelentésposztulátumokból” logikailag triviálisan következik Q. Ennek az elvnek alapján ha az $F^{(k)}$ és $G^{(k)}$ $\mathcal{L}_{rep}$-beli predikátumok közötti $\mathsf{syncon}^{-1}(F) \preceq \mathsf{syncon}^{-1}(G)$ viszonyt úgy interpretáljuk, hogy teljesül a $\forall x_1, \ldots, x_k(F(x_1, \ldots, x_k) \to G(x_1, \ldots, x_k))$ összefüggés, akkor azt mondhatjuk, hogy mivel

$$\forall x_1, \ldots, x_k(F(x_1, \ldots, x_k) \to G(x_1, \ldots, x_k)) \wedge F(f_1, \ldots, f_k) \models G(f_1, \ldots, f_k)$$

és

$$\forall x_1, \ldots, x_k(F(x_1, \ldots, x_k) \to G(x_1, \ldots, x_k)) \wedge \neg G(f_1, \ldots, f_k) \models \neg F(f_1, \ldots, f_k),$$

(minden f_i $(i \in \mathbb{N})$ individuumkonstans) ezért egy $F(f_1, \ldots f_k)$ formula akkor fed le tartalmilag egy $G(f_1, \ldots f_k)$ formulát, ha dominálóan egyesíthetők, míg $\neg F(f_1, \ldots f_k)$ akkor fedi le $\neg G(f_1, \ldots f_k)$-t, ha $G(f_1, \ldots f_k)$ lefedi $F(f_1, \ldots f_k)$-t.

Ugyanezen elv alkalmazható az (N') alakú formulák viszonyára is: Abból kiindulva, hogy

$$\neg\exists x_1 \ldots \exists x_n(\Phi_1 \wedge \ldots \wedge \Phi_m) \models \neg\exists x_1 \ldots \exists x_i(\Psi_1 \wedge \ldots \wedge \Psi_j)$$

pontosan akkor áll fent, amikor

$$\exists x_1 \ldots \exists x_i(\Psi_1 \wedge \ldots \wedge \Psi_j) \models \exists x_1 \ldots \exists x_n(\Phi_1 \wedge \ldots \wedge \Phi_m),$$

és figyelembe véve az egzisztenciális kvantorok elhagyásának a 3.1.2 szakaszban bevezetett konvencióját, azt mondhatjuk, hogy egy $\neg\exists x_1 ... \exists x_n(\Phi_1 \wedge \ldots \wedge \Phi_m)$ alakú formula akkor fed le tartalmilag egy $\neg\exists x_1 ... \exists x_i(\Psi_1 \wedge \ldots \wedge \Psi_j)$ alakút, ha $\Psi_1 \wedge \ldots \wedge \Psi_j$ lefedi $\Phi_1 \wedge \ldots \wedge \Phi_m$-et a 19. definíció értelmében.

Mivel a fenti három típusból *különböző* típusba tartozó formulák között nem áll fent triviális logikai következményreláció, ezért azt is kiköthetjük, hogy atomi formula csak atomit, negált atomi formula csak negált atomit, és végül (N') alakú formula csak (N') alakú formulát fedhet le tartalmilag.

Áttérve a konjunkciók, illetve diszjunkciók lefedési viszonyaira, itt a következményreláció következő tulajdonságai vehetők figyelembe (Ψ_i és Φ_i $(i \in \mathbb{N})$ atomi formulákra utalnak):

$$\Psi_1 \wedge \ldots \wedge \Psi_j \models \Phi_1 \wedge \ldots \wedge \Phi_k$$

pontosan akkor, ha minden Φ_s-re $(1 \leq s \leq k)$ van olyan Ψ_r $(1 \leq r \leq j)$, hogy $\Psi_r \models \Phi_s$, és

$$\Psi_1 \vee \ldots \vee \Psi_j \models \Phi_1 \vee \ldots \wedge \Phi_k$$

pontosan akkor, ha minden Ψ_r-re $(1 \leq r \leq j)$ van olyan Φ_s $(1 \leq s \leq k)$, hogy $\Psi_r \models \Phi_s$. Mint látható, a konjunkcióra vonatkozó összefüggés jól motiválja a a 19. definíciót, míg a diszjunkcióra vonatkozó állítás alapján diszjunkciók tartalmi lefedését úgy definiálhatnánk, hogy $\Psi_1 \wedge \ldots \wedge \Psi_j$ akkor fedi le $\Phi_1 \wedge \ldots \wedge \Phi_k$-t, ha minden Ψ_r-re $(1 \leq r \leq j)$ van olyan Φ_s $(1 \leq s \leq k)$, hogy Ψ_r tartalmilag lefedi Φ_s-t.

