



Parciális approximációs téren alapuló logikai rendszerek vizsgálata és ezek alkalmazásai a mesterséges intelligenciában

Egyetemi doktori (PhD) értekezés

Kádek Tamás

Témavezető:
Dr. Mihálydeák Tamás

Konzulens:
Dr. Várterész Magda

DEBRECENI EGYETEM
Természettudományi Doktori Tanács
Informatikai Tudományok Doktori Iskola
Debrecen, 2017

Ezen értekezést a Debreceni Egyetem Természettudományi Doktori Tanács Informatikai Tudományok Doktori Iskola Elméleti számítástudomány, adatvédelem és kriptográfia programja keretében készítettem a Debreceni Egyetem természettudományi doktori (PhD) fokozatának elnyerése céljából.

Nyilatkozom arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Debrecen, 2017.

.....

a jelölt aláírása

Tanúsítom, hogy Kádek Tamás doktorjelölt 2007-2010 között a Matematika- és Számítástudományok Doktori Iskola Informatika programjának keretében irányításommal végezte munkáját.

Debrecen, 2017.

.....

a konzulens aláírása

Tanúsítom, hogy Kádek Tamás doktorjelölt 2015-2017 között az Informatikai Tudományok Doktori Iskola Elméleti számítástudomány, adatvédelem és kriptográfia programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Nyilatkozom továbbá arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Az értekezés elfogadását javasolom.

Debrecen, 2017.

.....

a témavezető aláírása

PARCIÁLIS APPROXIMÁCIÓS TÉREN ALAPULÓ LOGIKAI RENDSZEREK
VIZSGÁLATA ÉS EZEK ALKALMAZÁSAI A MESTERSÉGES
INTELLIGENCIÁBAN

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében
az informatika tudományágban

Írta: Kádek Tamás okleveles programtervező matematikus

Készült a Debreceni Egyetem
Informatikai Tudományok Doktori Iskolája
(Elméleti számítástudomány, adatvédelem és kriptográfia programja)
keretében

Témavezető: Dr. Mihálydeák Tamás

A doktori szigorlati bizottság:

elnök:	Dr.	
tagok:	Dr.	
	Dr.	

A doktori szigorlat időpontja: 20... ..

Az értekezés bírálói:

Dr.	
Dr.	
Dr.	

A bírálóbizottság:

elnök:	Dr.	
tagok:	Dr.	
	Dr.	
	Dr.	
	Dr.	

Az értekezés védésének időpontja: 20... ..

Tartalomjegyzék

Előszó	1
1. Bevezetés	3
1.1. Általánosított approximációs keretrendszer	3
1.2. Approximációs tér az elsőrendű logikában	5
1.3. Elsőrendű logika közelítő mondatfunktorokkal	8
1.4. Állapottér a mesterséges intelligenciában	14
2. Logikai rendszerek vizsgálata	16
2.1. Parciális kvantormentes logika	16
2.1.1. A közelítő funktorokat tartalmazó kvantormentes nyelv	16
2.1.2. Interpretáció	17
2.1.3. Approximációs tér Pawlak-féle approximációs párral	17
2.1.4. Szemantika	18
2.1.5. Centrális logikai fogalmak	21
2.1.6. Formulahalmazok kielégíthetősége	22
2.1.7. Modus ponens	24
2.1.8. Modus tollens	26
2.2. Parciális elsőrendű logika	27
2.2.1. Kvantoros törvények és cáfolatok	28
2.2.2. A nyelv és a kielégíthetőség összefüggései	32
2.3. Eszközalapú parciális szemantika a gyakorlatban	34
2.3.1. Az interpretációt adó adattábla	35
2.3.2. A közelítő szemantika SQL implementációja	36
2.4. Szillogizmusok érvényessége	38
2.4.1. Arisztotelész kategorikus szillogizmusai	40
2.4.2. A szillogizmusok formalizálása	41
2.4.3. A szillogizmusok cáfolatainak keresése	45
2.4.4. Eredmények	48
2.5. A közelítés relevanciája	50
2.6. Összegzés	56

3. Approximáció állapottereken	57
3.1. A bizonytalanság kezelése klasszikus megoldáskeresőben	57
3.1.1. Visszalépéses kereső	57
3.1.2. Keresőfával keresők	61
3.2. Az állapothalmaz közelítése	63
3.2.1. Az <i>n</i> -vezér probléma állapotter-reprezentációi	64
3.2.2. A zebra probléma állapotter-reprezentációi	66
3.3. SAT probléma közelítő megoldása	69
3.3.1. Véges diszkrét állapotterek mint SAT problémák	69
3.3.2. SAT feladat megoldása approximációs térrel	70
3.3.3. Halmazmetszeteken alapuló approximációs tér	73
3.3.4. A megoldás közelítése	76
3.3.5. Közelítő megoldáskereső algoritmus	80
3.3.6. A közelítő megoldáskereső jellemzése	81
3.3.7. A zebra probléma közelítő megoldása	82
3.3.8. Az <i>n</i> -vezér probléma közelítő megoldása	85
4. Összefoglaló	87
5 Summary	89
5.1 Logical Systems Based on Partial Approximation Space	90
5.1.1 Aristotle's Syllogisms	92
5.2 Partial Approximation Space in Artificial Intelligence	94
5.2.1 Backtracking and Searching with Tree	95
5.2.2 Approximated Solutions Using Finite and Discrete State Spaces	96
5.3 Conclusion	98
Köszönetnyilvánítás	99
A. Halmazelméleti jelölések	100
B. A keresőfával kereső működésének illusztrációja	102
Irodalomjegyzék	104
Publikációs lista	107

Előszó

Az életlenhalmaz-elmélet – amelyet elsősorban Pawlak [28, 29] nevéhez fűzünk – számos esetben bizonyította létjogosultságát. Aktualitása azonnal érthetővé válik, amint belegondolunk, mennyi információ áll rendelkezésünkre, és mennyi keletkezik minden pillanatban. Az információk ábrázolása ugyanakkor sokszor természeténél fogva pontatlan, más esetben a rendelkezésünkre álló tudás hiányos, amelyből bizonytalanság fakad. Ezt a bizonytalanságot tette Pawlak egy megkülönböztethetlenségen alapuló ekvivalenciareláció által kezelhetővé.

Az életlen halmazok elméletében a bizonytalanságot a rendelkezésre álló tudás generálja az objektumok megkülönböztethetlensége alapján. Ez a megkülönböztethetlenség rányomja bélyegét arra, hogy mit tudunk mondani az egyes objektumokról, és azt, hogy ez mennyire bizonyos.

A halmazelméleti megközelítésben maga az *eleme* reláció az, ami bizonytalanná válik. Az életlen halmazok elméletében olyan objektumok mellett, amelyek bizonyosan elemei vagy bizonyosan nem elemei egy halmaznak, megjelenik a köztes, a halmazba lehetségesen tartozó elemek kategóriája is. Megjelenik az approximációs pár, amellyel egy halmazt alsó és felső közelítése (bizonyos vagy lehetséges elemei) segítségével határozhatunk meg.

Az elmúlt 30 évben az ekvivalenciareláción alapuló Pawlak-féle rendszer több általánosítása is napvilágot látott [6, 34], és kialakultak a halmazközelítés olyan általános keretrendszerei [4, 5], amelyek segítségével a hiányzó információ úgy kezelhető, hogy parciálissá teszi a teret, amelyben a közelítést végezzük.

A klasszikus halmazelméleten alapuló logika mintájára a parciális halmazközelítéssel alapuló logikai rendszerek jöttek létre [26, 23], amelyek segítségével újabb terület nyílt meg az életlen halmazok elméletének alkalmazása előtt. Ehhez ugyanakkor alaposan meg kell vizsgálni, hogy miként befolyásolja a halmazapproximáció segítségével megjelenő bizonytalanság a formuláink igazságértékének meghatározását, és az új alapokra helyezett szemantikai rendszer mely tulajdonságait tartja meg, vagy épp mely területeken tér el attól, amit a klasszikus logikában megszokhattunk. Más szavakkal: milyen előnyöket hozhat az approximációs tér segítségével felépített szemantika, és hol kell megfizetni ennek az árát.

Az elsőrendű logika fontos szerepet játszik abban, ahogy a problémáinkat modellezzük. Kérdés, hogy milyennek kell lennie egy, az általános approximációs tereken felépített logikai rendszernek ahhoz, hogy továbbra is alkalmas legyen erre a feladatra. A mesterséges intelligencia ugyanakkor nem csak a modellek megalkotásakor veheti hasznát a halmazközelítések elméletének, illetve nem csak logikai formulákon keresztül

húzhat hasznát a tudományterület eredményeiből, hiszen a tér maga, ahol a megoldásokat keressük, lehet egy halmazközelítéssel kialakított tér.

A dolgozat első bevezető fejezetében a vizsgált kérdéskör tekintetében releváns definíciók kerülnek ismertetésre. Először bemutatjuk a kiinduló alapot képező keretrendszert, amelynek segítségével a halmazközelítés definiálható. Bár mindvégig egy általánosított approximációs tér fogalmára építkezünk, a központban a parciálitás áll. A későbbi fejezetekben bemutatott rendszerek és alkalmazások jellemzően parciális approximációs teret feltételeznek a háttérben. A fejezet második felében a vizsgálatra kiválasztott parciális elsőrendű logika kerül bemutatásra. Ez – annak érdekében, hogy kihasználhassuk mindazt, amit az életlen halmazok nyújtanak – nem csupán egy új szemantikát, hanem egy gazdagabb elsőrendű nyelvet is jelent. A fejezet célja a későbbiekben felhasznált fogalmak tisztázása, a fejezetben a szerzőhöz köthető új tudományos eredmény nem jelenik meg.

A második fejezetet a parciális logikai rendszer vizsgálatának szenteljük. Előbb néhány szokatlan – a klasszikus logikában nem jelentkező – tulajdonságra hívjuk fel a figyelmet (melyek részben már publikálásra kerültek [11, 15]-ben), majd bemutatásra kerül a logikai rendszer egy implementációja. Mindezek célja, hogy előrevetítsék, milyen problémákkal kell szembesülnie annak, aki a parciális logikai rendszerrel dolgozik. Majd felhasználva Arisztotelész szillogizmusait, szeretnénk teljesebb képet adni a felmerülő sajátosságokról, és ismertetünk néhány eszközt ezek kezelésére (amelyeket először [12]-ben mutattunk be).

A harmadik fejezetben a mesterséges intelligencia területén kínálgató alkalmazási lehetőségeket járjuk körbe (melyek alapötletét [16]-ban mutattuk meg röviden). Előbb abból a nézőpontból indulunk ki, hogy miként élhetünk a probléma modellezésére használt formulákban a parciális approximációs tér adta lehetőségekkel, majd az állapothalmaz és a halmazközelítés képezi vizsgálat tárgyát. Végezetül a megoldáskezeső algoritmust (mint SAT problémamegoldót) alakítjuk át úgy, hogy profitálhasson az életlen halmazok elméletének eredményeiből.

1. fejezet

Bevezetés

Ebben a fejezetben részletesen áttekintjük a vizsgálni kívánt elmélet – az általánosított halmazközelítés, valamint a hozzá kapcsolódó logikai rendszerek – alapjait a dolgozat megírásakor viszonylag frissnek számító kutatások alapján, továbbá röviden ismertetésre kerül az állapottér-reprezentáció fogalma. A halmazelméletből átvett alapvető jelöléseket az A függelék tartalmazza. A fejezetben összegyűjtött definíciók és az általuk meghatározott halmazapproximációs, illetve logikai rendszerek nem számítanak a szerző új eredményének, hanem a vizsgálatunk tárgyának pontos meghatározása végett kerültek a vonatkozó irodalmakból idézésre, összegyűjtésre.

1.1. Általánosított approximációs keretrendszer

Az approximációs pár leginkább általános megfogalmazását Düntsch és Gediga adta meg [6]-ban. Eszerint az $\langle l, u \rangle$ függvénypárt, ahol $l, u : 2^U \rightarrow 2^U$ alakú leképezések, *approximációs pár*nak nevezzük (az U nemüres halmazon), ha tetszőleges $S \subseteq U$ esetén fennáll, hogy

$$l(S) \subseteq S \text{ és } S \subseteq u(S), \quad (1.1)$$

míg *gyenge approximációs pár*ról akkor beszélünk, ha teljesül, hogy

$$l(S) \subseteq u(S). \quad (1.2)$$

A szerzők úgy fogalmazzák, hogy ez utóbbi tűnik a leggyengébb feltételnek ahhoz, hogy észszerű közelítéseit tudjuk megadni az U halmaz részhalmazainak. Ezen feltételekkel később mint erős és gyenge approximációs tulajdonság találkozhatunk.

Halmazközelítések általános halmazelméleti keretrendszere

Ebben a dolgozatban a [4]-ben definiált általános halmazelméleti keretrendszert vetjük alapul a halmazközelítés mibenlétének meghatározására.

Legyen adott U nemüres halmaz mint univerzum. Legyen továbbá $\mathfrak{D} \subseteq 2^U$ úgy, hogy $\emptyset \in \mathfrak{D}$, és létezik legalább egy nemüres $S \subseteq U$ halmaz, amelyre igaz, hogy $S \in \mathfrak{D}$. Ekkor azt mondjuk, hogy $\mathfrak{D}$ elemei a *definálható halmazok*, míg $2^U \setminus \mathfrak{D}$ elemei a *nem*

definiálható halmazok. A cél az, hogy tetszőleges $S \subseteq U$ halmazt valamely $\mathfrak{D}$ -beli definiálható halmaz segítségével közelítsük (alulról és felülről). (A definiálhatóság kérdése önmagában is érdekes [2].)

1.1. Definíció. Az $\langle \mathfrak{l}, \mathfrak{u} \rangle$ függvénpárt gyenge approximációs párnak nevezzük, ha az $\mathfrak{l}$ és $\mathfrak{u}$ függvények olyan $\mathfrak{l}, \mathfrak{u} : 2^U \rightarrow \mathfrak{D}$ leképezések, amelyek az alábbi négy kritériumnak megfelelnek: [4]

1. Az $\mathfrak{l}$ és $\mathfrak{u}$ legyenek monoton függvények, azaz tetszőleges $S, S' \subseteq U$ esetén

$$S \subseteq S' \quad \Rightarrow \quad \mathfrak{l}(S) \subseteq \mathfrak{l}(S'), \quad \mathfrak{u}(S) \subseteq \mathfrak{u}(S').$$

2. Megköveteljük $\mathfrak{u}$ normalitását, miszerint $\mathfrak{u}(\emptyset) = \emptyset$.
3. Megköveteljük $\mathfrak{l}$ granularitását, vagyis minden $S \in \mathfrak{D}$ esetén $\mathfrak{l}(S) = S$.
4. Az $\langle \mathfrak{l}, \mathfrak{u} \rangle$ pár teljesítse az – (1.2)-ben megadott – gyenge approximációs tulajdonságot.

Ekkor az mondjuk, hogy tetszőleges $S \subseteq U$

1. alsó közelítése az $\mathfrak{l}(S) \in \mathfrak{D}$ halmaz,
2. felső közelítése pedig az $\mathfrak{u}(S) \in \mathfrak{D}$ halmaz.

Általános approximációs tér

1.2. Definíció. Az $\langle U, \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{l}, \mathfrak{u} \rangle$ rendezett ötöst általános approximációs térnek nevezzük, ha

1. U egy nemüres halmaz (univerzum);
2. $\mathfrak{B}$ alaphalmazokat tartalmazó halmaz, ahol $\mathfrak{B} \subseteq 2^U \setminus \emptyset$ és $\mathfrak{B} \neq \emptyset$;
3. $\mathfrak{D}_{\mathfrak{B}}$ tartalmazza a definiálható halmazokat (amelyeket az alaphalmazok kiterjesztésével kapunk) úgy, hogy $\mathfrak{B} \subseteq \mathfrak{D}_{\mathfrak{B}}$ és $\emptyset \in \mathfrak{D}_{\mathfrak{B}}$;
4. $\mathfrak{l}, \mathfrak{u} : 2^U \rightarrow \mathfrak{D}_{\mathfrak{B}}$ függvények egy gyenge approximációs párt alkotnak.

Az approximációs tér jellemzése

Legyen $\mathcal{GAS} = \langle U, \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{l}, \mathfrak{u} \rangle$ általános parciális approximációs tér.

- Az alaphalmazok kiválasztása alapján azt mondhatjuk, hogy $\mathcal{GAS}$:
 - teljes (total), ha $\bigcup \mathfrak{B} = U$;
 - parciális (partial), ha $\bigcup \mathfrak{B} \neq U$;
 - egyrétegű (single-layered), ha minden $B \in \mathfrak{B}$ esetén

$$B \cap \bigcup (\mathfrak{B} \setminus \{B\}) \neq B;$$

– *egyrétegű* (one-layered), ha minden $B \in \mathfrak{B}$ esetén

$$B \cap \bigcup(\mathfrak{B} \setminus \{B\}) = \emptyset;$$

– *Pawlak-féle alaprendszeren nyugszik*, ha $\mathfrak{B}$ az U halmaz egy partíciója.

- A definiálható halmazok kialakítása szerint a $\mathcal{GAS}$ approximációs tér
 - *unió típusú* (union-type), ha minden $\mathcal{B} \subseteq \mathfrak{B}$ esetén $\bigcup \mathcal{B} \in \mathfrak{D}_{\mathfrak{B}}$;
 - *metszet típusú* (intersection-type), ha minden $\mathcal{B} \subseteq \mathfrak{B}$ esetén $\bigcap \mathcal{B} \in \mathfrak{D}_{\mathfrak{B}}$;
 - *komplementer típusú*, ha minden $\mathcal{D} \in \mathfrak{D}_{\mathfrak{B}}$ esetén $U \setminus \mathcal{D} \in \mathfrak{D}_{\mathfrak{B}}$;
 - *Boole-típusú*, ha $\mathfrak{D}_{\mathfrak{B}}$ a $\mathfrak{B}$ alaphalmazok által generált Boole-algebra.
- Az $\langle l, u \rangle$ approximációs pár megválasztása szerint a $\mathcal{GAS}$ approximációs tér
 - *alulról félig erős*, ha minden $S \subseteq U$ esetén $l(S) \subseteq S$;
 - *felülről félig erős*, ha minden $S \subseteq U$ esetén $S \subseteq u(S)$;
 - *erős*, amennyiben alulról és felülről is félig erős (ami így megfelel Düntsch és Gediga (1.1) szerinti approximációs pár fogalmának).

1.2. Approximációs tér az elsőrendű logikában

A dolgozatban a klasszikus elsőrendű logikai nyelvire mint $L^{(c)}$ fogunk hivatkozni. A jelölések és a magyar nyelvű terminológia tekintetében [24] fogalomrendszerére támaszkodunk.

1.3. Definíció. Az $L^{(c)} = \langle LC^{(c)}, Var, Con, Term, Form^{(c)} \rangle$ rendezett ötös egy klasszikus elsőrendű logikai nyelv, amennyiben

1. $LC^{(c)} = \{\neg, \supset, \wedge, \vee, \forall, \exists, (,), '\}$ a logikai konstansok halmaza (balról jobbra rendre negáció-, implikáció-, konjunkció- és diszjunkciójel, univerzális és egzisztenciális kvantorok, zárójelek és vessző).
2. $Var = \{x_i : i = 0, 1, 2, \dots\}$ individuumváltozók megszámlálhatóan végtelen halmaza.
3. $Con = \bigcup_{n=0}^{\infty} (\mathcal{F}(n) \cup \mathcal{P}(n))$ nem logikai konstansok véges, nemüres halmaza, ahol
 - (a) $\mathcal{F}(0)$ a névparaméterek (névkonstansok) halmaza,
 - (b) $\mathcal{F}(n)$ (ahol $n \geq 1$) az n argumentumú függvényjelek (műveleti jelek) halmaza,
 - (c) $\mathcal{P}(0)$ az állításparaméterek (állításkonstansok, ítéletváltozók) halmaza,
 - (d) $\mathcal{P}(n)$ (ahol $n \geq 1$) az n argumentumú predikátumparaméterek (predikátumszimbólumok) halmaza.
4. Az $LC^{(c)}, Var, \mathcal{F}(n), \mathcal{P}(n)$ ($n = 0, 1, 2, \dots$) páronként diszjunktak.

A nyelv szavait termeknek és formuláknak nevezzük, amelyeket rendre a $Term$ és $Form^{(c)}$ induktívan definiált halmazok tartalmaznak.