Bár az alább következő általánosított lefedésdefiníció főbb vonalaiban követi a következményreláció tulajdonságaiból eddig leszűrt elveket, néhány ponton eltér ezektől, többek között azért, mert eddigi megfontolásaink során nem vettük figyelembe sem a helyettesítések, sem az individuumkonstansok és változók speciális szerepét a domináló egyesítés definíciójában. Pl. problémát jelent a változók és az individuumnevek eltérő szerepe a keresőkifejezésekben és az igénypontreprezentációkban: mivel az igénypontreprezentációkban nem szerepelhetnek (implicite tág hatókörű egzisztenciális kvantorral lekötött) változók, ezért pl. az, hogy $\neg\Phi$ lefedi a $\neg\Psi$ keresőkifejezést általános esetben nem vezethető vissza közvetlenül arra, hogy Φ dominálóan egyesíthető Ψ-vel, mert a d-egyesíthetőség nem is definiált olyan terminuspárok esetében, amelyeknek a második tagja egy változó. A következő definíció, amely csakis az (N) alakú formulák lefedését definiálja (némi bonyolultság árán) megoldja ezt a problémát.

20. definíció

<u>*Általánosított tartalmi lefedés*</u>

Legyen $\sigma = \{x_1/c_1, x_2/c_2, \ldots, x_n/c_n\}$ az aktuális helyettesítés, és legyen adott egy $Q \in \mathit{Form}$ keresőkifejezés. A C igénypontból nyert $\varphi \in \mathit{Form}$ formula *tartalmilag lefedi* Q-t, ha a következő (egymást nem kölcsönösen kizáró) feltételek valamelyike teljesül:

I. Valamely $F^{(k)}$ és $G^{(k)}$ predikátumokra $Q = \neg F(f_1, \ldots, f_k)$, $\varphi = \neg G(g_1, \ldots, g_k)$, és $G(f_1, \ldots, f_k) \stackrel{\succ}{=} F(g_1, \ldots, g_k)$ fennáll a σ helyettesítés mellett.

II. Valamely $y_1, \ldots, y_l$ és $z_1, \ldots, z_m$ változókra, $\Psi_1, \ldots, \Psi_i$ és $\Phi_1, \ldots, \Phi_k$ atomi formulákra, valamint $\pi{=}\{z_1/d_1, \ldots, z_m/d_m\}$ helyettesítésre, ahol $d_1, \ldots, d_m$ Q-ban, φ-ben és σ-ban nem szereplő helypótló konstansok, $Q = \neg\exists y_1, \ldots, y_l(\Psi_1 \wedge \ldots \wedge \Psi_i)$, $\varphi = \neg\exists z_1, \ldots, z_m(\Phi_1 \wedge \ldots \wedge \Phi_k)$, és teljesül, hogy bármely φ-ben szereplő Φ atomi formulára van olyan Q-ban szereplő Ψ atomi formula, hogy a $\neg\Phi\pi$ formula tartalmilag lefedi $\neg\Psi$-t a σ helyettesítés mellett.

III. Mind Q, mind φ (N) alakú, olymódon, hogy

$$Q{=}\Psi_1{\wedge}\ldots\Psi_i{\wedge}(\neg\Psi_{i+1} \vee \ldots \vee \neg\Psi_j \vee \neg\exists y_1 ... \exists y_n(\Psi_{j+1}{\wedge}\ldots{\wedge}\Psi_k)),$$

$$\varphi{=}\Phi_1{\wedge}\ldots\Phi_r{\wedge}(\neg\Phi_{r+1} \vee \ldots \vee \neg\Phi_s \vee \neg\exists z_1 ... \exists z_m(\Phi_{s+1}{\wedge}\ldots{\wedge}\Phi_t)),$$

(bármelyik blokk lehet üres), és teljesül, hogy

(a) a Q konjunkció minden atomi Ψ tagjához van φ-nek olyan Φ atomi tagja, amelyre $\Psi \stackrel{\succ}{=} \Phi$ fennáll a σ helyettesítés mellett, és egyúttal

(b) ha a Q konjunkciónak van egy q nem atomi tagja is, akkor φ-nek is van egy olyan nem atomi f tagja, amelynek minden f' tagjához van q-nak olyan q' tagja, hogy f' lefedi q'-t a σ helyettesítés mellett.