5. A $Term$ halmaz induktív definíciója:

- (a) $Var \cup \mathcal{F}(0) \subseteq Term$;
- (b) $f(t_1, t_2, \dots, t_n) \in Term$, ha $f \in \mathcal{F}(n)$ és $t_1, t_2, \dots, t_n \in Term$ ($n \geq 1$).

6. A $Form^{(c)}$ halmaz induktív definíciója:

- (a) $\mathcal{P}(0) \subseteq Form^{(c)}$, vagyis az állításkonstansok (atomi) formulák;
- (b) $P(t_1, t_2, \dots, t_n) \in Form^{(c)}$ egy atomi formula, ha $P \in Pred(n)$, ahol $n \geq 1$ és $t_1, t_2, \dots, t_n \in Term$;
- (c) $\neg A$, $(A \supset B)$, $(A \wedge B)$, $(A \vee B) \in Form^{(c)}$, ha $A, B \in Form^{(c)}$;
- (d) $\forall x A$, $\exists x A \in Form^{(c)}$, ha $A \in Form^{(c)}$ és $x \in Var$.

Az $LC^{(c)}$ és $Form^{(c)}$ halmazok felső indexében a (c) jelölés a klasszikus elsőrendű logikára utal. A későbbiekben bevezetett logikai nyelvek definíciója ezekben tér majd el a fent megadottaktól.

1.4. Definíció. Az $L^{(c)} = \langle LC^{(c)}, Var, Con, Term, Form^{(c)} \rangle$ klasszikus elsőrendű logikai nyelv egy interpretációja alatt egy olyan $\langle U, \varrho \rangle$ rendezett párt értünk, amely esetén:

1. U egy nemüres halmaz,

2. ϱ egy függvény úgy, hogy

- (a) $\text{dom}(\varrho) = Con$,
- (b) ha $a \in \mathcal{F}(0)$, akkor $\varrho(a) \in U$,
- (c) ha $p \in \mathcal{P}(0)$, akkor $\varrho(p) \in \{0, 1\}$,
- (d) ha $f \in \mathcal{F}(n)$ ($n = 1, 2, \dots$), akkor $\varrho(f) \in U^{U^n}$,
- (e) ha $P \in \mathcal{P}(n)$ ($n = 1, 2, \dots$), akkor $\varrho(P) \in \{0, 1\}^{U^n}$.

Ahhoz, hogy tisztázzuk a halmazközelítés szerepét, először definiáljuk a logikai függvények pozitivitási és negativitási tartományát.

1.5. Definíció. Ha s egy tetszőleges függvény, akkor az s függvény pozitivitási tartománya alatt az $[s]^+$ -szal jelölt és a következőképpen definiált halmazt értjük:

$$[s]^+ \stackrel{\text{def}}{=} \{ u \in \text{dom}(s) : s(u) = 1 \}.$$

1.6. Definíció. Ha s egy tetszőleges függvény, akkor az s függvény negativitási tartománya alatt az $[s]^-$ -szal jelölt és a következőképpen definiált halmazt értjük:

$$[s]^- \stackrel{\text{def}}{=} \{ u \in \text{dom}(s) : s(u) = 0 \}.$$

Ha s egy klasszikus értelemben vett logikai függvény, akkor a $\{0, 1\}$ halmazba – mint az igazságértékek halmazába – képez, ahol szokás szerint 0 a *hamis*, míg 1 az *igaz* igazságértéket reprezentálja. Ekkor a pozitivitási tartomány azon univerzumelemek halmazát jelöli, ahol s igaz, míg a negativitási tartomány azokat, ahol s hamis. A pozitivitási és negativitási tartomány tetszőleges függvény esetében definiált, így abban az esetben is, ha a függvény értékkészlete nem tartalmazza a 0 vagy 1 értékeket.

1.7. Definíció. Egy adott U univerzum felett az $S_1, S_2 \subseteq U$ halmazokból alkotott $\langle S_1, S_2 \rangle$ párt orthopairnek nevezzük, ha S_1 és S_2 diszjunktak ($S_1 \cap S_2 = \emptyset$).

Amint azt később látni fogjuk, a parciális logikában találkozhatunk olyan logikai függvénnyel, amelynek mind a pozitivitási, mind a negativitási tartománya üres, ugyanakkor $\langle [s]^+, [s]^- \rangle$ orthopairt alkot¹.

A klasszikus elsőrendű nyelv minden P predikátumparamétere a nyelvhez tartozó adott $\langle U, \varrho \rangle$ interpretációban jellemezhető a hozzá tartozó $[\varrho(P)]^+$ pozitivitási tartománnyal (szokás is a predikátumparaméterek interpretálását – logikai függvény helyett – ezzel megadni):

$$[\varrho(P)]^+ \subseteq U^n, \text{ ha } P \in \mathcal{P}(n) \text{ és } n \geq 1.$$

A predikátumparaméterekhez rendelt halmazok már képezhetik alapját egy approximációs térnek, vagy lehetnek éppen az approximációs tér segítségével közelíteni kívánt halmazok.

Arra, hogy hogyan lehet észszerű módon közelítést definiálni a pozitivitási tartományokhoz, elsőként [23] alapján láthattunk példát. Példaként tekintsünk egy $P \in \mathcal{P}(n)$ predikátumparamétert egy $L^{(c)}$ nyelvben. Legyen adott a nyelv egy $\langle U, \varrho \rangle$ interpretációja, továbbá egy $\langle \mathbf{l}, \mathbf{u} \rangle$ approximációs pár, amely esetén teljesül, hogy tetszőleges $S \subseteq U^n$ esetén $\mathbf{l}(S) \subseteq U^n$ és $\mathbf{u}(S) \subseteq U^n$. Jelölje $S = [\varrho(P)]^+$ a P predikátumparaméter pozitivitási tartományát, amelynek segítségével az $s^\ell, s^u, s^m : U^n \rightarrow \{0, 1, 2\}$ függvényeket definiáljuk a következőképpen:

$$s^\ell(u_1, \dots, u_n) = \begin{cases} 1 & \text{ha } \langle u_1, \dots, u_n \rangle \in \mathbf{l}(S), \\ 0 & \text{ha } \langle u_1, \dots, u_n \rangle \in \mathbf{l}(U^n \setminus \mathbf{u}(S)), \\ 2 & \text{egyébként;} \end{cases} \quad (1.3)$$

$$s^u(u_1, \dots, u_n) = \begin{cases} 1 & \text{ha } \langle u_1, \dots, u_n \rangle \in \mathbf{u}(S), \\ 0 & \text{ha } \langle u_1, \dots, u_n \rangle \in \mathbf{l}(U^n \setminus \mathbf{u}(S)), \\ 2 & \text{egyébként;} \end{cases} \quad (1.4)$$

$$s^m(u_1, \dots, u_n) = \begin{cases} 1 & \text{ha } \langle u_1, \dots, u_n \rangle \in \mathbf{l}(S), \\ 2 & \text{ha } \langle u_1, \dots, u_n \rangle \in \mathbf{u}(S) \setminus \mathbf{l}(S), \\ 0 & \text{egyébként.} \end{cases} \quad (1.5)$$

Az s^ℓ, s^u és s^m függvényeket parciális logikai függvényeknek fogjuk tekinteni, ahol a 2-es reprezentálja az igazságérték hiányát. A három parciális logikai függvény rendre

¹Az orthopair fogalmát most azért említjük, mert számos ponton kapcsolódik az életlen halmazok elméletéhez [1].

alsó, felső és vegyes közelítése P ϱ szerinti szemantikai értékének. Ugyancsak [23]-ban található meg az $\langle U, \varrho^\ell \rangle$, $\langle U, \varrho^u \rangle$ és $\langle U, \varrho^m \rangle$ parciális interpretációk bevezetése, amelyek rendre (1.3), (1.4) és (1.5) szerint definiálják a predikátumparaméterek szemantikai értékét.

A ϱ leképezés segítségével létrejött pozitivitási tartományok az approximációs tér generálásában is segítségünkre lehetnek. Az approximációs tér alaphalmazai kitüntetett predikátumparaméterek segítségével határozhatók meg – amelyeket eszközöknek nevezünk és a $\mathfrak{T}$ halmazban sorolunk fel [23] –, ahol

$$\mathfrak{T} \subseteq \bigcup_{n=1}^{\infty} \mathcal{P}(n).$$

Ahhoz, hogy egy-egy eszköz pozitivitási tartománya egy $\langle U, \varrho \rangle$ interpretációban alkalmas legyen arra, hogy approximációs tér alaphalmazaként használjuk, mindössze annyit kell elvárjunk tőle, hogy ne legyen üres.

1.3. Elsőrendű logika közelítő mondatfunktorokkal

Az előző fejezetben bepillantottunk abba, hogy hogyan lehetséges egy klasszikus interpretáció alapján észszerű közelítő interpretációkat generálni. A következőkben ([21, 26] alapján) egy olyan kiterjesztését tekintjük át az elsőrendű nyelvnek, amelyben a különféle közelítési módszerek együtt, a nyelv szintjén is megjelenítve használhatók és kombinálhatók. Tekintettel arra, hogy a későbbi vizsgálatok alapját is ez a rendszer, illetve ennek különböző egyszerűsített változatai képezik, így itt [21] alapján mind maga a nyelv, mind annak szemantikája részletesen bemutatásra kerül.

1.8. Definíció. Az $L = \langle LC, Var, Con, Term, \mathfrak{T}, Pred, Form \rangle$ rendezett hetes egy eszköz alapú parciális elsőrendű logikai nyelv (*tool-based partial first-order logic*), amennyiben:

1. $LC = \{\neg, \supset, +, \forall, \exists, \downarrow, \uparrow, (,), '\}$ a logikai konstansok halmaza.
2. A Var , Con és $Term$ halmazok definíciója egybeesik a klasszikus (1.3. definícióban megadott) esettel.
3. Az LC , Var , $\mathcal{F}(n)$, $\mathcal{P}(n)$ ($n = 0, 1, 2, \dots$) páronként diszjunktak.
4. $\mathfrak{T} \subseteq \bigcup_{n=1}^{\infty} \mathcal{P}(n)$ az eszközök nemüres és véges halmaza.
5. $Pred = \bigcup_{n=1}^{\infty} Pred(n)$ a predikátumok halmaza, amely az n argumentumú predikátumok $Pred(n)$ halmazainak segítségével definiált.
A $Pred(n)$ induktív definíciója:

- (a) $\mathcal{P}(n) \subseteq Pred(n)$, ahol ($n = 1, 2, \dots$);
- (b) $P^\downarrow, P^\uparrow \in Pred(n)$, ha $P \in Pred(n)$.

6. A nyelv formuláinak $Form$ halmaza induktívan definiált:

- (a) $\mathcal{P}(0) \subseteq Form$;
- (b) $P(t_1, t_2, \dots, t_n) \in Form$,
ha $P \in Pred(n)$, ahol $n \geq 1$ és $t_1, t_2, \dots, t_n \in Term$;
- (c) $\neg A, +A, (A \supset B) \in Form$, ha $A, B \in Form$;
- (d) $\forall x A, \exists x A \in Form$, ha $A \in Form$ és $x \in Var$.

A logikai konstansok halmazát a definíció igyekszik minél szűkebben meghatározni. A hiányzó szokásos logikai jeleket később mint rövidítéseket fogjuk bevezetni. A új $+$ operátorra tipikusan a többértékű logikai rendszerekben van szükség. Segítségével minden igazságértéket a $\{0, 1\}$ halmazra (a *hamis* és *igaz* igazságértékek halmazára) képzünk, és – amint azt később a szemantika megadásakor látni fogjuk – az „*igaz*, *hoggy*” értelemben használjuk. Ezt az operátort értékészlete miatt szoktuk digitalizáló operátornak is hívni. Természetesen definiálható volna a „*hamis*, *hoggy*” jelentéssel bíró – operátor is, amit szintén a minél szűkebb nyelvre való törekvés miatt hagyunk el. A $\downarrow$ és $\uparrow$ jeleket közelítő mondatfunktorknak nevezzük (és amint ezt később látni fogjuk, alsó és felső közelítésre utalnak). Ezeken felül az approximációs tér későbbi kialakításához a nyelv szintjén megjelenik a kitüntetett – eszköznek nevezett – predikátumparaméterek $\mathfrak{T}$ halmaza.

1.9. Definíció. $A \downarrow$ és $\uparrow$ közelítő mondatfunktork induktívan vezethetők be az alábbiak szerint:

1. Ha $p \in \mathcal{P}(0)$, akkor $\downarrow p \stackrel{def}{=} p, \uparrow p \stackrel{def}{=} p$.
2. Ha $P \in Pred(n)$, és $t_1, t_2, \dots, t_n \in Term$, úgy
 - (a) $\downarrow P(t_1, t_2, \dots, t_n) \stackrel{def}{=} P^\downarrow(t_1, t_2, \dots, t_n)$,
 - (b) $\uparrow P(t_1, t_2, \dots, t_n) \stackrel{def}{=} P^\uparrow(t_1, t_2, \dots, t_n)$.
3. Ha $A \in Form$, akkor
 - (a) $\uparrow \neg A \stackrel{def}{=} \neg \uparrow A$, és $\downarrow \neg A \stackrel{def}{=} \neg \downarrow A$,
 - (b) $\uparrow +A \stackrel{def}{=} + \uparrow A$, és $\downarrow +A \stackrel{def}{=} + \downarrow A$.
4. Ha $A, B \in Form$, akkor $\uparrow (A \supset B) \stackrel{def}{=} (\uparrow A \supset \uparrow B)$ és
 $\downarrow (A \supset B) \stackrel{def}{=} (\downarrow A \supset \downarrow B)$.
5. Ha $A \in Form$ és $x \in Var$, akkor $\uparrow \forall x A \stackrel{def}{=} \forall x \uparrow A$ és $\uparrow \exists x A \stackrel{def}{=} \exists x \uparrow A$.

Így például egy olyan egyszerű nyelvben, ahol $Con = \mathcal{P}(1) = \{Q\}$, azaz amelyben egyetlen Q predikátumparaméter van, és az is egyargumentumú, felírható a következő formula:

$$\downarrow (\neg Q(x) \supset \uparrow Q(y)) = (\downarrow \neg Q(x) \supset \downarrow \uparrow Q(y)) = (\neg Q^\downarrow(x) \supset Q^{\uparrow\downarrow}(y)).$$

1.10. Definíció. Legyen $L = \langle LC, Var, Con, Term, \mathfrak{T}, Pred, Form \rangle$ egy eszközalapú parciális elsőrendű logikai nyelv. Az $\langle U, \varrho \rangle$ rendezett pár egy interpretációja az L nyelvnek, ha

1. $\langle U, \varrho \rangle$ interpretációja az $L^{(c)} = \langle LC^{(c)}, Var, Con, Term, Form^{(c)} \rangle$ nyelvnek;
2. minden $T \in \mathfrak{T} \cap \mathcal{P}(n)$ esetén van olyan $\langle u_1, u_2, \dots, u_n \rangle \in U^n$, hogy $\varrho(T)(\langle u_1, u_2, \dots, u_n \rangle) = 1$.

Az utolsó kritérium az egyetlen, amely egy klasszikus elsőrendű logikai nyelv interpretációja esetén nem merül fel. Az eszközeinket szeretnénk később felhasználni az approximációs tér generálására. Az alaphalmazaink az eszközök pozitivitási tartományai lesznek, így ezek nem lehetnek üresek.

1.11. Definíció. A v függvény egy változókiértékelés az $\langle U, \varrho \rangle$ interpretációban, ha

1. $\text{dom}(v) = Var$, és
2. $\text{rng}(v) \subseteq U$.

1.12. Definíció. Legyen v egy változókiértékelés az $\langle U, \varrho \rangle$ interpretációban, $x \in Var$ egy változó és $u \in U$. A $v[x:u]$ függvény a v változókiértékelés x szerinti variánsa, ha

1. $v[x:u]$ egy változókiértékelés az $\langle U, \varrho \rangle$ interpretációban, és
2. $v[x:u](y) = \begin{cases} u & \text{ha } x = y; \\ v(y) & \text{egyébként.} \end{cases}$

Az eszközök (a $\mathfrak{T}$ halmazba tartozó predikátumparaméterek) feladata, hogy meghatározzák a logikailag releváns approximációs teret úgy, hogy az eszközök értékei segítségével alkotjuk meg azokat a halmazokat, amelyekkel halmazközelítést végzünk.

1.13. Definíció. A $\mathcal{GAS}(\mathfrak{T}) = \langle PR(U), \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{l}, \mathfrak{u} \rangle$ rendezett ötös az $\langle U, \varrho \rangle$ interpretációhoz tartozó logikailag releváns általánosított (parciális) approximációs tér, amelyet a $\mathfrak{T}$ eszközök generáltak, ha

1. $PR(U) = \bigcup_{n=1}^{\infty} U^n$
2. $\mathfrak{B} = \left\{ [\varrho(T)]^+ : T \in \mathfrak{T} \right\}$, azaz az alaphalmazok az eszközként választott predikátumparaméterek interpretációhoz tartozó (garantáltan nem üres) pozitivitási tartományai,
3. $\mathfrak{D}_{\mathfrak{B}} \subseteq \bigcup_{n=1}^{\infty} 2^{U^n}$,
4. ha $S \subseteq U^n$, akkor $\mathfrak{l}(S), \mathfrak{u}(S) \in \mathfrak{D}_{\mathfrak{B}} \cap 2^{U^n}$.

A szemantikai definíciók

A következőkben is a [21]-ben definiált eszközalapú parciális logikai rendszer (Tool-based Partial First-order Logic: TbPFoL) felépítését követjük, amelynek eredményeképpen egy kétértékű, de parciális szemantika jelenik meg. Az igazságértékek továbbra is 0-val és 1-gyel lesznek ábrázolva, az igazságérték hiányát pedig 2-es jelöli.

1.14. Definíció. Egy $L = \langle LC, Var, Con, Term, \mathfrak{T}, Pred, Form \rangle$ eszközalapú parciális elsőrendű logikai nyelv termjeinek szemantikai értéke a nyelvhez tartozó $\langle U, \varrho \rangle$ interpretációban és az interpretációhoz tartozó v változókiértékelés mellett rekurzívan definiálható:

1. Ha $x \in Var$ változó, akkor $\llbracket x \rrbracket_v^{\langle U, \varrho \rangle} = v(x)$.
2. Ha $a \in \mathcal{F}(0)$ (azaz a egy névparaméter), akkor $\llbracket a \rrbracket_v^{\langle U, \varrho \rangle} = \varrho(a)$.
3. Ha $f(t_1, t_2, \dots, t_n) \in Term$ (ahol $n \geq 1$), továbbá $\varrho(f) = f'$, akkor
$$\llbracket f(t_1, t_2, \dots, t_n) \rrbracket_v^{\langle U, \varrho \rangle} = f' \left(\llbracket t_1 \rrbracket_v^{\langle U, \varrho \rangle}, \llbracket t_2 \rrbracket_v^{\langle U, \varrho \rangle}, \dots, \llbracket t_n \rrbracket_v^{\langle U, \varrho \rangle} \right).$$

A definícióból leolvasható, hogy a termék szemantikai értéke nem különbözik a klasszikus elsőrendű logika által meghatározottól, és így nem függ az approximációs tértől sem.

1.15. Definíció. Egy $L = \langle LC, Var, Con, Term, \mathfrak{T}, Pred, Form \rangle$ eszközalapú parciális elsőrendű logikai nyelv $P \in Pred(n)$ predikátumparaméterének szemantikai értéke adott $\langle U, \varrho \rangle$ interpretációban és az interpretációhoz tartozó v változókiértékelés, valamint adott $\mathcal{GAS}(\mathfrak{T})$ logikailag releváns approximációs tér mellett az alábbi – rekurzív definícióval megadott – $U^n \rightarrow \{0, 1, 2\}$ függvény. (A $P \in Pred(n)$ predikátumparaméterhez tartozó függvényt $\llbracket P \rrbracket_v^{\langle U, \varrho \rangle, \mathcal{GAS}(\mathfrak{T})}$ -vel, vagy ha félreértést nem okoz, egyszerűen $\llbracket P \rrbracket_v$ -vel fogjuk jelölni.)