A szakasz hátralevő részében megadunk négy eljárást, amelyek együttesen képesek eldönteni egy (N) alakú igénypontreprezentációról, hogy az tartalmilag lefed-e egy szintén (N) alakú query-t adott helyettesítés mellett.

Az első, NAlakúFedi eljárás két, bemenetként megadott (N) alakú formula lefedését vizsgálja, és felhasználja KTagFedi eljárást, amely az egyes konjunkciós tagokat hasonlítja össze. A KTagFedi eljárás az esetleges diszjunkciókat a DiszjunkcióFedi eljárással hasonlítja össze, ami a diszjunkció tagjait a DTagFedi eljárással kezeli. Ez utóbbiban van egy — könnyen kiküszöbölhető —

rekurzió, amelyet az tett szükségessé, hogy az egzisztenciális kvantort tartalmazó tagok maguk is atomi tagokat tartalmaznak, amelyeket a negált atomi formulákkal analóg módon kezelünk az összehasonlítás során (lásd az utolsó algoritmus 10. sorát).

Algoritmus 1 (N) alakú formulák lefedését vizsgáló algoritmus

---

```
procedure NAlakúFedi(Q, Φ, σ)  ▷ Q (N) alakú keresőkifejezésreprezentáció,
        Φ (N) alakú igénypontreprezentáció, σ tetszőleges helyettesítés. Az eljárás
        1-et ad vissza, ha Φ lefedi Q-t σ mellett, és 0-t, ha nem.
    for all conjunct q in Q do
        found ← 0
        for all conjunct φ in Φ do
            if KTagFedi(q, φ, σ) then
                found ← 1
            end if
        end for
        if found = 0 then
            return 0
        end if
    end for
    return 1
end procedure
```

Algoritmus 2 Konjunkciós tagok lefedését vizsgáló algoritmus

```
procedure KTagFedi(q, φ, σ)
    if q és φ atomi, és q ≿= φ fennáll σ helyettesítés mellett then
        return 1
    else if q és φ ¬Φ_1 ∨ ... ∨ ¬Φ_i ∨ ¬∃x_1...∃z_m(Φ_{i+1} ∧ ... ∧ Φ_n) alakú then
        return DiszjunkcóFedi(q, φ, σ)
    else
        return 0
    end if
end procedure
```

Algoritmus 3 Diszjunktív formulák lefedését vizsgáló algoritmus

```
procedure DiszjunkcióFedi(Q, Φ, σ)
    for all disjunct φ in Φ do
        found ← 0
        for all disjunct q in Q do
            if DTagFedi(q, φ, σ) then
                found ← 1
            end if
        end for
        if found = 0 then
            return 0
        end if
    end for
    return 1
end procedure
```

Algoritmus 4 Diszjunkciós tagok lefedését vizsgáló algoritmus

```
procedure DTagFedi(Q, Φ, σ)
    if Q = ¬F(f_1, ..., f_k), Φ = ¬G(g_1, ..., g_k), és G(f_1, ..., f_k) ≽= F(g_1, ..., g_k)
      fennáll a σ helyettesítés mellett then
        return 1
    else if Q = ∃x_1...∃x_i(Ψ_1 ∧ ... ∧ Ψ_j) és Φ = ¬∃z_1...∃z_m(Φ_1 ∧ ... ∧ Φ_n) then
        π ← egy olyan helyettesítés, amely Φ kötött változóihoz Q-ban, Φ-ben és
          σ-ban nem szereplő individuumkonstansokat rendel
        Φ′ ← (Φ_1 ∧ ... ∧ Φ_n)π
        for all conjunct φ in Φ′ do
            found ← 0
            for all conjunct q in Q do
                if DTagFedi(¬φ, ¬q, σ) then
                    found ← 1
                end if
            end for
            if found = 0 then
                return 0
            end if
        end for
        return 1
    else
        return 0
    end if
end procedure
```