1. Ha $P \in \mathcal{P}(n) \setminus \mathfrak{T}$ és $n \geq 1$, akkor $\llbracket P \rrbracket_v = \varrho(P)$.
2. Ha $T \in \mathfrak{T}$ egy n argumentumú predikátumparaméter, akkor hozzá egy $\llbracket T \rrbracket_v = s$ függvényt rendelünk, ahol $s : U^n \rightarrow \{0, 1, 2\}$, továbbá

$$s(u_1, u_2, \dots, u_n) = \begin{cases} 1 & \text{ha } \langle u_1, u_2, \dots, u_n \rangle \in [\varrho(T)]^+; \\ 0 & \text{ha } \langle u_1, u_2, \dots, u_n \rangle \in \mathfrak{I}(U^n \setminus [\varrho(T)]^+); \\ 2 & \text{egyébként.} \end{cases} \quad (1.6)$$

3. Ha $P^\perp \in Pred(n)$, ahol $n \geq 1$, akkor $\llbracket P^\perp \rrbracket_v = s$ egy olyan függvény, ahol $s : U^n \rightarrow \{0, 1, 2\}$, továbbá

$$s(u_1, u_2, \dots, u_n) = \begin{cases} 1 & \text{ha } \langle u_1, u_2, \dots, u_n \rangle \in \mathfrak{I}([\llbracket P \rrbracket_v]^+); \\ 0 & \text{ha } \langle u_1, u_2, \dots, u_n \rangle \in \mathfrak{I}(U^n \setminus \mathfrak{U}([\llbracket P \rrbracket_v]^+)); \\ 2 & \text{egyébként.} \end{cases} \quad (1.7)$$

4. Ha $P^\uparrow \in \text{Pred}(n)$, ahol $n \geq 1$, akkor $\llbracket P^\uparrow \rrbracket_v = s$ egy olyan függvény, ahol $s : U^n \rightarrow \{0, 1, 2\}$, továbbá

$$s(u_1, u_2, \dots, u_n) = \begin{cases} 1 & \text{ha } \langle u_1, u_2, \dots, u_n \rangle \in \mathbf{u} \left(\llbracket [P] \rrbracket_v^+ \right); \\ 0 & \text{ha } \langle u_1, u_2, \dots, u_n \rangle \in \mathbf{l} \left(U^n \setminus \mathbf{u} \left(\llbracket [P] \rrbracket_v^+ \right) \right); \\ 2 & \text{egyébként.} \end{cases} \quad (1.8)$$

A közelítő mondatfunktorok szemantikai jelentése mögött meghúzódó ötlet bemutatására [24]-ből idézzük ide a következő magyarázatot:

„Ha P egy olyan egyargumentumú predikátumparaméter, amely nem eszik (azaz $P \notin \mathfrak{T}$), és $u \in U$, akkor

- a $\downarrow$ funktor a következő eredményeket adja:
 1. $P^\downarrow$ igaz u -ra, ha u eleme a $[\varrho(P)]^+$ halmaz alsó approximációjának (azaz a háttértudás alapján P bizonyosan igaz u -ra);
 2. $P^\downarrow$ hamis u -ra, ha u eleme az $U \setminus [\varrho(P)]^+$ halmaz alsó approximációjának (azaz a háttértudás alapján P bizonyosan hamis u -ra);
 3. más esetekben $P^\downarrow$ definiálatlan u -ra (azaz a háttértudás nem elegendő arra, hogy bizonyosan teljesülő információra tegyünk szert P és u vonatkozásában);
- a $\uparrow$ funktor a következő eredményeket adja:
 1. $P^\uparrow$ igaz u -ra, ha u eleme a $[\varrho(P)]^+$ halmaz felső approximációjának (azaz a háttértudás alapján P lehet igaz – feltehetően igaz – u -ra);
 2. $P^\uparrow$ hamis u -ra, ha u eleme azon halmaz alsó approximációjának, amelynek elemeire nem igaz a $P^\uparrow$ (azaz a háttértudás alapján P bizonyosan hamis u -ra);
 3. más esetekben $P^\uparrow$ definiálatlan u -ra (azaz a háttértudás nem elegendő arra, hogy eldöntsük, hogy P lehet-e igaz, vagy bizonyosan hamis u -ra);
- a $\uparrow, \downarrow$ kontextuálisan bevezetett mondatfunktorok szándékolt jelentése azt határozza meg, hogy a hatókörben szereplő formulában előforduló predikátumok szemantikai értékének alsó, illetve felső közelítését vegyük figyelembe a kimenet szemantikai értékének meghatározásakor.”

Megjegyzés. Az (1.7)-es és (1.8)-as függvények lényegében megegyeznek az (1.3)-ban és (1.4)-ben megadott konstrukciókkal. Ugyanakkor $\text{Pred}(n)$ induktív definíciója miatt lehetőség nyílik a közelítő mondatfunktorok többszörös alkalmazására, például $P, P^\uparrow, P^{\uparrow\uparrow}$ vagy $P^{\downarrow\downarrow}$ alakban.

1.16. Definíció. Egy $L = \langle LC, \text{Var}, \text{Con}, \text{Term}, \mathfrak{T}, \text{Pred}, \text{Form} \rangle$ eszközalapú parciális elsőrendű logikai nyelv egy $A \in \text{Form}$ formulájának szemantikai értéke a nyelvhez

tartozó adott $\langle U, \varrho \rangle$ interpretációban és az interpretációhoz tartozó v változókiértékelés, valamint adott $\mathcal{GAS}(\mathfrak{T})$ logikailag releváns approximációs tér mellett az alábbi rekurzív definícióval megadott $\{0, 1, 2\}$ halmazba tartozó érték. Az $A \in \text{Form}$ formula értékét $\llbracket A \rrbracket_v^{\langle U, \varrho \rangle, \mathcal{GAS}(\mathfrak{T})}$ -vel, vagy ha félreértést nem okoz, egyszerűen $\llbracket A \rrbracket_v$ -vel jelöljük.

1. Ha $p \in \mathcal{P}(0)$ (azaz p egy állításparaméter), akkor

$$\llbracket p \rrbracket_v = \varrho(p).$$

2. Ha $P(t_1, \dots, t_n) \in \text{Form}$ egy atomi formula és $n \geq 1$ (azaz $P \in \text{Pred}$), akkor

$$\llbracket P(t_1, \dots, t_n) \rrbracket_v = \llbracket P \rrbracket_v (\llbracket t_1 \rrbracket_v, \dots, \llbracket t_n \rrbracket_v).$$

3. Ha $A \in \text{Form}$, $B \in \text{Form}$ és $x \in \text{Var}$, akkor

$$\begin{aligned} \llbracket +A \rrbracket_v &= \begin{cases} 1 & \text{ha } \llbracket A \rrbracket_v = 1, \\ 0 & \text{egyébként;} \end{cases} \\ \llbracket \neg A \rrbracket_v &= \begin{cases} 2 & \text{ha } \llbracket A \rrbracket_v = 2, \\ 0 & \text{ha } \llbracket A \rrbracket_v = 1, \\ 1 & \text{egyébként;} \end{cases} \\ \llbracket (A \supset B) \rrbracket_v &= \begin{cases} 0 & \text{ha } \llbracket A \rrbracket_v = 1, \text{ és } \llbracket B \rrbracket_v = 0, \\ 2 & \text{ha } \llbracket A \rrbracket_v = 2, \text{ vagy } \llbracket B \rrbracket_v = 2, \\ 1 & \text{egyébként;} \end{cases} \\ \llbracket \forall x A \rrbracket_v &= \begin{cases} 0 & \text{ha van olyan } u \in U, \text{ amely esetén } \llbracket A \rrbracket_{v[x:u]} = 0, \\ 2 & \text{ha minden } u \in U \text{ esetén } \llbracket A \rrbracket_{v[x:u]} = 2, \\ 1 & \text{egyébként;} \end{cases} \\ \llbracket \exists x A \rrbracket_v &= \begin{cases} 1 & \text{ha van olyan } u \in U, \text{ amely esetén } \llbracket A \rrbracket_{v[x:u]} = 1, \\ 2 & \text{ha minden } u \in U \text{ esetén } \llbracket A \rrbracket_{v[x:u]} = 2, \\ 0 & \text{egyébként.} \end{cases} \end{aligned}$$

A szokásos logikai összekötő jelek a negáció és implikáció segítségével definiálhatók. Ha $A \in \text{Form}$ és $B \in \text{Form}$, akkor

$$\begin{aligned} (A \wedge B) &\stackrel{\text{def}}{=} \neg(A \supset \neg B) \\ (A \vee B) &\stackrel{\text{def}}{=} (\neg A \supset B) \\ (A \equiv B) &\stackrel{\text{def}}{=} ((A \supset B) \wedge (B \supset A)) \end{aligned}$$

Az eszközalapú parciális elsőrendű logika az 1.16. definícióban megadott szemantika szerint értékréses elsőrendű logika². Ez abban érhető tetten, hogy egyrészt az

²Az értékréses elsőrendű logika alatt Ruzsa Imre és Máté András [32]-ben (235–236. oldal) közölt meghatározását értjük.

ítéletlogikai műveletek ($a +$ kivételével) örökítik az értékrést, másrészt a kvantifikáció eredménye csak akkor 2, ha a kötött változó minden lehetséges helyettesítési értéke mellett 2-es értéket vesz fel a közvetlen részformula. A 2-es érték az értékréses logikában nem tekintendő igazságértéknek, hanem épp az igazságérték hiányát jelöli, épp úgy, ahogy a $\llbracket P^+ \rrbracket_v$ és $\llbracket P^\uparrow \rrbracket_v$ parciális logikai függvények esetén is értékhiányt jelöltünk vele (még ha formálisan a parciális logikai függvényt $\{0, 1, 2\}$ halmazba való totális leképezésként is ábrázoljuk).

Az, hogy értékréses logikai rendszer került kiválasztásra, a parcialitás egyenes következménye, és magát az approximációs teret is az információhiány teszi parciálissá. A fenti definíciókból is látható, hogy az értékréses logika műveletei egyes többértékű logikai rendszerekben is fellelhetők – például az értékréses logika ítéletlogikai összekötő jeleinek szemantikája megegyezik Kleene háromértékű logikai rendszerének gyenge konnektívumaival ([19] 2438. oldal). Az előbb bemutatott rendszert követve most a klasszikus logika értékréses változatát vizsgáljuk (innen vesszük át mind az ítéletlogikai műveletek, mind a kvantorok szemantikáját). Természetesen többféle módon is lehet a parciális approximációs térre logikai rendszert építeni, ahogy többféle logikai rendszer mögé is elhelyezhető a parciális logikai rendszer. Így például [26]-ban találkozhatunk háromértékű rendszer parciális változatával (ahol a három igazságérték, a 0, $1/2$ és 1 egészülni ki az értékhiányt jelölő 2-essel), illetve tagadhatatlan a kapcsolat megléte a többértékű logikai rendszerek és az élethen halmazok elmélete közt [9, 10].

1.4. Állapottér a mesterséges intelligenciában

A parciális approximációs téren alapuló parciális logikai rendszerek mesterséges intelligencia területén történő felhasználási lehetőségeit elsőként az állapottéren definiált megoldáskereső algoritmusok körében vizsgáltuk. Ennek oka, hogy az állapottér-reprezentáció segítségével modellezett problémák leírásában az elsőrendű logika számos ponton szerepet kap.

1.17. Definíció. Egy p probléma állapottér-reprezentációja alatt az $\langle \mathcal{A}, a_0, \mathcal{C}, \mathcal{O} \rangle$ rendezett négyest értjük [7], ahol

1. $\mathcal{A}$ egy nemüres halmaz, a (valódi) állapotok halmaza;
2. $a_0 \in \mathcal{A}$ a probléma kezdőállapota;
3. $\mathcal{C} \subset \mathcal{A}$ a célállapotok halmaza;
4. $\mathcal{O}$ egy nemüres halmaz, amely az állapottér operátorait tartalmazza, ahol minden $o \in \mathcal{O}$ esetén o egy olyan leképezés, ahol $\text{dom}(o) \subseteq \mathcal{A}$ és $\text{rng}(o) \subseteq \mathcal{A}$.

Az állapottér-reprezentáció meghatároz egy címkézett, irányított éleket tartalmazó gráfot. Az állapottér és a hozzá tartozó gráf fogalma is többféleképp általánosítható, a [13]-ban közzétett változathoz most csak annyit veszünk át, hogy az állapottérgráf irányított éleit címkézzük egy-egy operátor nevével.

1.18. Definíció. Az $\langle \mathcal{A}, a_0, \mathcal{C}, \mathcal{O} \rangle$ állapottér-reprezentáció által meghatározott állapottérgráf egy olyan $G = \langle \mathcal{A}, \mathcal{E} \rangle$ pár, ahol

1. a gráf csúcsai az állapottér-reprezentáció $\mathcal{A}$ halmazába tartozó állapotai;
2. a gráf élei $\mathcal{E} \subseteq \mathcal{A} \times \mathcal{A} \times \mathcal{O}$ halmazba tartozó rendezett hármasok úgy, hogy $\langle a, a', o \rangle \in \mathcal{E}$ akkor és csak akkor teljesül, ha $a \in \text{dom}(o)$ és $o(a) = a'$.

A definíció sajátossága, hogy egy-egy állapotpár önmagában nem határoz meg egy irányított élt a gráfban, hiszen létezhet olyan $a \in \mathcal{A}$ és $a' \in \mathcal{A}$ állapotpár, valamint $o_1 \in \mathcal{O}$ és $o_2 \in \mathcal{O}$ operátorok, hogy $o_1(a) = a'$ és $o_2(a) = a'$. A különböző operátorok alkalmazása eltérő következményekkel jár; például ha költséget rendelünk az élekhez, akkor az o_1 és o_2 operátorokkal címkézett élek költsége nem feltétlenül lesz azonos, még ha az irányított élek ugyanazon állapotokat kötik is össze.

Az állapottér-reprezentáció segítségével felírt problémát megoldani annyit tesz, hogy kell találni egy utat az állapottérgráfban, amely a kezdőállapotból a célállapotok egyikébe vezet. Az már a problémától függ, hogy ilyenkor elegendő-e egy célállapot megkonstruálása, vagy a célállapothoz vezető út is érdekes. Attól, hogy az állapottér-reprezentáció elkészítése során a probléma célállapotainak halmazát definiáljuk, a célállapot megkonstruálása még komoly feladat lehet, ahogy azt sem egyszerű eldönteni, hogy egyáltalán létezik-e ilyen állapot.

A megoldáskereső algoritmusokban a keresés során egy állapottérgráf-részletet tárolnak. Az állapottérgráf csúcsait olyan rekordokkal jellemezzük, amelyek tartalmazzák a rekordhoz tartozó állapot leírását, az azt előállító operátort és a megelőző (szülő) rekord referenciáját. Ilyen megoldáskereső algoritmus a visszalépéses keresés [31], ahol az állapottérgráf feltárt részéből egyetlen utat tárolunk el, és ilyen a szélességi és a mélységi kereső is [3, 31], ahol már egy részfa kerül tárolásra a megoldáskereső során. Azt, hogy hogyan lehet különféle problémákhoz állapottér-reprezentációkat készíteni, majd ezekhez keresőalgoritmusokat hatékonyan és szemléletesen implementálni objektumorientált vagy akár multiparadigmás programozási nyelven, azt [18, 30]-ban tekinthetjük át.

A dolgozatban meg fogjuk vizsgálni, hogy az említett algoritmusok esetén miként kezelhető az a bizonytalanság, amit az állapotteret definiáló formulák kiértékelése okoz egy parciális logikai rendszerben. Majd véges, diszkrét állapotterek esetén foglalkozunk közelítő megoldások keresésével.

2. fejezet

Logikai rendszerek vizsgálata

2.1. Parciális kvantormentes logika

A parciális logikai rendszer vizsgálatát egy szűkített elsőrendű nyelvvel kezdjük, amely segít érzékeltetni, hogy milyen szokatlan tulajdonságok jelennek meg a közelítéssel végzett számítások esetében.

A következőkben az egyszerűség kedvéért kvantormentes és kizárólag pontosan egy argumentummal rendelkező predikátumokat tartalmazó esetre szűkítjük a vizsgálatot.

2.1.1. A közelítő funktorokat tartalmazó kvantormentes nyelv

2.1. Definíció. Az $L = \langle LC, Var, \mathcal{P}(1), \mathfrak{T}, Form, Snt \rangle$ rendezett hatos egy eszköz-alapú parciális kvantormentes logikai nyelv, amennyiben

1. $LC = \{\neg, \supset, +, \downarrow, \uparrow, (,)\}$ logikai konstansok halmaza;
2. $Var = \{x_i : i = 0, 1, 2, \dots\}$ megszámlálhatóan végtelen sok változó;
3. a nem logikai konstansok halmaza kizárólag az egy argumentummal rendelkező predikátumparaméterek $\mathcal{P}(1)$ halmaza;
4. $\mathfrak{T} \subseteq \mathcal{P}(1)$ az eszközök nemüres halmaza;
5. az $LC, Var, \mathcal{P}(1)$ halmazok páronként diszjunktak;
6. a formulák $Form$ halmaza induktívan definiált úgy, hogy
 - (a) $P(x) \in Form$ atomi formula, ha $P \in \mathcal{P}(1)$ és $x \in Var$, továbbá
 - (b) ha $A \in Form$ és $B \in Form$, akkor $\neg A, (A \supset B), +A \in Form$;
7. a mondatok Snt halmaza a legszűkebb olyan halmaz, hogy
 - (a) $Form \subseteq Snt$, és
 - (b) ha $F \in Form$, akkor $\uparrow F \in Snt$, és $\downarrow F \in Snt$.

Az egyszerűsített nyelv szavai az eredeti nyelvnek is szavai, így az ebben konstruált interpretációk és az azok létezéséből levezetett következtetések mindkét konstrukcióban érvényesek. Következésképpen elegendő az egyszerűbb nyelvben egy formulahalmaz kielégíthetőségét bizonyítani. Az új nyelvben is alkothatunk közelítő mondatfunktorok segítségével mondatokat, ugyanakkor a közelítő mondatfunktor ez esetben legfeljebb egyszer szerepelhet egy mondatban. A szemantikából hamarosan látni fogjuk, hogy ilyenkor a közelítő mondatfunktor szerepe csupán annyi, hogy jelezze, hogy az atomi formulák értékének meghatározásakor milyen közelítést használjunk.

2.1.2. Interpretáció

2.2. Definíció. Az $L = \langle LC, Var, \mathcal{P}(1), \mathfrak{T}, Form, Snt \rangle$ nyelv egy interpretációja olyan $\langle U, \varrho \rangle$ rendezett pár, ahol

1. $U \neq \emptyset$, és
2. $\varrho : \mathcal{P}(1) \rightarrow \{0, 1\}^U$,
3. $[\varrho(T)]^+ \neq \emptyset$, ha $T \in \mathfrak{T}$.

2.1.3. Approximációs tér Pawlak-féle approximációs párral

2.3. Definíció. Az $\langle U, \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{l}, \mathfrak{u} \rangle$ rendezett ötös egy általánosított parciális approximációs tér Pawlak-féle approximációs párral, ha

1. U egy nemüres halmaz;
2. $\mathfrak{B} \subseteq 2^U \setminus \{\emptyset\}$ tartalmazza az alaphalmazokat, ahol $\mathfrak{B} \neq \emptyset$;
3. $\mathfrak{D}_{\mathfrak{B}}$ tartalmazza a definiálható halmazokat, ami $\mathfrak{B}$ olyan $\mathfrak{B} \subseteq \mathfrak{D}_{\mathfrak{B}}$ kiterjesztése, hogy
 - (a) $\emptyset \in \mathfrak{D}_{\mathfrak{B}}$, és
 - (b) $\bigcup B \in \mathfrak{D}_{\mathfrak{B}}$ minden nemüres $B \subseteq \mathfrak{B}$ esetén;
4. az $\mathfrak{l}$ és $\mathfrak{u}$ függvények egy $\langle \mathfrak{l}, \mathfrak{u} \rangle$ Pawlak-féle approximációs párt alkotnak, ahol
 - (a) az alsó közelítése egy $S \in 2^U$ halmaznak

$$\mathfrak{l}(S) \stackrel{\text{def}}{=} \bigcup \{B : B \in \mathfrak{B} \text{ és } B \subseteq S\}, \quad (2.1)$$

- (b) a felső közelítése egy $S \in 2^U$ halmaznak

$$\mathfrak{u}(S) \stackrel{\text{def}}{=} \bigcup \{B : B \in \mathfrak{B} \text{ és } B \cap S \neq \emptyset\}. \quad (2.2)$$

A Pawlak-féle approximációs párra azért esett a választás, mert jól ismert és széles körben elterjedt.

2.4. Definíció. Legyen $\mathcal{GAS} = \langle U, \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{l}, \mathfrak{u} \rangle$ az $L = \langle LC, Var, \mathcal{P}(1), \mathfrak{T}, Form, Snt \rangle$ nyelv egy $\langle U, \varrho \rangle$ interpretációja által generált approximációs tér, ha

1. $\mathfrak{B} = \left\{ [\varrho(T)]^+ : T \in \mathfrak{T} \right\}$ és
2. $\langle \mathfrak{I}, \mathfrak{u} \rangle$ egy Pawlak-féle approximációs pár.

A fenti definíció esetén minden L nyelvhez és a hozzá tartozó $\langle U, \varrho \rangle$ interpretációhoz tartozó parciális approximációs tér egyértelműen létezik. Egyfelől eleve csak az olyan $\langle U, \varrho \rangle$ párokat tekintettük interpretációnak, amelyek esetében az eszköznek választott predikátumparaméterekhez tartozó logikai függvények pozitivitási tartománya nem üres (tehát $\emptyset \notin \mathfrak{B}$ teljesül). Másfelől a Pawlak-féle approximációs párra szűkítettük az approximációs terek körét.

2.1.4. Szemantika

Legyen $\langle U, \varrho \rangle$ az L nyelv egy interpretációja, $\langle U, \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{I}, \mathfrak{u} \rangle$ az interpretáció által generált parciális approximációs tér és v az interpretációhoz tartozó változókiértékelés.

2.5. Definíció. Az L nyelv egy $P \in \mathcal{P}(1)$ predikátumparaméterének szemantikai értéke az $\langle U, \varrho \rangle$ interpretációban a v változókiértékelés mellett a következő $\llbracket P \rrbracket_v^{\langle U, \varrho \rangle}$ -vel jelölt $\{0, 1, 2\}^U$ halmazbeli függvény:

1. ha $P \in \mathcal{P}(1) \setminus \mathfrak{T}$, akkor $\llbracket P \rrbracket_v^{\langle U, \varrho \rangle} = \varrho(P)$;
2. ha $P \in \mathfrak{T}$, akkor

$$\left(\llbracket P \rrbracket_v^{\langle U, \varrho \rangle} \right)(u) = \begin{cases} 1 & \text{ha } u \in [\varrho(P)]^+, \\ 0 & \text{ha } u \in \mathfrak{I}(U \setminus [\varrho(P)]^+), \\ 2 & \text{egyébként.} \end{cases}$$

2.6. Definíció. Egy $\psi : U \rightarrow \{0, 1, 2\}$ függvény alsó közelítése alatt a következőképpen definiált $\psi^\downarrow : U \rightarrow \{0, 1, 2\}$ leképezést értjük:

$$\psi^\downarrow(u) = \begin{cases} 1 & \text{ha } u \in \mathfrak{I}([\psi]^+), \\ 0 & \text{ha } u \in \mathfrak{I}(U \setminus \mathfrak{u}([\psi]^+)), \\ 2 & \text{egyébként.} \end{cases}$$

2.7. Definíció. Egy $\psi : U \rightarrow \{0, 1, 2\}$ függvény felső közelítése alatt a következőképpen definiált $\psi^\uparrow : U \rightarrow \{0, 1, 2\}$ leképezést értjük:

$$\psi^\uparrow(u) = \begin{cases} 1 & \text{ha } u \in \mathfrak{u}([\psi]^+), \\ 0 & \text{ha } u \in \mathfrak{I}(U \setminus \mathfrak{u}([\psi]^+)), \\ 2 & \text{egyébként.} \end{cases}$$

A $\psi^\downarrow$ és $\psi^\uparrow$ függvények definíciója rendre (1.7) és (1.8) egyszerűsítése. Ha P egy predikátumparaméter a közelítő funktorokat tartalmazó kvantormentes nyelvben, amelynek szemantikai értéke adott interpretációban ψ , úgy $\psi^\downarrow$ és $\psi^\uparrow$ fejezi ki annak

alsó és felső közelítését. Minderre azért van szükség, mert a $P^\downarrow$ és $P^\uparrow$ szimbólumok nem részei az egyszerűsített nyelvnek.

2.8. Definíció. Egy $L = \langle LC, Var, \mathcal{P}(1), \mathfrak{T}, Form, Snt \rangle$ logikai nyelv $F \in Snt$ mondatának szemantikai értéke a nyelvhez tartozó adott $\langle U, \varrho \rangle$ interpretációban és az interpretációhoz tartozó v változókiértékelés mellett az $\llbracket F \rrbracket_v^{(U, \varrho)}$ -vel jelölt és az alábbi rekurzív definícióval megadott $\{0, 1, 2\}$ -beli érték:

1. ha $P \in \mathcal{P}(1)$ predikátumparaméter, akkor

$$\begin{aligned}\llbracket \downarrow P(x) \rrbracket_v^{(U, \varrho)} &= \psi^\downarrow(v(x)), \\ \llbracket \uparrow P(x) \rrbracket_v^{(U, \varrho)} &= \psi^\uparrow(v(x)), \\ \llbracket P(x) \rrbracket_v^{(U, \varrho)} &= \psi(v(x))\end{aligned}$$

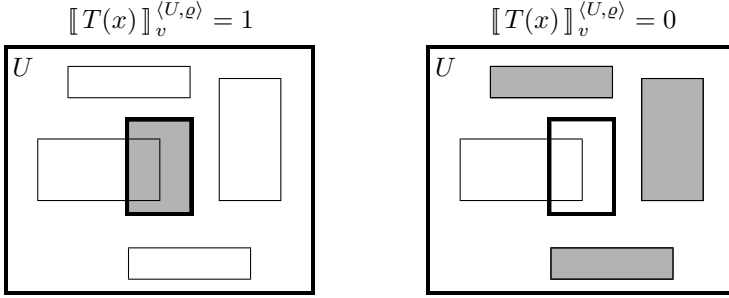
ahol $\psi = \llbracket P \rrbracket_v^{(U, \varrho)}$ a P predikátumparaméter szemantikai értéke;

2. ha $A, B \in Form$ és az $\mathfrak{f}$ helyén vagy nem szerepel szimbólum, vagy a $\uparrow$, illetve $\downarrow$ mondatfunktorok egyikét jelöli (azaz $\mathfrak{f} A \in Snt$), akkor

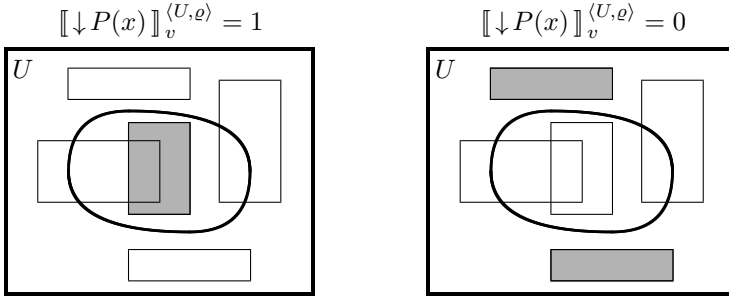
$$\begin{aligned}\llbracket \mathfrak{f} + A \rrbracket_v^{(U, \varrho)} &= \begin{cases} 1 & \text{ha } \llbracket \mathfrak{f} A \rrbracket_v^{(U, \varrho)} = 1, \\ 0 & \text{egyébként;} \end{cases} \\ \llbracket \mathfrak{f} \neg A \rrbracket_v^{(U, \varrho)} &= \begin{cases} 2 & \text{ha } \llbracket \mathfrak{f} A \rrbracket_v^{(U, \varrho)} = 2, \\ 1 & \text{ha } \llbracket \mathfrak{f} A \rrbracket_v^{(U, \varrho)} = 0, \\ 0 & \text{egyébként;} \end{cases} \\ \llbracket \mathfrak{f} (A \supset B) \rrbracket_v^{(U, \varrho)} &= \begin{cases} 0 & \text{ha } \llbracket \mathfrak{f} A \rrbracket_v^{(U, \varrho)} = 1 \text{ és } \llbracket \mathfrak{f} B \rrbracket_v^{(U, \varrho)} = 0, \\ 2 & \text{ha } \llbracket \mathfrak{f} A \rrbracket_v^{(U, \varrho)} = 2 \text{ vagy } \llbracket \mathfrak{f} B \rrbracket_v^{(U, \varrho)} = 2, \\ 1 & \text{egyébként.} \end{cases}\end{aligned}$$

Az egyszerűsített logikai nyelvben annak köszönhetően, hogy a predikátumparaméterek pontosan egy argumentummal rendelkezhetnek, egyszerűen tudjuk illusztrálni a fontosabb szemantikai definíciókat. Ehhez most egy olyan nyelvet használunk, ami 5 darab $\mathfrak{T}$ -beli eszközt és további egy $\mathcal{P}(1) \setminus \mathfrak{T} = \{P\}$ predikátumparamétert tartalmaz. Minden $T \in \mathfrak{T}$ eszköz ϱ interpretáció által meghatározott $[\varrho(T)]^+$ pozitivitási tartományát, mint az U univerzum valódi részhalmazát, egy-egy téglalappal, míg a $[\varrho(P)]^+$ halmazt egy ovális vonallal határolt területtel fogjuk jelölni. Egyben az így kapott téglalapok reprezentálják az approximációs tér alaphalmazait. A 2.1. ábra a közepén vastag folytonos vonallal határolt téglalap alakú területhez tartozó T eszköz segítségével előállított $T(x)$ formula szemantikai értékét illusztrálja. A bal oldalon a szürkére színezett terület azon $u \in U$ univerzumelemeket jelöli, amelyek esetén $\llbracket T(x) \rrbracket_v^{(U, \varrho)} = 1$, feltéve hogy $v(x) = u$. A jobb oldalon hasonlóképpen azon univerzumelemek vannak kiemelve, amelyek esetében $\llbracket T(x) \rrbracket_v^{(U, \varrho)} = 0$. Egyszerűbben szólva azt mondhatjuk, hogy a bal oldalon a T predikátumhoz tartozó parciális logikai függvény pozitivitási tartománya, jobb oldalon pedig a negativitási tartománya van kiemelve. A 2.2. és 2.3. ábrákon P alsó és felső közelítésének pozitivitási, illetve negativitási tartományát látjuk.

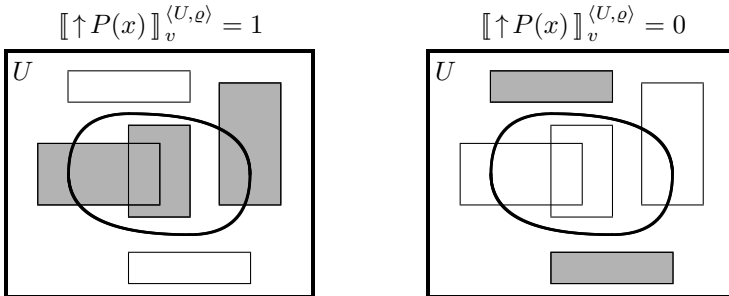
2.1. ábra. A $T \in \mathfrak{T}$ eszköz szemantikája



2.2. ábra. Az alsó közelítés illusztrációja



2.3. ábra. A felső közelítés illusztrációja



2.1.5. Centrális logikai fogalmak

A centrális logikai fogalmak megadásakor kihasználjuk, hogy az interpretáció egyértelműen meghatározza a parciális approximációs teret.

2.9. Definíció. Legyen adott egy $L = \langle LC, Var, \mathcal{P}(1), \mathfrak{T}, Form, Snt \rangle$ nyelv és annak egy $\langle U, \varrho \rangle$ interpretációja, továbbá egy, az interpretációhoz tartozó v változókiértékelés. Amennyiben $\Gamma \subseteq Snt$ egy mondathalmaz, akkor azt mondjuk, hogy

1. az $\langle U, \varrho, v \rangle$ rendezett hármas a Γ mondathalmaz egy reprezentációja, ha minden $B \in \Gamma$ esetén teljesül, hogy $\llbracket B \rrbracket_v^{\langle U, \varrho \rangle} \neq 2$;
2. az $\langle U, \varrho, v \rangle$ rendezett hármas a Γ mondathalmaz egy modellje, ha minden $B \in \Gamma$ esetén teljesül, hogy $\llbracket B \rrbracket_v^{\langle U, \varrho \rangle} = 1$;
3. a Γ mondathalmaz kielégíthető, ha létezik modellje;
4. az $A \in Snt$ mondat a Γ mondathalmaz gyenge szemantikai következménye – és ezt $\Gamma \models_w A$ -ként jelöljük –, ha Γ minden $\langle U, \varrho, v \rangle$ modellje esetén $\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} \neq 0$;
5. az $A \in Snt$ mondat cáfolhatatlan, ha $\emptyset \models_w A$;
6. az $A \in Snt$ mondat a Γ mondathalmaz erős szemantikai következménye – és ezt $\Gamma \models_s A$ -ként jelöljük –, ha
 - (a) Γ -nak létezik reprezentációja,
 - (b) Γ minden reprezentációja $\{A\}$ -nak is reprezentációja,
 - (c) Γ minden modellje $\{A\}$ -nak is modellje;
7. az $A \in Snt$ mondat érvényes, ha $\emptyset \models_s A$.

Megjegyzés. Ha egy $\Gamma \cup \{f \neg A\}$ mondathalmaz (ahol $A \in Form$ és $f A \in Snt$) kielégíthető, azaz van olyan $\langle U, \varrho, v \rangle$, amely modellje, akkor

- $\Gamma \not\models_w f A$, mivel ekkor $\langle U, \varrho, v \rangle$ modellje Γ -nak is, és $\llbracket f A \rrbracket_v^{\langle U, \varrho \rangle} = 0$;
- $\Gamma \not\models_s f A$, mivel ekkor $\langle U, \varrho, v \rangle$ ugyan Γ -nak reprezentációja, de egyben olyan modellje is, ami $\{f A\}$ halmaznak nem modellje.

Abban az esetben, ha $\Gamma \not\models_w f A$, úgy kell legyen olyan $\langle U, \varrho, v \rangle$ modellje Γ -nak, ahol $\llbracket f A \rrbracket_v^{\langle U, \varrho \rangle} = 0$. Mivel ebben a modellben $\llbracket f \neg A \rrbracket_v^{\langle U, \varrho \rangle} = 1$, ezért $\langle U, \varrho, v \rangle$ a $\Gamma \cup \{f \neg A\}$ mondathalmaznak is modellje. Ennél fogva, ha a későbbiekben cáfolni szeretnénk, hogy a Γ mondathalmaznak az $f A$ mondat erős vagy gyenge szemantikai következménye, akkor a $\Gamma \cup \{f \neg A\}$ mondathalmaz kielégíthetőségét fogjuk igazolni. Ugyanakkor, ha a $\Gamma \cup \{f \neg A\}$ mondathalmaz kielégíthetetlennek bizonyul, az alátámasztja, hogy $\Gamma \models_w f A$.

Elsőként a következő tételben a $+$ művelet szemantikájának sajátosságaira szeretnénk rámutatni, amelyet a későbbi tételek bizonyítása során is fel fogunk használni.

2.10. Tétel. *Tetszőleges $A \in \text{Form}$ formula esetében igaz, hogy $\models_s (+\neg A \supset \neg +A)$, továbbá megválasztható úgy az A formula, hogy $\not\models_w (\neg +A \supset +\neg A)$.*

Bizonyítás. Elsőként jegyezzük meg, hogy a $+$ logikai konstans szemantikája miatt

$$\llbracket (+\neg A \supset \neg +A) \rrbracket_v^{(U,\varrho)} \in \{0, 1\}.$$

Indirekt tételezzük fel, hogy $\llbracket (+\neg A \supset \neg +A) \rrbracket_v^{(U,\varrho)} = 0$, ekkor a következő megállapításokat tehetjük az implikáció elő- és utótagjára:

$$\begin{aligned} \llbracket +\neg A \rrbracket_v^{(U,\varrho)} = 1 &\Rightarrow \llbracket \neg A \rrbracket_v^{(U,\varrho)} = 1 \Rightarrow \llbracket A \rrbracket_v^{(U,\varrho)} = 0, \text{ és} \\ \llbracket \neg +A \rrbracket_v^{(U,\varrho)} = 0 &\Rightarrow \llbracket +A \rrbracket_v^{(U,\varrho)} = 1 \Rightarrow \llbracket A \rrbracket_v^{(U,\varrho)} = 1, \text{ ami ellentmondás.} \end{aligned}$$

Mivel értékrés ezen formula esetében nem jelenhet meg, így igazságértéke csak igaz lehet. A tétel második felének igazolásához tegyük fel, hogy van olyan interpretáció és változókiértékelés, ahol $\llbracket A \rrbracket_v^{(U,\varrho)} = 2$. Ekkor az alábbi megállapításokat tehetjük:

$$\begin{aligned} \llbracket A \rrbracket_v^{(U,\varrho)} = 2 &\Rightarrow \llbracket +A \rrbracket_v^{(U,\varrho)} = 0 \Rightarrow \llbracket \neg +A \rrbracket_v^{(U,\varrho)} = 1, \text{ és} \\ \llbracket A \rrbracket_v^{(U,\varrho)} = 2 &\Rightarrow \llbracket \neg A \rrbracket_v^{(U,\varrho)} = 2 \Rightarrow \llbracket +\neg A \rrbracket_v^{(U,\varrho)} = 0, \text{ ezért} \\ \llbracket \neg +A \supset +\neg A \rrbracket_v^{(U,\varrho)} &= 0. \end{aligned}$$

Innen már csak annyit kell megmutatni, hogy a feltevésben megadott interpretáció és A formula létezik. Válasszuk A -t $T(x)$ -nek, ahol $T \in \mathfrak{T}$ egy eszköz a nyelvben (legalább egy eszköz minden nyelvben szerepel). Legyen $U = \{u_1, u_2\}$ és ϱ olyan, hogy minden $P \in \mathcal{P}(1)$ esetén (így a $T \in \mathcal{P}(1)$ eszköz esetében is):

$$\varrho(P)(x) = \begin{cases} 1 & \text{ha } x = u_1, \\ 0 & \text{egyébként;} \end{cases}$$

továbbá $v(x) = u_2$. Tekintettel arra, hogy u_2 egyetlen eszköz pozitivitási tartományában sem szerepel, $\llbracket T \rrbracket_v^{(U,\varrho)}(u_2) = 2$, és így

$$\llbracket T(x) \rrbracket_v^{(U,\varrho)} = 2.$$

Mivel megmutattuk, hogy tetszőleges nyelven megkonstruálható a megfelelő A formula, így a tétel második fele is bizonyítást nyert. $\blacksquare$

Megjegyzés. Ez egyben példa arra is, hogy értékrés még akkor is keletkezhet, ha nem használtunk közelítő mondatfunktorokat.

2.1.6. Formulahalmazok kielégíthetősége

A következőkben a vizsgálat célja, hogy megmutassuk vagy épp megcáfoljuk, hogy a parciális logikai rendszer viselkedése egybeesik a klasszikus esetben tapasztaltakkal.

A formulahalmazok kielégíthetőségét formulák és interpretációk konstruálásával fogjuk igazolni, ehhez kezdetben annyit feltételezünk, hogy a vizsgált nyelvben van

egy $P \in \mathcal{P}(1) \setminus \mathfrak{T}$ predikátumparaméter (és természetesen egy $T \in \mathfrak{T}$ eszköz). Ennek megfelelően legyen az L_1 nyelv definiálva a következőképpen:

$$L_1 = \langle \{\neg, \supset, +, \downarrow, \uparrow, (,)\}, \{x, y, z, \dots\}, \{P, T\}, \{T\}, \text{Form}, \text{Snt} \rangle$$

Ekkor konstruálható egy olyan $\langle U_1, \varrho_1 \rangle$ interpretáció és v_1 változókiértékelés, ahol van olyan $u \in U$ univerzumelem, amely esetén

1. $v_1(x) = u$,
2. $\llbracket \downarrow P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 2$, és
3. $\llbracket \uparrow P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1$.

Ez mindössze annyit jelent, hogy az u érték a P predikátum pozitivitási tartományának felső közelítésében benne van, de az alsó közelítésében nincs. Mivel P alsó közelítésének negativitási tartománya a felső közelítés komplementerének alsó közelítése, így u ebbe sem tartozhat bele, azaz értékrés keletkezik. Példaként legyen az interpretáció univerzuma $U_1 = \{u, u', u''\}$, továbbá

$$\begin{aligned} \varrho_1(P) &= P_1, & \text{ahol} & \quad P_1(u) = 1, \quad P_1(u') = 0, \quad P_1(u'') = 1, \\ \varrho_1(T) &= T_1, & \text{ahol} & \quad T_1(u) = 1, \quad T_1(u') = 1, \quad T_1(u'') = 0. \end{aligned}$$

Az interpretáció által létrejött approximációs térben, mivel P_1 és T_1 pozitivitási tartományának van metszete, de nem részhalmazai egymásnak, P_1 pozitivitási tartományának alsó közelítése üres lesz, míg felső közelítése $\{u, u'\}$. Ebben az interpretációban így valóban azt kapjuk, hogy $\llbracket \uparrow P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1$, és $\llbracket \downarrow P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 2$.

Elsőként tekintsük a klasszikus logikában $\{A, \neg A\}$ alakban felírt formulahalmaz tulajdonságait, amelyet a parciális logikában a következő két tétel segítségével jellemezünk.

2.11. Tétel. *Az $\{\mathfrak{f} A, \mathfrak{f} \neg A\}$ mondathalmaz kielégíthetetlen, feltéve hogy $A \in \text{Form}$ egy tetszőleges formula, továbbá az $\mathfrak{f}$ helyén vagy nem szerepel szimbólum, vagy egy közelítő mondatfunktort jelöl.*

Bizonyítás. Indirekt tegyük fel, hogy a mondathalmaz kielégíthető. Ekkor van olyan $\langle U, \varrho \rangle$ interpretáció és hozzá tartozó v változókiértékelés, amely esetén $\llbracket \mathfrak{f} A \rrbracket_v^{\langle U, \varrho \rangle} = 1$ és $\llbracket \mathfrak{f} \neg A \rrbracket_v^{\langle U, \varrho \rangle} = 1$ egyszerre teljesül. Ugyanakkor a negáció szemantikájából következik, hogy ha $\llbracket \mathfrak{f} \neg A \rrbracket_v^{\langle U, \varrho \rangle} = 1$, akkor $\llbracket \mathfrak{f} A \rrbracket_v^{\langle U, \varrho \rangle} = 0$, ami ellentmondás. ■

2.12. Tétel. *Van olyan $A \in \text{Form}$ formulája az L_1 nyelvnek, amely esetén a $\{\uparrow A, \downarrow \neg A\}$ és $\{\downarrow A, \uparrow \neg A\}$ mondathalmazok kielégíthetőek.*

Bizonyítás. Az állítás bizonyításához elegendő egy megfelelő A formulát konstruálni. Elsőként legyen az A formula $+P(x)$, ekkor

$$\begin{aligned} \llbracket \uparrow A \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} &= \llbracket \uparrow +P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1, & \text{hiszen} & \quad \llbracket \uparrow P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1, \\ \llbracket \downarrow \neg A \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} &= \llbracket \downarrow \neg +P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1, & \text{hiszen} & \quad \llbracket \downarrow +P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 0, \end{aligned}$$

így $\{\uparrow + P(x), \downarrow \neg + P(x)\}$ valóban kielégíthető. Válasszuk most az A formulát $\neg + P(x)$ -nek, ekkor

$$\begin{aligned} \llbracket \downarrow A \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} &= \llbracket \downarrow \neg + P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1, \text{ hiszen } \llbracket \downarrow + P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 0, \\ \llbracket \uparrow \neg A \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} &= \llbracket \uparrow \neg \neg + P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1, \text{ hiszen } \llbracket \uparrow + P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1, \end{aligned}$$

így $\{\downarrow \neg + P(x), \uparrow \neg \neg + P(x)\}$ szintén kielégíthető. Ezzel az állítás mindkét része bizonyítást nyert. ■

Megjegyzés. Az, hogy egy mondathalmaz kielégíthetőségét az L_1 nyelven igazoljuk annak $\langle U_1, \varrho_1 \rangle$ interpretációja segítségével, nem jelent valódi megszorítást. Ugyanígy kell eljárni tetszőleges, L_1 -nél bővebb (több predikátumparamétert és eszközt tartalmazó) nyelv esetében is. Ez esetben a bővebb nyelvet interpretálhatjuk az U_1 univerzumon úgy, hogy minden eszközhöz a T_1 , és az eszközöktől különböző minden predikátumparaméterhez pedig a P_1 függvényt rendeljük. A nyelvvel szemben támasztott egyetlen követelmény mindössze az, hogy tartalmazzon legalább egy, az eszközöktől különböző predikátumparamétert.

2.1.7. Modus ponens

Az alapvető logikai törvények vizsgálata elképzelhetetlen a *modus ponens* vizsgálata nélkül¹. Először is azt kell megállapítsuk, hogy a

„Ha mind az A mind az $(A \supset B)$ formulák igazak, akkor a B formula is az.”

állítás a vizsgált rendszerben is érvényes, feltéve hogy nem használtunk közelítő mondatfunktorokat. A kérdés vizsgálata azért is releváns, mert a mondatfunktorok hiánya nem jelenti azt, hogy mindenben a klasszikus logikával megegyező konstrukciónk van. Az értékrés ugyanis – amint azt korábban bemutattuk – az eszközeink szemantikája miatt ebben az esetben is megjelenhet.

2.13. Tétel. Az $\{\mathfrak{f} A, \mathfrak{f}(A \supset B), \mathfrak{f} \neg B\}$ mondathalmaz kielégíthetetlen, feltéve hogy $A, B \in \text{Form}$ tetszőleges formulák, továbbá az $\mathfrak{f}$ helyén vagy nem szerepel szimbólum, vagy egy közelítő mondatfunktort jelöl.

Bizonyítás. Az indirekt bizonyítás az implikáció és negáció szemantikáján alapul. Tegyük fel, hogy a formulahalmaz kielégíthető, így

1. $\llbracket \mathfrak{f} A \rrbracket_v^{\langle U, \varrho \rangle} = 1$, $\llbracket \mathfrak{f} \neg B \rrbracket_v^{\langle U, \varrho \rangle} = 1$ és $\llbracket \mathfrak{f}(A \supset B) \rrbracket_v^{\langle U, \varrho \rangle} = 1$;
2. $\llbracket \mathfrak{f} \neg B \rrbracket_v^{\langle U, \varrho \rangle} = 1$ esetén a negáció szemantikája miatt $\llbracket \mathfrak{f} B \rrbracket_v^{\langle U, \varrho \rangle} = 0$;
3. ekkor az implikáció szemantikájából következően $\llbracket \mathfrak{f}(A \supset B) \rrbracket_v^{\langle U, \varrho \rangle} = 0$.

Ezzel az implikációs formula értékére vonatkozóan ellentmondás állt elő. ■

¹A *modus ponens* és a *modus tollens* a klasszikus logika talán legismertebb kétpremisszás következtetései.

A 2.13. tétel pozitív választ ad a kérdésre, nemcsak a szokásos formulák, hanem az azonos közelítő mondatfunktorok használata esetében is:

$$\{ A, (A \supset B) \} \models_w B \quad \text{és} \quad \{ \mathfrak{f} A, \mathfrak{f}(A \supset B) \} \models_w \mathfrak{f} B.$$

A következőkben azt fogjuk vizsgálni, hogy miben változik a helyzet akkor, ha különböző közelítő mondatfunktorok jelennek meg a premisszák és a konklúzió oldalán.

„Ha tudjuk, hogy A és $(A \supset B)$ alsó közelítése igaz,
akkor ebből következik-e, hogy B felső közelítése is igaz?”

A motiváció abból a sejtésből adódik, hogy az alsó közelítést hajlamosak vagyunk a biztos tudásunkként tekinteni.

2.14. Tétel. $\{ \downarrow P(x), \downarrow (P(x) \supset Q(x)), \uparrow \neg Q(x) \}$ mondatthalmaz kielégíthetetlen, feltéve hogy $P, Q \in \mathcal{P}(1)$ predikátumparaméterek és $x \in \text{Var}$ egy változó.

Bizonyítás. Indirekt tegyük fel, hogy van olyan $\langle U, \varrho \rangle$ interpretáció és hozzá tartozó v változókiértékelés, amely esetén

$$\llbracket \downarrow (P(x) \supset Q(x)) \rrbracket_v^{\langle U, \varrho \rangle} = 1, \quad \llbracket \downarrow P(x) \rrbracket_v^{\langle U, \varrho \rangle} = 1, \quad \llbracket \uparrow Q(x) \rrbracket_v^{\langle U, \varrho \rangle} = 0.$$

Ez esetben az implikáció szemantikája miatt $\llbracket \downarrow Q(x) \rrbracket_v^{\langle U, \varrho \rangle} = 1$. $\llbracket \uparrow Q(x) \rrbracket_v^{\langle U, \varrho \rangle} = 0$ azt jelenti, hogy a $v(x) \in U$ érték a $\llbracket Q \rrbracket_v^{\langle U, \varrho \rangle}$ felső közelítésének negativitási tartományába tartozik, ekkor azonban $\llbracket Q \rrbracket_v^{\langle U, \varrho \rangle}$ alsó közelítésének negativitási tartományában is szerepel, mivel az alsó és felső közelítés negativitási tartományát megegyező módon definiáltuk. Ekkor viszont $\llbracket \downarrow Q(x) \rrbracket_v^{\langle U, \varrho \rangle} = 0$ ellentmondás. ■

A 2.13. tétel alátámasztja, amit az alsó és felső közelítés kapcsolatáról feltételeztünk. Ugyanakkor, ha az A és B formulák helyébe nem csak atomi formulát helyettesíthetünk, megváltozik a helyzet.

2.15. Tétel. Van olyan $A \in \text{Form}$ és $B \in \text{Form}$ formulája az L_1 nyelvnek, amely esetében $\{ \downarrow A, \downarrow (A \supset B), \uparrow \neg B \}$ kielégíthető.

Bizonyítás. A bizonyításhoz fel fogjuk használni a korábban bevezetett L_1 nyelv $P \in \mathcal{P}(1)$ predikátumparaméterét, valamint az $\langle U_1, \varrho_1 \rangle$ interpretációját. Ahhoz, hogy a formulahalmaz kielégíthetőségét bizonyítsuk, elegendő olyan A és B formulákat konstruálni, amelyek esetében

$$\llbracket \downarrow A \rrbracket_v^{\langle U, \varrho \rangle} = 1, \quad \llbracket \downarrow (A \supset B) \rrbracket_v^{\langle U, \varrho \rangle} = 1, \quad \llbracket \uparrow B \rrbracket_v^{\langle U, \varrho \rangle} = 0.$$

A korábban bizonyított tételek (és az implikáció szemantikája) miatt tudjuk, hogy $\llbracket \downarrow B \rrbracket_v^{\langle U, \varrho \rangle} = 1$ szintén teljesülhet. Egy nagyon egyszerű példa olyan A formulára, amely minden interpretációban igaz, lehet a következő:

$$A \stackrel{\text{def}}{=} (+P(x) \supset +P(x)),$$

ahol az implikációs formula elő- és utótagjának igazságértéke azonos, és a digitalizáló $+$ művelet miatt garantáltan nem lehet 2-es. Korábban azt is sikerült belátni,

hogy a $\{\downarrow B, \uparrow \neg B\}$ mondathalmaz kielégíthető, és ehhez elegendő a következőképp megválasztani a B formulát:

$$B \stackrel{\text{def}}{=} \neg +P(x).$$

Ekkor az alábbiak teljesülnek:

1. $\llbracket \downarrow A \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1$ és $\llbracket \downarrow B \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1$, ebből következően
2. $\llbracket \downarrow (A \supset B) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1$,
3. $\llbracket \uparrow \neg B \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1$, hiszen $\llbracket \uparrow B \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 0$,

és ezzel az állítást igazoltuk. ■

Hasonló eredményeket kapunk abban az esetben is, ha ellentétes irányú közelítő mondatfunktorokat használunk, azaz az alábbi kérdést vizsgáljuk:

*„Ha tudjuk, hogy A és $(A \supset B)$ felső közelítése igaz,
akkor ebből következik-e, hogy B alsó közelítése is igaz?”*

2.16. Tétel. $A \{\uparrow P(x), \uparrow (P(x) \supset Q(x)), \downarrow \neg Q(x)\}$ mondathalmaz kielégíthetetlen, feltéve hogy $P, Q \in \mathcal{P}(1)$ predikátumparaméterek és $x \in \text{Var}$ egy változó.

Bizonyítás. Indirekt tegyük fel, hogy van olyan $\langle U, \varrho \rangle$ interpretáció és hozzá tartozó v változókiértékelés, amely esetén

$$\llbracket \uparrow (P(x) \supset Q(x)) \rrbracket_v^{\langle U, \varrho \rangle} = 1, \quad \llbracket \uparrow P(x) \rrbracket_v^{\langle U, \varrho \rangle} = 1, \quad \llbracket \downarrow Q(x) \rrbracket_v^{\langle U, \varrho \rangle} = 0.$$

Ez esetben az implikáció szemantikája miatt az első két mondat igazságértékéből következik $\llbracket \uparrow Q(x) \rrbracket_v^{\langle U, \varrho \rangle} = 1$. Ugyanakkor, ha $\llbracket \downarrow Q(x) \rrbracket_v^{\langle U, \varrho \rangle} = 0$, akkor (a 2.14. tétel bizonyításában már kifejtett okból) $\llbracket \uparrow Q(x) \rrbracket_v^{\langle U, \varrho \rangle} = 0$ ellentmondást kapunk. ■

2.17. Tétel. Van olyan $A \in \text{Form}$ és $B \in \text{Form}$ formulája az L_1 nyelvnek, amely esetében az $\{\uparrow (A \supset B), \uparrow \neg B, \downarrow \neg A\}$ mondathalmaz kielégíthető.

Bizonyítás. Mivel a bizonyítás lényegében nem különbözik az előző esettől, itt csak az A és B formulákat adjuk meg, rendre mint $+P(x)$ és $\neg(+P(x) \supset +P(x))$. ■

2.1.8. Modus tollens

A *modus ponens*-hez hasonló megállapításokat tehetünk a *modus tollens* esetén is. Ez esetben is elmondható, hogy a

„Ha az $(A \supset B)$ és $\neg B$ formulák igazak, akkor a $\neg A$ formula is az.”

állítás a vizsgált rendszerben is érvényes, amennyiben nem használunk közelítő mondatfunktorokat. Ugyancsak tovább általánosíthatunk arra az esetre, amikor a formulák előtt azonos közelítő mondatfunktor szerepel:

$$\{ (A \supset B), \neg B \} \models_w \neg A \quad \text{és} \quad \{ \mathfrak{f}(A \supset B), \mathfrak{f}\neg B \} \models_w \mathfrak{f}\neg A.$$

Az atomi formulák, illetve az atomi formulákból képzett mondatok esetében sem változik a helyzet:

$$\{ \downarrow(P(x) \supset Q(x)), \downarrow\neg Q(x) \} \models_w \uparrow\neg P(x),$$

$$\{ \uparrow(P(x) \supset Q(x)), \uparrow\neg Q(x) \} \models_w \downarrow\neg P(x);$$

de tetszőleges formulákat megengedve ismét lehetségessé válik a cáfolatok konstruálása.

2.18. Tétel. *Van olyan $A \in Form$, $B \in Form$ és $C \in Form$ formulája az L_1 nyelvnek, amely esetében az alábbi formulahalmazok kielégíthetők:*

$$\{ \downarrow(A \supset B), \downarrow\neg B, \uparrow\neg\neg A \} \text{ és } \{ \uparrow(C \supset B), \uparrow\neg B, \downarrow\neg\neg C \}.$$

Bizonyítás. A részletes bizonyítás helyett álljon itt egy-egy példa a három formulára:

$$A \stackrel{def}{=} +P(x), \quad B \stackrel{def}{=} \neg(+P(x) \supset +P(x)), \quad C \stackrel{def}{=} \neg+P(x).$$

■

2.2. Parciális elsőrendű logika

Annak érdekében, hogy valódi elsőrendű formulákat is vizsgálhassunk, kiegészítjük az egyszerűsített nyelvet az univerzális és egzisztenciális kvantorokkal:

A $Form$ halmaz induktív definíciójához adjuk hozzá az alábbi két indukciós lépést (és vegyük fel a logikai konstansok közé a kvantorokat): ha $A \in Form$ és $x \in Var$, akkor

$$\forall x A \in Form \quad \text{és} \quad \exists x A \in Form.$$

A szemantikát változatlanul átvehetjük, ahol f egy közelítő mondatfunktort jelölhet (vagy elmaradhat).

$$\begin{aligned} \llbracket f \forall x A \rrbracket_v^{(U, \varrho)} &= \begin{cases} 0 & \text{ha van olyan } u \in U, \text{ amely esetén } \llbracket f A \rrbracket_{v[x:u]}^{(U, \varrho)} = 0, \\ 2 & \text{ha minden } u \in U \text{ esetén } \llbracket f A \rrbracket_{v[x:u]}^{(U, \varrho)} = 2, \\ 1 & \text{egyébként.} \end{cases} \\ \llbracket f \exists x A \rrbracket_v^{(U, \varrho)} &= \begin{cases} 1 & \text{ha van olyan } u \in U, \text{ amely esetén } \llbracket f A \rrbracket_{v[x:u]}^{(U, \varrho)} = 1, \\ 2 & \text{ha minden } u \in U \text{ esetén } \llbracket f A \rrbracket_{v[x:u]}^{(U, \varrho)} = 2, \\ 0 & \text{egyébként.} \end{cases} \end{aligned}$$

Az univerzális kvantorok definíciójában a klasszikus modell azon tulajdonságát vettük át, hogy az univerzális formula hamis, ha található olyan helyettesítés, amely szerint a közvetlen részformula hamis (és ebben az értelemben az univerzális kvantor tagadást fejez ki). Hasonlóképp jártunk el az egzisztenciális kvantor esetében.

Az eddigi tételek bizonyítása során mindig a $+$ művelet tulajdonságait használtuk ki, egészen pontosan arra építettünk, hogy a művelet nem tartja meg az értékrést. A

kvantorok szemantikája is hasonló lehetőséget ad a kezünkbe. Így a korábbi tételek egy elsőrendű, de a $+$ műveletet nem tartalmazó nyelvben is érvényesek.

2.19. Tétel. *Konstruálható olyan $B \in \text{Form}$ formula megfelelő logikai nyelven, amely esetében a $\{\uparrow B, \downarrow \neg B\}$ formulahalmaz kielégíthető úgy, hogy B -ben a $+$ művelet nem fordul elő.*

Bizonyítás. Vegyük a következő elsőrendű nyelvet (amely most már kvantorokat is tartalmaz):

$$L_2 = \langle \{\neg, \supset, +, \downarrow, \uparrow, (,), \forall, \exists\}, \{x, y, \dots\}, \{Q, S, T\}, \{S, T\}, \text{Form}, \text{Snt} \rangle$$

Majd készítsünk L_2 nyelvhez olyan $\langle U_2, \varrho_2 \rangle$ interpretációt, hogy $U_2 = \{u, u', u''\}$, továbbá:

$$\varrho(Q) = Q_2 \quad \varrho(S) = S_2 \quad \varrho(T) = T_2$$

α	$Q_2(\alpha)$	$S_2(\alpha)$	$T_2(\alpha)$	$\llbracket \uparrow Q(x) \rrbracket_{v[x:\alpha]}^{\langle U_2, \varrho_2 \rangle}$	$\llbracket \downarrow Q(x) \rrbracket_{v[x:\alpha]}^{\langle U_2, \varrho_2 \rangle}$
u	0	0	1	1	2
u'	1	0	1	1	2
u''	0	1	0	0	0

Ebben az interpretációban tetszőleges v változókiértékelés mellett

$$\llbracket \uparrow \exists x Q(x) \rrbracket_v^{\langle U_2, \varrho_2 \rangle} = 1, \text{ hiszen } \llbracket \uparrow Q(x) \rrbracket_{v[x:u]}^{\langle U_2, \varrho_2 \rangle} = 1, \text{ ugyanakkor}$$

$$\llbracket \downarrow \neg \exists x Q(x) \rrbracket_v^{\langle U_2, \varrho_2 \rangle} = 1, \text{ hiszen } \llbracket \downarrow Q(x) \rrbracket_{v[x:u'']}^{\langle U_2, \varrho_2 \rangle} = 0, \text{ továbbá } \downarrow Q(x) \text{ egyetlen univerzumelem esetében sem igaz.}$$

Így, ha $B \stackrel{\text{def}}{=} \exists x Q(x)$, akkor $\{\uparrow B, \downarrow \neg B\}$ kielégíthető. ■

Megjegyzés. A továbbiakban az L_1 nyelv alatt is már a kvantorokkal kiegészített nyelvet fogjuk érteni, ahol a logikai konstansok megegyeznek az előbbi bizonyításban megadott L_2 nyelv logikai konstansával.

2.2.1. Kvantoros törvények és cáfolatok

Az alfejezetben sorra vesszük néhány klasszikus kvantoros törvény cáfolatát (ismét a mondatfuntorok közti szemantikai különbségre alapozva). A következő három tétel az alábbi állításokat alapozza meg:

$$\{ \uparrow \forall x (A \supset B), \uparrow A \} \not\models_w \downarrow B, \quad (2.3)$$

$$\{ \downarrow \forall x (A \supset B), \downarrow A \} \not\models_w \uparrow B, \quad (2.4)$$

$$\{ \forall x (A \supset B), \exists x A \} \not\models_w \exists x B, \quad (2.5)$$

ahol A és B az L_1 nyelv formulái. Ezt követően hasonló módszerrel igazoljuk, hogy

$$\{ \forall x (A \wedge B) \} \not\models_w (\forall x A \vee \forall x B), \quad (2.6)$$

ehhez azonban egy bővebb (több eszközt tartalmazó) L_3 nyelvre lesz szükség. A nyelv és a kielégíthetőség összefüggéseivel később külön alfejezetben foglalkozunk.

2.20. Tétel. *Van olyan $\{\uparrow \forall x(A \supset B), \uparrow A, \downarrow \neg B\}$ alakú mondathalmaz az L_1 nyelvben, amely kielégíthető.*

Bizonyítás. A bizonyításhoz az L_1 nyelv $\langle U_1, \varrho_1 \rangle$ interpretációját fogjuk használni. Legyen A és B formula a következőképp definiálva:

$$A \stackrel{\text{def}}{=} P(x), \quad B \stackrel{\text{def}}{=} +P(x)$$

A kvantoros részkifejezés értékének meghatározásához foglaljuk táblázatba a lehetséges x szerinti variánsait a v_1 változókiértékelésnek:

α	$\llbracket \uparrow P(x) \rrbracket_{v_1[x:\alpha]}^{\langle U_1, \varrho_1 \rangle}$	$\llbracket \uparrow +P(x) \rrbracket_{v_1[x:\alpha]}^{\langle U_1, \varrho_1 \rangle}$	$\llbracket \uparrow (P(x) \supset +P(x)) \rrbracket_{v_1[x:\alpha]}^{\langle U_1, \varrho_1 \rangle}$
u	1	1	1
u'	1	1	1
u''	2	0	2

Látható, hogy mondathalmazunk minden formulája igaz:

$$\llbracket \uparrow \forall x(A \supset B) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = \llbracket \uparrow \forall x(P(x) \supset +P(x)) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1,$$

$$\llbracket \uparrow A \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = \llbracket \uparrow P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1, \text{ hiszen } v_1(x) = u,$$

$$\llbracket \downarrow \neg B \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = \llbracket \downarrow \neg +P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1.$$

Ezzel igazoltuk a tételt és egyben a (2.3) állítást. ■

2.21. Tétel. *Van olyan $\{\downarrow \forall x(A \supset B), \downarrow A, \uparrow \neg B\}$ alakú mondathalmaz az L_1 nyelvben, amely kielégíthető.*

Bizonyítás. A bizonyításhoz az L_1 nyelv $\langle U_1, \varrho_1 \rangle$ interpretációját használjuk fel, az A és B formulákat pedig a következőképpen definiáljuk:

$$A \stackrel{\text{def}}{=} (+P(x) \supset +P(x)), \quad B \stackrel{\text{def}}{=} \neg +P(x)$$

Ebben az esetben:

$$\llbracket \downarrow A \rrbracket_v^{\langle U_1, \varrho_1 \rangle} = 1 \text{ tetszőleges } v \text{ változókiértékelés esetén, és}$$

$$\llbracket \downarrow B \rrbracket_v^{\langle U_1, \varrho_1 \rangle} = 1 \text{ tetszőleges } v \text{ változókiértékelés esetén, ezért}$$

$$\llbracket \downarrow \forall x(A \supset B) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1, \text{ továbbá}$$

$$\llbracket \uparrow \neg B \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = \llbracket \uparrow \neg \neg +P(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1, \text{ hiszen } v_1(x) = u.$$

Mivel a mondathalmazban szereplő minden mondat igaz, így igazoltuk a tételt és egyben a (2.4) állítást. ■

2.22. Tétel. *Van olyan $\{\forall x(A \supset B), \exists xA, \neg\exists xB\}$ alakú formulahalmaz az L_1 nyelvben, amely kielégíthető.*

Bizonyítás. A bizonyításhoz az L_1 nyelv $\langle U_1, \varrho_1 \rangle$ interpretációját használjuk fel, az A és B formulákat pedig a következőképpen definiáljuk:

$$A \stackrel{def}{=} \neg +T(x), \quad B \stackrel{def}{=} \neg T(x)$$

A kvantoros részkifejezés értékének meghatározásához foglalkozunk táblázatba a v_1 változókiértékelés lehetséges x szerinti variánsait:

α	$\llbracket T(x) \rrbracket_{v_1[x:\alpha]}^{\langle U_1, \varrho_1 \rangle}$	$\llbracket \neg T(x) \rrbracket_{v_1[x:\alpha]}^{\langle U_1, \varrho_1 \rangle}$	$\llbracket +T(x) \rrbracket_{v_1[x:\alpha]}^{\langle U_1, \varrho_1 \rangle}$	$\llbracket \neg +T(x) \rrbracket_{v_1[x:\alpha]}^{\langle U_1, \varrho_1 \rangle}$
u	1	0	1	0
u'	1	0	1	0
u''	2	2	0	1

A táblázatból leolvashatjuk a három eredeti formula igazságértékét:

$$\llbracket \forall x(A \supset B) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = \llbracket \forall x(\neg +T(x) \supset \neg T(x)) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1 \text{ és}$$

$$\llbracket \exists xA \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = \llbracket \exists x\neg +T(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 1,$$

$$\llbracket \exists xB \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = \llbracket \exists x\neg T(x) \rrbracket_{v_1}^{\langle U_1, \varrho_1 \rangle} = 0, \text{ tehát } \llbracket \neg\exists xB \rrbracket_v = 1.$$

Így a formulahalmazunk valóban kielégíthető, és ezzel igazoltuk a (2.5) állítást. ■

2.23. Tétel. *Konstruálható olyan parciális elsőrendű nyelv, amelyben felírható olyan $\{\forall x(A \wedge B), \neg\forall xA, \neg\forall xB\}$ formulahalmaz, amely kielégíthető.*

Bizonyítás. A célunk az, hogy a $\forall xA$ és $\forall xB$ formulák magja legalább egy univerzum-elem esetében hamis legyen úgy, hogy ekkor a másik magformula nem hamis, továbbá legyen legalább egy univerzumelem, amely esetében mindkét magformula egyszerre igaz. Ehhez legalább háromelemű univerzumra van szükség, ahol a magformulák értéke alakulhat például az alábbi táblázat szerint:

α	$\llbracket A \rrbracket_{v_3[x:\alpha]}^{\langle U_3, \varrho_3 \rangle}$	$\llbracket B \rrbracket_{v_3[x:\alpha]}^{\langle U_3, \varrho_3 \rangle}$	$\llbracket (A \wedge B) \rrbracket_{v_3[x:\alpha]}^{\langle U_3, \varrho_3 \rangle}$
u	1	1	1
u'	2	0	2
u''	0	2	2

A kérdés inentől már csak annyi, hogy megkonstruálható-e ilyen A és B formula, valamint interpretáció. Példaként készítünk egy L_3 nyelvet, ahol $\mathfrak{T} = \{T_1, T_2, T_3, T_4\}$ és $\mathcal{P}(1) = \mathfrak{T}$.

α	T_1	T_2	T_3	T_4	$\llbracket T_1(x) \rrbracket_{v_3[x:\alpha]}^{\langle U_3, \varrho_3 \rangle}$	$\llbracket T_2(x) \rrbracket_{v_3[x:\alpha]}^{\langle U_3, \varrho_3 \rangle}$	$\llbracket T_3(x) \rrbracket_{v_3[x:\alpha]}^{\langle U_3, \varrho_3 \rangle}$
u	1	1	1	0	1	1	1
u'	0	1	0	1	0	1	2
u''	0	0	1	1	0	2	1

Az értékrés azért jelenik meg T_2 és T_3 értékeként, mert esetükben nincs olyan eszköz, amely tartalmazná a negativitási tartományuk elemeit, de annál nem többet. A T_1 , T_2 és T_3 predikátumparamétereket felhasználva már összeállíthatjuk a kívánt formulákat:

$$A \stackrel{\text{def}}{=} (T_1(x) \vee \neg T_3(x)), \quad B \stackrel{\text{def}}{=} (T_1(x) \vee \neg T_2(x)).$$

A kívánt formulahalmaz megkonstruálásával igazoltuk a tételt és a (2.6) állítást. ■

Amennyiben az előbb vizsgált $\{\forall x(A \wedge B), \neg \forall x A, \neg \forall x B\}$ formulahalmaz (megfelelőképpen választva a nyelvet és annak A , illetve B formuláit) kielégíthető, úgy az

$$\{ \forall x(A \wedge B) \} \not\models_w (\forall x A \wedge \forall x B) \quad (2.7)$$

állítást is bizonyítottuk. Ez valójában a klasszikus logika egy kétoldali kvantorki-emelési szabályának egyik fele. A következő tételben foglalt állítás így azért lehet meglepő, mert azt is egy kétoldali kvantorkiemelési szabály részeként kaptuk.

2.24. Tétel. *Tetszőleges A és B formulák esetén $\{ \exists x(A \vee B) \} \models_w (\exists x A \vee \exists x B)$.*

Bizonyítás. Indirekt tegyük fel, hogy a tétel nem teljesül, azaz létezik olyan $\langle U, \varrho \rangle$ interpretáció és v változókiértékelés, amely esetén $\llbracket \exists x(A \vee B) \rrbracket_v^{\langle U, \varrho \rangle} = 1$, ugyanakkor $\llbracket (\exists x A \vee \exists x B) \rrbracket_v^{\langle U, \varrho \rangle} = 0$. Ekkor az alábbi megállapításokat tehetjük:

1. $\llbracket \exists x(A \vee B) \rrbracket_v^{\langle U, \varrho \rangle} = 1$ miatt van olyan $u \in U$, hogy $\llbracket (A \vee B) \rrbracket_{v[x:u]}^{\langle U, \varrho \rangle} = 1$;
2. ebből következően $\llbracket A \rrbracket_{v[x:u]}^{\langle U, \varrho \rangle} \neq 2$ és $\llbracket B \rrbracket_{v[x:u]}^{\langle U, \varrho \rangle} \neq 2$;
3. $\llbracket A \rrbracket_{v[x:u]}^{\langle U, \varrho \rangle} \neq 2$ maga után vonja, hogy $\llbracket \exists x A \rrbracket_v^{\langle U, \varrho \rangle} \neq 2$;
4. $\llbracket B \rrbracket_{v[x:u]}^{\langle U, \varrho \rangle} \neq 2$ maga után vonja, hogy $\llbracket \exists x B \rrbracket_v^{\langle U, \varrho \rangle} \neq 2$;
5. szintén $\llbracket (A \vee B) \rrbracket_{v[x:u]}^{\langle U, \varrho \rangle} = 1$ miatt

$$(a) \text{ vagy } \llbracket A \rrbracket_{v[x:u]}^{\langle U, \varrho \rangle} = 1, \text{ de ekkor } \llbracket \exists x A \rrbracket_v^{\langle U, \varrho \rangle} = 1,$$

$$(b) \text{ vagy } \llbracket B \rrbracket_{v[x:u]}^{\langle U, \varrho \rangle} = 1, \text{ de ekkor } \llbracket \exists x B \rrbracket_v^{\langle U, \varrho \rangle} = 1.$$

Mindkét fenti esetben azt kaptuk, hogy a $\llbracket (\exists x A \vee \exists x B) \rrbracket_v^{\langle U, \varrho \rangle}$ formula legalább egyik diszjunkciós tagja igaz úgy, hogy a másik tagja esetében értékrés nem jelenhet meg. Ennél fogva $\llbracket (\exists x A \vee \exists x B) \rrbracket_v^{\langle U, \varrho \rangle} = 1$, ami ellentmondás. ■

Első ránézésre hasonlóan meglepő – és ezért számunkra most érdekes – lehet a következő tétel, melynek formulái nagyon hasonlítanak a (2.5) állításban szereplőkhöz.

2.25. Tétel. *Tetszőleges A és B formulák esetén $\{\forall x(A \supset B), \forall xA\} \models_w \exists xB$.*

Bizonyítás. Ismét indirekt tételezzük fel, hogy van olyan interpretáció és változókiértékelés, amely esetén $\llbracket \forall x(A \supset B) \rrbracket_v^{(U, \varrho)} = 1$ és $\llbracket \forall xA \rrbracket_v^{(U, \varrho)} = 1$, de $\llbracket \exists xB \rrbracket_v^{(U, \varrho)} = 0$. Ekkor az alábbi megállapításokat tehetjük:

1. $\llbracket \forall x(A \supset B) \rrbracket_v^{(U, \varrho)} = 1$ miatt van olyan $u \in U$, hogy $\llbracket (A \supset B) \rrbracket_{v[x:u]}^{(U, \varrho)} = 1$;
2. $\llbracket A \supset B \rrbracket_{v[x:u]}^{(U, \varrho)} \neq 2$ miatt $\llbracket A \rrbracket_{v[x:u]}^{(U, \varrho)} \neq 2$;
3. $\llbracket \forall xA \rrbracket_v^{(U, \varrho)} = 1$ miatt $\llbracket A \rrbracket_{v[x:u]}^{(U, \varrho)} \neq 0$;
4. az előbbi két állításból következően $\llbracket A \rrbracket_{v[x:u]}^{(U, \varrho)} = 1$;
5. $\llbracket A \rrbracket_{v[x:u]}^{(U, \varrho)} = 1$ és $\llbracket (A \supset B) \rrbracket_{v[x:u]}^{(U, \varrho)} = 1$ miatt $\llbracket B \rrbracket_{v[x:u]}^{(U, \varrho)} = 1$;
6. $\llbracket B \rrbracket_{v[x:u]}^{(U, \varrho)} = 1$ maga után vonja, hogy $\llbracket \exists xB \rrbracket_v^{(U, \varrho)} = 1$.

$\llbracket \exists xB \rrbracket_v^{(U, \varrho)} = 1$ ellentmond a $\llbracket \exists xB \rrbracket_v^{(U, \varrho)} = 0$ indirekt feltevésnek. ■

2.2.2. A nyelv és a kielégíthetőség összefüggései

Az előző fejezetben formulahalmazok kielégíthetőségének kérdéseit vizsgáltuk. Több esetben a kielégíthetőség bizonyítására azt a módszert választottuk, hogy megkonstruáltunk egy modellt. A klasszikus elsőrendű logikában is felmerült a kérdés, hogy milyen tulajdonságú interpretációkon érdemes a kielégíthetőségi kérdéseket vizsgálni. Ismeretes például, hogy a $\{P(x), \neg P(y)\}$ formulahalmaz nem kielégíthető olyan interpretációban, amelynek univerzuma egyelemű, de létezik olyan, legalább két univerzumelemmel rendelkező interpretáció, amely a formulahalmaz modellje. Azt, hogy hogyan konstruálhatunk megfelelő számosságú univerzumot, azt Herbrand mutatta meg. Ennek segítségével azt mondhattuk, hogy ha egy formulahalmaz (klózhalmaz) nem rendelkezik Herbrand-moddellel, akkor egészen biztosan kielégíthetetlen.

A klasszikus elsőrendű logikában a kielégíthetőség kérdése független magától az elsőrendű nyelvtől. Ezalatt azt értjük, hogy ha adott egy formulahalmaz valamely elsőrendű logikai nyelven és az kielégíthetetlen, akkor az a formulahalmaz tetszőleges olyan elsőrendű nyelven kielégíthetetlen, amelynek formulahalmaza. Azaz ha a nyelvben vannak olyan szimbólumok, amik nem szerepelnek a formulahalmaz formuláiban, az nem változtatja meg a formulahalmaz szemantikai tulajdonságait. Elképzelhető persze, hogy a bővebb nyelvhez tartozó Herbrand-univerzum nagyobb elemszámú, de ettől még nem találunk Herbrand-modellt a formulahalmazunkhoz. (Természetesen nem célszerű olyan elsőrendű nyelven vizsgálódni, ami a tekintett formulahalmazban szereplő szimbólumoknál többet tartalmaz, de megengedett.)

A közelítő mondatfunktorok használata esetén nem elegendő megfelelő számosságú univerzumon vizsgálódni, a világunkról alkotott képünket ugyanis ez esetben korlátozzák az eszközök. Így fordulhat elő, hogy a $\{\downarrow P(x), \downarrow \neg P(y)\}$ formulahalmaz bizonyos nyelven kielégíthetetlen (még megszámlálhatóan végtelen univerzum esetében is), más nyelven viszont kielégíthető.

2.26. Tétel. Az $L = \langle LC, Var, \{T, P\}, \{T\}, Form, Snt \rangle$ parciális elsőrendű nyelven felírt $\{\downarrow P(x), \downarrow \neg P(y)\}$ mondathalmaz kielégíthetetlen.

Bizonyítás. Legyen $\langle U, \varrho \rangle$ egy tetszőleges interpretáció, v pedig az interpretációhoz tartozó tetszőleges változókiértékelés. Indirekt tegyük fel, hogy

$$\llbracket \downarrow P(x) \rrbracket_v^{\langle U, \varrho \rangle} = 1 \quad \text{és} \quad \llbracket \downarrow \neg P(y) \rrbracket_v^{\langle U, \varrho \rangle} = 1.$$

Ekkor a következő állítások teljesülnek:

$$\llbracket \downarrow P(x) \rrbracket_v^{\langle U, \varrho \rangle} = 1 \text{ miatt } [\varrho(T)]^+ \subseteq [\varrho(P)]^+,$$

$$\text{ha } \varrho(T)(v(y)) = 1, \text{ akkor } \llbracket \downarrow P(y) \rrbracket_v^{\langle U, \varrho \rangle} = 1, \text{ ezért } \llbracket \downarrow \neg P(y) \rrbracket_v^{\langle U, \varrho \rangle} = 0,$$

$$\text{ha } \varrho(T)(v(y)) = 0, \text{ akkor } \llbracket \downarrow P(y) \rrbracket_v^{\langle U, \varrho \rangle} = 2, \text{ hiszen nincs más eszköz a nyelvben.}$$

Ekkor $\llbracket \downarrow \neg P(y) \rrbracket_v^{\langle U, \varrho \rangle} = 2$.

Mivel $\llbracket \downarrow \neg P(y) \rrbracket_v^{\langle U, \varrho \rangle} \neq 1$, így ellentmondásra jutottunk, tehát a formulahalmaz nem kielégíthető. ■

A bizonyítás azon múlt, hogy egyetlen eszköz volt a nyelvben, így legyen akár-mennyi univerzumelem, ezeket csupán két kategóriába sorolhatjuk: az eszközhöz tartozó vagy nem tartozó elemek közé. Az előbbi esetben attól függően, hogy P és T pozitivitási tartománya milyen relációban van egymással, a P predikátumparaméterrel készített atomok alsó közelítésének értéke 0, 1 vagy 2, de minden ilyen univerzumelem esetén ugyanaz. Utóbbi esetben garantáltan értékrés keletkezik.

2.27. Tétel. Legyen az $L = \langle LC, Var, \{T_1, T_2, P\}, \{T_1, T_2\}, Form, Snt \rangle$ parciális elsőrendű nyelven felírt $\{\downarrow P(x), \downarrow \neg P(y)\}$ mondathalmaz kielégíthető.

Bizonyítás. Megfelelő $\langle U, \varrho \rangle$ interpretáció és hozzá tartozó v változókiértékelés az alábbi:

$$U = \{u, u'\},$$

$$\varrho(T_1) = T_1^\varrho, \text{ úgy hogy } T_1^\varrho(u) = 1 \text{ és } T_1^\varrho(u') = 0,$$

$$\varrho(T_2) = T_2^\varrho, \text{ úgy hogy } T_2^\varrho(\alpha) = (1 - T_1^\varrho(\alpha)) \text{ minden } \alpha \in U \text{ esetén,}$$

$$\varrho(P) = T_1^\varrho,$$

$$v(x) = u \text{ és } v(y) = u'.$$

Ekkor $\llbracket \downarrow P(x) \rrbracket_v^{\langle U, \varrho \rangle} = 1$ és $\llbracket \downarrow \neg P(y) \rrbracket_v^{\langle U, \varrho \rangle} = 1$, ami bizonyítja, hogy a mondathalmaz kielégíthető. ■

2.3. Eszközalapú parciális szemantika a gyakorlatban

Annak érdekében, hogy az eszközalapú parciális szemantikát a gyakorlatban is kipróbálhassuk, relációsadatbázis-alapú implementáció készült. A cél az volt, hogy valós adatokon vizsgálhassuk a kialakított rendszer működését, és összehasonlíthassuk a közelítéssel végzett számítások eredményét a klasszikus logikában kapott eredménnyel. Nemcsak a formuláinkat szeretnénk szabadon variálni, hanem lehetőséget akartunk adni az eszközök szabad megválasztására, annak érdekében, hogy egy-egy formulát akár több approximációs teret felhasználva is vizsgálhassunk.

Az implementációhoz a nyelv egyszerűsített változatát vettük alapul, amelynek formális definiálásától most eltekintünk. Az elkészült implementáció csak x változó szerint kvantált prenex formulák kiértékelését teszi lehetővé, amelynek magja konjunktív normálforma, és amelyben az atomok egyargumentumú predikátumparaméterek, ahol a paraméter x . Például:

$$\exists x(P_1(x) \wedge \neg P_2(x)).$$

A kiértékelés alsó és felső közelítéssel, valamint közelítő mondatfunktorok használata nélkül is megtörténik. Az eszközöket a nyelv szavaiban nem használjuk (csak az approximációs tér elkészítésében játszanak szerepet), így a mondatfunktorok nélküli formulák kiértékelésének eredménye egybevág a klasszikus logikában kapott eredménnyel (értékrés nem jelenhet meg).

Egy $\langle U, \varrho \rangle$ interpretáció univerzumát a relációs adatbázis egyetlen táblája alkotja. A tábla minden rekordjának egy-egy univerzumelemet feleltetünk meg. Így egy véges univerzumot kapunk (pontosan annyi univerzumelemmel, ahány rekordja épp van a táblának). A tábla rekordtípusán definiálható logikai függvények részhalmaza alkotja ϱ értékkészletét. Ezeket a logikai függvényeket tárolt függvényként definiáljuk a relációs adatbázis-kezelő rendszerben.

Ha n tárolt logikai függvényt készítettünk a rekordtípuson (jelölje ezeket $p_1^e, \dots, p_n^e$) és ezek közül k darabot használunk fel eszközként, akkor a nyelv nem logikai konstansai a következőképpen alakulnak:

$$Con = \{P_1, \dots, P_n, T_1, \dots, T_k\}, \quad \mathfrak{T} = \{T_1, \dots, T_k\},$$

interpretációjuk pedig

$$\begin{aligned} \varrho(T_i) &= \varrho(P_i) = p_i^e, \quad \text{ahol } i = 1, \dots, k; \\ \varrho(P_j) &= p_j^e, \quad \text{ahol } j = k + 1, \dots, n. \end{aligned}$$

Az eszközöket (a $\mathfrak{T}$ halmaz elemeit) atomi formulák készítésére nem használjuk fel, de minden eszközhöz tartozik egy olyan predikátumparaméter, amelyhez az interpretáció ugyanazt a logikai függvényt rendeli.

Az alkalmazásban az univerzumot és a tárolt logikai függvényeket rögzítettük, a $\mathfrak{T}$ halmaz elemeit pedig szabadon válogathatja össze a felhasználó, meghatározva ezzel az approximációs teret.

2.4. ábra. Logikai függvény PL/pgSQL nyelven definiálva

```
CREATE OR REPLACE FUNCTION pl1(IN data)
  RETURNS boolean
  LANGUAGE plpgsql
AS
$BODY$
  BEGIN
    -- accepted solution written in Java
    RETURN $1.status = 'PASS' AND trim($1.lng) = 'JAVA';
  END;
$BODY$;
```

2.3.1. Az interpretációt adó adattábla

Az adattábla, amely az univerzum elemeit adja, és a tárolt logikai függvények, amelyek az interpretáció alapját képezik, a fejlesztéskor kerültek rögzítésre, de a későbbiekben egyszerűen módosíthatóak vagy cserélhetőek.

Az adatok forrásaként a ProgCont szoftver adatbázisa szolgált. A ProgCont szoftver egy, a Debreceni Egyetem Informatikai Karán működő, programozási feladatokkal és az ezekre érkező megoldások tesztelésével és kiértékelésével foglalkozó rendszer. 2012-es indulása óta működése során a feladatokról, a felhasználókról, a beküldött megoldásokról és azok tesztelésének eredményéről számos adatot gyűjtött össze, amelyek elemzésére több szoftver is készült. Ez az adattömeg alkalmasnak látszott arra, hogy teszteljük az eszközalapú parciális szemantika működését éles körülmények között (valódi adatokon).

Az adatok forrásául szolgáló rendszer kiválasztása után ki kellett alakítani egy olyan adattáblát, amelynek a rekordjai az interpretáció univerzumát adják, attribútumai segítségével pedig változatos logikai függvények definiálhatók. Az interpretáció univerzumának a ProgCont rendszerbe beküldött megoldásokat választottuk, amelynek az adatfeltöltés időpontjában 10 693 rekordja volt. Az attribútumok közé (a beküldött megoldások egyedi azonosítóján kívül) bekerültek a megoldásnak szánt programkódok adatai (a forrásszöveg nyelve, a beküldő személy vagy csapat azonosítója, a forrásszöveg hossza stb.) és a kiértékelésükkel kapcsolatos információk (elfogadásra került-e, vagy milyen hiba miatt került elutasításra), így végeredményül 15 attribútum került meghatározásra.

A logikai függvények az adattábla rekordtípusán értelmezettek, és definíciójukban a tábla attribútumai tetszőlegesen felhasználhatóak. Ezt demonstrálja a 2.4. kódrészlet, amely egy olyan logikai függvényt definiál, amely pontosan akkor igaz, ha a paraméterében megadott megoldás Java programozási nyelven készült, és a tesztelő rendszer elfogadta helyes megoldásként.

Bár a jelenlegi implementációban nem kapott helyet, de a parcialitás akár már ezen logikai függvényekben is megjelenhetett volna. Elképzelhető ugyanis, hogy a logikai függvény definíciójában szereplő attribútumok egyikének aktuális értéke null legyen, és így a logikai függvény igazságértékeként is megjelenhet a null mint harmadik logikai érték. Mivel azonban az implementációban követni szerettük volna a korábban

definiált nyelvet és interpretációt, ezért csak azon rekordokat emeltük át az adattáblába, amelyeknek minden attribútuma definiált (így például csak azon megoldásokkal foglalkoztunk, amelyeket a kiértékelő rendszer már értékelt).

A nyelv definiálásához a predikátumparaméterek nevét egy adatbázistáblában felsoroltuk. Predikátumparaméter csak olyan azonosító lehet, amellyel megegyező nevű logikai függvény szerepel az adatbázisban. Ezzel egyben az interpretáció is rögzítésre kerül. Formálisan azt mondhatjuk, hogy az interpretáció minden predikátumparaméterhez a vele megegyező nevű tárolt függvényt rendeli.

Első ránézésre úgy tűnhet, hogy a nyelv és a szemantika is rögzítésre került azzal, hogy kialakítottuk az adattáblát és a rekordjain definiált függvényeket. Valójában a felhasználó szabadon válogathatja össze, hogy milyen halmazokat használ fel a halmazközelítéshez azáltal, hogy miként választja ki az eszközöket a predikátumparaméterek közül. Szigorúan véve magának a nyelvnek a definíciója is ezzel válik teljessé. Egyetlen korlátozás természetesen, hogy az eszköz csak olyan predikátumparaméter lehet, amelyhez az interpretáció nemüres pozitivitási tartománnyal rendelkező logikai függvényt rendel, biztosítva ezzel az approximációs tér generálásának feltételeit. (Most ez valódi korlátozást nem jelent, mivel az adatbázisban jelenleg definiált tárolt logikai függvények mindegyike ilyen.)

2.3.2. A közelítő szemantika SQL implementációja

Elméletben a formulák szemantikai értékének definiálása előtt meg kell határozni az eszközök és az interpretáció által generált logikailag releváns approximációs teret. A szemantikai definíciókban az eszközökhöz tartozó logikai függvények pozitivitási tartományát használtuk fel a mondatfunktorokkal ellátott formulák értékének meghatározásához. Az SQL implementáció szintjén ez azt jelenti, hogy össze kell gyűjteni azokat az univerzumelemeket (vagy ezek azonosítóit), amelyek az egyes eszközök pozitivitási tartományához tartoznak. Az approximációs tér elkészítése a gyakorlatban azt jelenti, hogy egy relációs adatbázistáblában eszköz (mint predikátumparaméter) és univerzumelem (mint az adattáblabeli azonosító) párokat képzünk. Azért is célszerű külön adattáblában eltárolni a pozitivitási tartományok elemeit, mert így egy-egy univerzumelem esetében elegendő egyszer ténylegesen kiszámítani a megfelelő tárolt logikai függvények értékét. Nem is beszélve arról, hogy az így kapott relációs adattábla indexelhető, ami tovább gyorsítja a feldolgozást.

A közelítő számítások elvégzéséhez felhasználunk két, a pozitivitási tartományok egymáshoz való viszonyát megjelenítő mátrixot. Legyenek a nyelv predikátumparaméterei $P_1, \dots, P_n$ és az eszközök $T_1, \dots, T_k$, ahol $k \leq n$, továbbá $\varrho(P_i) = \varrho(T_i)$ minden $i \in \{1, \dots, k\}$ esetén, továbbá

$$\begin{aligned} A &= (a_{i,j})_{k \times n} \quad \text{és} \quad B = (b_{i,j})_{k \times n}, \\ a_{i,j} &= \left| [\varrho(T_i)]^+ \cap [\varrho(P_j)]^+ \right|, \\ b_{i,j} &= \left| [\varrho(T_i)]^+ \setminus [\varrho(P_j)]^+ \right|. \end{aligned}$$

ahol $i = 1, \dots, k$ és $j = 1, \dots, n$. Tulajdonképpen elegendő volna annyit rögzíteni, hogy a fent definiált metszet és különbség üres halmaz-e. Az elemek számossága későbbi fuzzy rendszerek szemantikájának tesztelése miatt került kiválasztásra.

Az A és B mátrixokat felhasználva a P_j predikátumaink és T_i eszköz kapcsán a következőket mondhatjuk:

- ha $b_{i,j} = 0$, akkor $a_{i,j} > 0$ (hiszen az eszközök pozitivitási tartománya alkotta alaphalmazok nem lehetnek üresek), és $[\varrho(T_i)]^+$ elemei benne foglaltatnak $[\varrho(P_j)]^+$ alsó közelítésében, hiszen ekkor $[\varrho(T_i)]^+ \subseteq [\varrho(P_j)]^+$;
- ha $a_{i,j} > 0$ és $b_{i,j} > 0$, akkor $[\varrho(T_i)]^+$ elemei benne foglaltatnak $[\varrho(P_j)]^+$ felső közelítésében, hiszen ekkor $[\varrho(T_i)]^+ \cap [\varrho(P_j)]^+ \neq \emptyset$;
- ha $a_{i,j} = 0$, akkor $b_{i,j} > 0$ (hiszen az eszközök pozitivitási tartománya alkotta alaphalmazok nem lehetnek üresek), és $[\varrho(T_i)]^+$ elemei akkor elemei P_j negatív tartományának, ha a következő két állítás közül legalább egy teljesül:
 - az eszközök pozitivitási tartománya páronként diszjunkt, vagy
 - nincs olyan $\ell \in \{1, \dots, k\} \setminus \{i\}$ eszköz, hogy $a_{\ell,i} > 0$ és $a_{\ell,j} > 0$ (azaz a i . eszköz a j . predikátum felső közelítésébe sem lóg bele).

Az A és B mátrix értékei az adatbázis egy, a predikátumok és eszközök között felépített kapcsolótáblájában kerülnek rögzítésre, amint meghatározásra került az approximációs tér. (Az adatbázistábla szerkezetét a 2.6. ábrán mutatjuk be.) A kapcsolótábla feltöltése időigényes lehet, de egyszeri feladat. Az eszközök pozitivitási tartománya és a mátrixok segítségével az atomi formulák alsó és felső közelítésének értéke meghatározható anélkül, hogy a tárolt logikai függvények kiértékelésére szükség volna. Formálisan:

$$\left(\llbracket P_j^\downarrow \rrbracket_v^{\langle U, \varrho \rangle} \right) (u) = \begin{cases} 1 & \text{ha van olyan } i, \text{ hogy } u \in [\varrho(T_i)]^+ \text{ és } b_{i,j} = 0; \\ 0 & \text{ha van olyan } i, \text{ hogy } u \in [\varrho(T_i)]^+ \text{ és } a_{i,j} = 0, \\ & \text{és egyetlen } \ell \text{ esetén sem teljesül } a_{\ell,i} > 0 \text{ és } a_{\ell,j} > 0; \\ 2 & \text{egyébként.} \end{cases}$$

$$\left(\llbracket P_j^\uparrow \rrbracket_v^{\langle U, \varrho \rangle} \right) (u) = \begin{cases} 1 & \text{ha van olyan } i, \text{ hogy } u \in [\varrho(T_i)]^+ \text{ és } a_{i,j} > 0; \\ 0 & \text{ha van olyan } i, \text{ hogy } u \in [\varrho(T_i)]^+ \text{ és } a_{i,j} = 0, \\ & \text{és egyetlen } \ell \text{ esetén sem teljesül } a_{\ell,i} > 0 \text{ és } a_{\ell,j} > 0; \\ 2 & \text{egyébként.} \end{cases}$$

ahol $i, \ell \in \{1, \dots, k\}$ és $j \in \{1, \dots, n\}$.

Ahogy azt a 2.5. ábra is mutatja, a két segédtáblát felhasználva elég egyszerűen alakíthatjuk át a fenti függvényeket PL/pgSQL utasításokká. A bemutatott kódrészlet a `p_id` elsődleges kulccsal rendelkező univerzumelem helyen határozza meg a `p_pred` predikátumparaméterhez tartozó logikai függvény alsó közelítésének értékét a `p_session_id` azonosítóval megjelölt approximációs térben. A két segédtábla `tooldata` és `mtx`.

A következő példában a P_1, P_2, P_3 predikátumparaméterek segítségével generált approximációs térben szeretnénk kiszámítani a

$$\forall x (P_4(x) \vee \neg P_5(x))$$

formula értékét. A predikátumparaméterek interpretációja az implementációban a következő: a predikátumszimbólumhoz rendelt tárolt logikai függvény pontosan azon rekordok esetében igaz, ahol

P_1 : a megoldást egy csapat küldte be,

P_2 : a beküldött megoldás **C#** nyelven íródott, de szintaktikai hibás,

P_3 : a beküldött megoldás **C#** nyelven íródott és futási hibát produkált,

P_4 : a beküldött megoldás **Java** nyelven íródott és futási hibát produkált,

P_5 : a beküldött megoldás **C++** nyelven íródott, de szintaktikai hibás.

Az approximációs térhez tartozó A és B mátrixok értékei a következők:

$$A = \begin{pmatrix} 5402 & 8 & 757 & 126 & 159 \\ 8 & 28 & 0 & 0 & 0 \\ 757 & 0 & 1084 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 5394 & 4645 & 5276 & 5243 \\ 20 & 0 & 28 & 28 & 28 \\ 327 & 1084 & 0 & 1084 & 1084 \end{pmatrix}$$

Az, hogy az egyes eszközök pozitivitási tartománya hány elemű, $a_{i,i}$ értékéből leolvasható, innen tudhatjuk, hogy az approximációs tér az univerzum több mint felét lefedi. Ennek ellenére a formulánk közelítő és valódi igazságértékei különböznek, mivel nem jól választottuk meg az eszközöket.

$$\begin{aligned} \llbracket \uparrow \forall x (P_4(x) \vee \neg P_5(x)) \rrbracket_v^{(U,\varrho)} &= 1 \\ \llbracket \downarrow \forall x (P_4(x) \vee \neg P_5(x)) \rrbracket_v^{(U,\varrho)} &= 2 \\ \llbracket \forall x (P_4(x) \vee \neg P_5(x)) \rrbracket_v^{(U,\varrho)} &= 0 \end{aligned}$$

A felső közelítésnél a pontatlanság abból adódik, hogy a P_1 eszköz pozitivitási tartománya minden predikátum pozitivitási tartományába belelóg (van nemüres metszetük, de nem részhalmazai egymásnak), ezért a negativitási tartományok eltűnnek. Az alsó közelítésnél az értékrés megjelenését az okozza, hogy $[\varrho(P_4)]^+ = \emptyset$ (ami onnan is látható, hogy $b_{1,4}$, $b_{2,4}$, $b_{3,4}$ közül egyik sem 0).

2.4. Szillogizmusok érvényessége

A következőkben az eszközalapú parciális szemantikát Arisztotelész szillogizmusain keresztül vizsgáljuk meg. A szillogizmusokra azért esett a választás, mert úgy gondoltuk, ezek reprezentálják a leggyakoribb kategorikus következtetéseket, amelyek egyrétű kvantifikáció segítségével formalizálhatóak, maga az alapötlet pedig [22] mintájára született. A vizsgálat során abból a naiv feltételezésből indulunk ki, hogy az approximációs térben az alsó közelítés a biztos tudásunkat testesíti meg, így érdemes

2.5. ábra. Atomi formula igazságértéke pl/pgSQL nyelven

```
-- INPUT: p_session_id, p_id (object identifier), p_pred (predicate)
IF EXISTS (
    SELECT 1
    FROM tooldata t
    JOIN mtx m ON t.name = m.tool
    WHERE t.session_id = p_session_id
    AND m.session_id = p_session_id
    AND t.id = p_id
    AND m.rout = 0
    AND m.pred = p_pred )
THEN
    RETURN 1; -- OUTPUT: the lower approximation is (1) true
ELSIF EXISTS (
    SELECT 1
    FROM tooldata t
    JOIN mtx m ON t.name = m.tool
    WHERE t.session_id = p_session_id
    AND m.session_id = p_session_id
    AND NOT EXISTS (
        SELECT 1
        FROM mtx m1
        JOIN mtx m2 ON m1.tool = m2.tool
        WHERE m1.pred = m.tool AND m1.rin > 0
        AND m1.session_id = p_session_id
        AND m2.pred = p_pred AND m2.rin > 0
        AND m2.session_id = p_session_id )
    AND t.id = p_id
    AND m.rin = 0
    AND m.pred = p_pred )
THEN
    RETURN 0; -- OUTPUT: the lower approximation is (0) false
ELSE
    RETURN 2; -- OUTPUT: the lower approximation is (2) null
END IF;
```

2.6. ábra. A fontosabb táblák létrehozásának pl/pgSQL parancsai

```
CREATE TABLE tooldata (
    session_id integer NOT NULL, -- session identifier
    name        varchar(10) NOT NULL, -- predicate parameter (tool)
    id          bigint NOT NULL, -- object identifier
    CONSTRAINT pk_tooldata PRIMARY KEY (session_id, name, id) );

CREATE TABLE mtx (
    session_id integer NOT NULL, -- session identifier
    pred        varchar(10) NOT NULL, -- predicate parameter
    tool        varchar(10) NOT NULL, -- predicate parameter (tool)
    rin         integer NOT NULL, -- cardinality of the intersection
    rout        integer NOT NULL, -- cardinality of the difference
    CONSTRAINT pk_mtx PRIMARY KEY (session_id, pred, tool) );
```

megvizsgálni, hogy ebből a tudásból helyes következtetéseket vonhatunk-e le. Mivel a szillogizmusok a klasszikus elsőrendő logikában helyes következtetések, következő vizsgálatunk tárgya az érvényességük tesztelése lesz az approximációs téren. A korábbi eredmények arra engednek következtetni, hogy az értékrés megjelenése miatt az, amit a klasszikus logikában érvényesnek tudtunk, nem feltétlen teljesül majd a parciális logikai rendszerben. Ugyanakkor számos tulajdonság szerint kategorizálhatjuk az approximációs tereket, így nem egyszerűen egy-egy cáfolatot szeretnénk konstruálni a szillogizmusokhoz, hanem ezen kategóriákat külön-külön vizsgáljuk meg. Az alfejezetben ismertetett eredmények rövidített formában [12]-ben kerültek publikálásra.

2.4.1. Arisztotelész kategorikus szillogizmusai

Arisztotelész kategorikus szillogizmusai helyes következtetések, amelyek két premisszából és egy konklúzióból állnak, amelyeket összesen három különböző predikátum segítségével építünk fel. A premisszákat és a konklúziót négy kategóriába szokás sorolni aszerint, hogy miként formalizálhatók: [33]

- *a* típusú: $\forall x(P_1(x) \supset P_2(x))$,
- *i* típusú: $\exists x(P_1(x) \wedge P_2(x))$,
- *e* típusú: $\neg \exists x(P_1(x) \wedge P_2(x))$,
- *o* típusú: $\exists x(P_1(x) \wedge \neg P_2(x))$.

ahol P_1 és P_2 a három predikátumparaméter közül választott és egymástól különböző, úgy hogy minden predikátumparaméter a következtetést alkotó három állítás közül pontosan kettőben szerepel. A három predikátumparaméter szokásos jelölése: P, S, M .

1. alakzat	2. alakzat	3. alakzat	
$M - P$	$P - M$	$M - P$	premissza
$S - M$	$S - M$	$M - S$	premissza
$\frac{S - P}{S - P}$	$\frac{S - P}{S - P}$	$\frac{S - P}{S - P}$	konklúzió

Vizsgálatunk tárgyát az első három alakzat 14 darab szillogizmusa képezi:

- 1. alakzat: *Barbara, Celarent, Darii, Ferio*
- 2. alakzat: *Cesare, Camestres, Festion, Baroco*
- 3. alakzat: *Darapti, Felapton, Disamnis, Datisi, Bocardo, Ferison*

Vegyük például az első kép *Barbara* névvel illetett szillogizmusát, amely csak *a* típusú állításokból épül fel:

$$\underbrace{\forall x(M(x) \supset P(x))}_{1. \text{ premissza}}, \underbrace{\forall x(S(x) \supset M(x))}_{2. \text{ premissza}} \models \underbrace{\forall x(S(x) \supset P(x))}_{\text{konklúzió}}$$

A klasszikus logikában mind a 14 szillogizmus érvényes (*Darapti* és *Felapton* esetében egyetlen egzisztenciális előfeltétel szükséges, amelyet tekinthetünk harmadik premisszának).

2.1. táblázat. Arisztotelész kategorikus szillogizmusai – 1. alakzat

	1. premissza	2. premissza	konklúzió
<i>Barbara</i>	a típusú M, P $\forall x(M(x) \supset P(x))$	a típusú S, M $\forall x(S(x) \supset M(x))$	a típusú S, P $\forall x(S(x) \supset P(x))$
<i>Celarent</i>	e típusú M, P $\forall x(M(x) \supset \neg P(x))$	a típusú S, M $\forall x(S(x) \supset M(x))$	e típusú S, P $\neg \forall x(S(x) \supset P(x))$
<i>Darii</i>	a típusú M, P $\forall x(M(x) \supset P(x))$	i típusú S, M $\exists x(S(x) \wedge M(x))$	i típusú S, P $\neg \exists x(S(x) \wedge \neg P(x))$
<i>Ferio</i>	e típusú M, P $\forall x(M(x) \supset \neg P(x))$	i típusú S, M $\exists x(M(x) \wedge S(x))$	o típusú S, P $\neg \exists x(S(x) \wedge \neg P(x))$

2.4.2. A szillogizmusok formalizálása

A következőkben a szillogizmusok cáfolatainak elkészítése a cél különféle approximációs terekben úgy, hogy a predikátumok értékét alulról közelítjük.

Egyrétegű elsőrendű nyelv

A szillogizmusok formalizálásához elegendő egy olyan nyelv, amelyben csupán egy-argumentumú predikátumparaméterek szerepelnek, a formulák zártak és egyszeresen kvantáltak. (Olyan részhalmazát definiáltuk így az elsőrendű logikai nyelvnek, amely eldönthető.)

2.28. Definíció. Legyen $\langle LC, Var, Con, \mathfrak{T}, Form \rangle$ egy egyszerűsített egyrétegű logikai nyelv, ahol

- $LC = \{\neg, \wedge, \supset, \exists, \forall, (,)\}$ a szillogizmusok formalizálásához szükséges logikai konstansok halmaza;
- $Var = \{x\}$ (az egyrétegű nyelvben egyetlen változó elegendő);
- $Con = \{P_1, P_2, \dots, P_n\}$ ($n \geq 1$) a nem logikai konstansok halmaza, amely kizárólag egy argumentummal rendelkező predikátumparamétereket tartalmaz;
- $\mathfrak{T} = \{T_1, T_2, \dots, T_k\}$ eszközök halmaza, ahol $k \geq 1$.

A nyelv formuláit a következőképpen definiáljuk:

1. Legyen QF a kvantormentes kifejezések halmaza, úgy hogy

- (a) $P(x) \in QF$ egy kvantormentes atomi kifejezés, ha $P \in Con$,
- (b) $\neg A \in QF$, ha $A \in QF$,
- (c) $(A \wedge B) \in QF$ és $(A \supset B) \in QF$, ha $A \in QF$ és $B \in QF$.

2.2. táblázat. Arisztotelész kategorikus szillogizmusai – 2. alakzat

	1. premissza	2. premissza	konklúzió
<i>Cesare</i>	e típusú P, M $\forall x(P(x) \supset \neg M(x))$	a típusú S, M $\forall x(S(x) \supset M(x))$	e típusú S, P $\neg \forall x(S(x) \supset \neg P(x))$
<i>Camestres</i>	a típusú P, M $\forall x(P(x) \supset M(x))$	e típusú S, M $\forall x(S(x) \supset \neg M(x))$	e típusú S, P $\neg \forall x(S(x) \supset \neg P(x))$
<i>Festion</i>	e típusú P, M $\forall x(P(x) \supset \neg M(x))$	i típusú S, M $\exists x(S(x) \wedge M(x))$	o típusú S, P $\neg \exists x(S(x) \wedge \neg P(x))$
<i>Baroco</i>	a típusú P, M $\forall x(P(x) \supset M(x))$	o típusú S, M $\exists x(S(x) \wedge \neg M(x))$	o típusú S, P $\neg \exists x(S(x) \wedge \neg P(x))$

2.3. táblázat. Arisztotelész kategorikus szillogizmusai – 3. alakzat

	1. premissza	2. premissza	konklúzió
<i>Darapti*</i>	a típusú M, P $\forall x(M(x) \supset P(x))$	a típusú M, S $\forall x(M(x) \supset S(x))$	i típusú S, P $\neg \exists x(S(x) \wedge P(x))$
<i>Felapton*</i>	e típusú M, P $\forall x(M(x) \supset \neg P(x))$	a típusú M, S $\forall x(M(x) \supset S(x))$	o típusú S, P $\neg \exists x(S(x) \wedge \neg P(x))$
<i>Disamnis</i>	i típusú M, P $\exists x(M(x) \wedge P(x))$	a típusú M, S $\forall x(M(x) \supset S(x))$	i típusú S, P $\neg \exists x(S(x) \wedge P(x))$
<i>Datisi</i>	a típusú M, P $\forall x(M(x) \supset P(x))$	i típusú M, S $\exists x(M(x) \wedge S(x))$	i típusú S, P $\neg \exists x(S(x) \wedge P(x))$
<i>Bocardo</i>	o típusú M, P $\exists x(M(x) \wedge \neg P(x))$	a típusú M, S $\forall x(M(x) \supset S(x))$	o típusú S, P $\neg \exists x(S(x) \wedge \neg P(x))$
<i>Ferison</i>	e típusú M, P $\neg \exists x(M(x) \wedge P(x))$	i típusú M, S $\exists x(M(x) \wedge S(x))$	o típusú S, P $\neg \exists x(S(x) \wedge \neg P(x))$

* egzisztenciális előfeltétel szükséges: $\exists x M(x)$.

2. A nyelv formuláinak $Form$ halmazát a következő induktív definícióval határozzuk meg:

- (a) $\forall xA \in Form$ és $\exists xA \in Form$, ha $A \in QF$,
- (b) $\neg A \in Form$, ha $A \in Form$,
- (c) $(A \wedge B) \in Form$ és $(A \supset B) \in Form$, ha $A \in Form$ és $B \in Form$.

A $\vee$ logikai konstans csupán azért hiányzik a nyelvből, mert nem szükséges a szillogizmusok formalizálásához. Ez esetben a mondatfunktorok is hiányoznak a nyelvből, mivel – ahogy ez a szemantikánál definiálásra kerül – most csak az alsó közelítésre koncentrálunk.

Interpretáció és approximációs tér

Az $\langle U, \varrho \rangle$ pár egy interpretációja az $\langle LC, Var, Con, Tool, Form \rangle$ nyelvnek, ha U egy nemüres halmaz, $\varrho : Con \cup \mathfrak{T} \rightarrow \{0, 1\}^U$ leképezés, és $\varrho(T) \neq \emptyset$, ha $T \in \mathfrak{T}$.

Az $\langle U, \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{l}, u \rangle$ approximációs tér az interpretáció által egyértelműen generált Pawlak-féle approximációs tér, ahol – az előzőekben megadott módon – az eszközök pozitivitási tartományai határozzák meg az alaphalmazokat.

A kialakítható különböző approximációs terek száma

$$2^{(2^{|U|}-1)} - 1,$$

hiszen U összes nemüres részhalmaza alkalmas arra, hogy egy-egy eszközhöz rendeljük, és ezek közül tetszőlegesen sokat (de legalább egyet) fel is kell használni erre. A szillogizmusok formalizálásához szükséges $Con = \{P, S, M\}$ predikátumparaméterek

$$(2^{|U|})^3$$

különböző módon interpretálhatóak. Egy mindössze 4 elemű univerzumon ez összesen több mint 134 millió lehetőséget jelent.

Szemantika

A kvantortmentes – QF -beli – kifejezések értéke adott $\langle U, \varrho \rangle$ interpretációban az x változóhoz $u \in U$ univerzumelemet rendelve (amit $x \mapsto u$ alakban jelölünk) rekurzívan a következőképpen definiálható:

1. Ha $P(x) \in QF$, akkor $\llbracket P \rrbracket^{\langle U, \varrho \rangle} = \varphi$ egy olyan függvény, hogy

$$\varphi(u) \stackrel{def}{=} \begin{cases} 1 & \text{ha } u \in \mathfrak{l}([\varrho(P)]^+), \\ 0 & \text{ha } u \in \mathfrak{l}(U \setminus [\mathfrak{u}(\varrho(P))]^+), \\ 2 & \text{egyébként} \end{cases} \quad (2.8)$$

$$\llbracket P(x) \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} \stackrel{def}{=} \varphi(u)$$

2.4. táblázat. Kleene gyenge konnektívumai ([19] 2438. oldal)

$\neg$		$\supset$	0	1	2	$\wedge$	0	1	2	$\vee$	0	1	2
0	1	0	1	1	2	0	0	0	2	0	0	1	2
1	0	1	0	1	2	1	0	1	2	1	1	1	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2

2.5. táblázat. Kleene erős konnektívumai ([19] 2438. oldal)

$\neg$		$\supset$	0	1	2	$\wedge$	0	1	2	$\vee$	0	1	2
0	1	0	1	1	1	0	0	0	0	0	0	1	2
1	0	1	0	1	2	1	0	1	2	1	1	1	1
2	2	2	2	1	2	2	0	2	2	2	2	1	2

2. Ha $A \in QF$ és $B \in QF$ tetszőleges kvantormentes kifejezés, akkor

$$\begin{aligned} \llbracket \neg A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} &\stackrel{def}{=} \neg \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle}, \\ \llbracket (A \supset B) \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} &\stackrel{def}{=} \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} \supset \llbracket B \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle}, \\ \llbracket (A \wedge B) \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} &\stackrel{def}{=} \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} \wedge \llbracket B \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle}, \end{aligned}$$

ahol az ítéletlogikai konstansokhoz ez esetben is Kleene gyenge konnektívumait rendeljük (a korábban definiált szemantikának megfelelően).

A $Form$ halmaz formuláinak értéke adott $\langle U, \varrho \rangle$ interpretációban szintén rekurzívan számítható az alábbiak szerint (ahol arra való tekintettel, hogy a formulák zártak, helyettesítést nem kell megadni):

1. Ha $A \in Form$ és $x \in Var$, akkor

$$\begin{aligned} \llbracket \forall x A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} &\stackrel{def}{=} \begin{cases} 2 & \text{ha minden } u \in U \text{ esetén } \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 2, \\ 0 & \text{ha van olyan } u \in U, \text{ hogy } \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 0, \\ 1 & \text{egyébként;} \end{cases} \\ \llbracket \exists x A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} &\stackrel{def}{=} \begin{cases} 2 & \text{ha minden } u \in U \text{ esetén } \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 2, \\ 1 & \text{ha van olyan } u \in U, \text{ hogy } \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 1, \\ 0 & \text{egyébként.} \end{cases} \end{aligned}$$

2. Ha $A \in Form$ és $B \in Form$ tetszőleges formulák, akkor

$$\begin{aligned} \llbracket \neg A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} &\stackrel{def}{=} \neg \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle}; \\ \llbracket (A \supset B) \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} &\stackrel{def}{=} \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} \supset \llbracket B \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle}; \\ \llbracket (A \wedge B) \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} &\stackrel{def}{=} \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} \wedge \llbracket B \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle}. \end{aligned}$$

Az ítéletlogikai konstansokhoz ugyanazokat a logikai műveleteket rendeltük, a különbség pusztán annyi, hogy a formulák értékének meghatározásához nincs szükség változókiértékelésre (x értékére), mivel a $Form$ halmaz elemei zártak.

2.4.3. A szillogizmusok cáfolatainak keresése

Arra való tekintettel, hogy a szillogizmusokat formalizáló kifejezéseink logikai összetettsége nem túl magas, remény van arra, hogy viszonylag kis elemszámú univerzumon megkonstruálhatóak a cáfolatok. Az ötlet lényege, hogy kis méretű univerzumot feltételezve a formulák kiértékelése felgyorsítható.

Legyen az $\langle U, \varrho \rangle$ interpretációhoz tartozó univerzum $U = \{u_1, u_2, \dots, u_n\}$. Készítsünk olyan

$$\text{pos}_{\langle U, \varrho \rangle} : QF \rightarrow 2^U \quad \text{és} \quad \text{neg}_{\langle U, \varrho \rangle} : QF \rightarrow 2^U$$

függvényeket, amelyek segítségével kiterjesztjük a pozitivitási és negativitási tartomány fogalmát minden kvantormentes formulára:

- Ha $P(x) \in QF$ atomi formula, akkor

$$\begin{aligned} \text{pos}_{\langle U, \varrho \rangle}(P(x)) &\stackrel{\text{def}}{=} \left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^+ = \left\{ u_i \in U : \llbracket P(x) \rrbracket_{x \mapsto u_i}^{\langle U, \varrho \rangle} = 1 \right\}, \\ \text{neg}_{\langle U, \varrho \rangle}(P(x)) &\stackrel{\text{def}}{=} \left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^- = \left\{ u_i \in U : \llbracket P(x) \rrbracket_{x \mapsto u_i}^{\langle U, \varrho \rangle} = 0 \right\}. \end{aligned}$$

- Ha $A \in QF$ és $B \in QF$, akkor

$$\begin{aligned} \text{pos}_{\langle U, \varrho \rangle}(\neg A) &= \text{neg}_{\langle U, \varrho \rangle}(A), \\ \text{neg}_{\langle U, \varrho \rangle}(\neg A) &= \text{pos}_{\langle U, \varrho \rangle}(A), \\ \text{pos}_{\langle U, \varrho \rangle}((A \wedge B)) &= \text{pos}_{\langle U, \varrho \rangle}(A) \cap \text{pos}_{\langle U, \varrho \rangle}(B), \\ \text{neg}_{\langle U, \varrho \rangle}((A \wedge B)) &= \left(\text{neg}_{\langle U, \varrho \rangle}(A) \cap \left(\text{pos}_{\langle U, \varrho \rangle}(B) \cup \text{neg}_{\langle U, \varrho \rangle}(B) \right) \right) \cup \\ &\quad \cup \left(\text{neg}_{\langle U, \varrho \rangle}(B) \cap \left(\text{pos}_{\langle U, \varrho \rangle}(A) \cup \text{neg}_{\langle U, \varrho \rangle}(A) \right) \right), \\ \text{pos}_{\langle U, \varrho \rangle}((A \supset B)) &= \left(\text{neg}_{\langle U, \varrho \rangle}(A) \cap \left(\text{pos}_{\langle U, \varrho \rangle}(B) \cup \text{neg}_{\langle U, \varrho \rangle}(B) \right) \right) \cup \\ &\quad \cup \left(\text{pos}_{\langle U, \varrho \rangle}(B) \cap \left(\text{pos}_{\langle U, \varrho \rangle}(A) \cup \text{neg}_{\langle U, \varrho \rangle}(A) \right) \right), \\ \text{neg}_{\langle U, \varrho \rangle}((A \supset B)) &= \text{pos}_{\langle U, \varrho \rangle}(A) \cap \text{neg}_{\langle U, \varrho \rangle}(B). \end{aligned}$$

A halmazműveletek segítségével a kvantormentes formulák értékét az összes univerzumelem esetében egyszerre tudjuk meghatározni. Ráadásul a kvantoros kifejezések

értéke az előbb definiált halmazpár segítségével az alábbiak szerint adódik:

$$\begin{aligned} \llbracket \forall x A \rrbracket^{(U, \varrho)} &= \begin{cases} 0 & \text{ha } \text{neg}_{(U, \varrho)}(A) \neq \emptyset; \\ 1 & \text{ha } \text{neg}_{(U, \varrho)}(A) = \emptyset \text{ és } \text{pos}_{(U, \varrho)}(A) \neq \emptyset; \\ 2 & \text{egyébként;} \end{cases} \\ \llbracket \exists x A \rrbracket^{(U, \varrho)} &= \begin{cases} 1 & \text{ha } \text{pos}_{(U, \varrho)}(A) \neq \emptyset; \\ 0 & \text{ha } \text{pos}_{(U, \varrho)}(A) = \emptyset \text{ és } \text{neg}_{(U, \varrho)}(A) \neq \emptyset; \\ 2 & \text{egyébként.} \end{cases} \end{aligned}$$

Ezt a tulajdonságot (kis méretű univerzumok esetében) abban az algoritmusban is ki fogjuk használni, amely segítségével a formuláink adott interpretációban történő kiértékelését végezzük.

Ahhoz, hogy eldöntsük, hogy egy szillogizmus az approximációs térre épített rendszerben is érvényes-e, azaz például *Barbara* esetében

$$\forall x(M(x) \supset P(x)), \forall x(S(x) \supset M(x)) \models \forall x(S(x) \supset P(x))$$

fennáll-e, elegendő olyan interpretációt keresni, ahol a

$$\forall x(M(x) \supset P(x)) \wedge \forall x(S(x) \supset M(x)) \wedge \neg \forall x(S(x) \supset P(x))$$

formula igaz. Ha sikerrel járunk – találunk ilyen tulajdonsággal bíró interpretációt –, azzal megkonstruáltuk a szillogizmus cáfolatát. A korábbi tapasztalatok alapján a várakozásunk az volt, hogy sok esetben cáfolhatók a szillogizmusok. Annak érdekében, hogy leszűkítsük a vizsgálandó szillogizmusok körét, érdemesnek tűnt kiszűrni azokat, amelyek már viszonylag kis elemszámú univerzumon is cáfolhatók. Ezért olyan alkalmazást készítettünk, amely adott univerzumméretig minden lehetséges interpretációban és approximációs téren kiértékeli a szillogizmusokhoz rendelt formulákat.

Az interpretációban felhasználtuk a formulák értékének meghatározásához a pozitivitási és negativitási tartományokról mint halmazokról összegyűjtött tulajdonságokat. A számításokat a halmazokon végeztük el, egy-egy halmaz pedig egy egész számként (mint bitvektor) került ábrázolásra. 32 bites architektúrát feltételezve így a processzor egy lépésben egyszerre legfeljebb 32 univerzumelemmel dolgozott a kvantormentes részkifejezések kiértékelésekor úgy, hogy bitműveletekkel helyettesítettük az előbb definiált halmazműveleteket. Ennek köszönhetően 4 elemű univerzumot feltételezve – ami az interpretációk és approximációs terek vonatkozásában több mint 134 millió különböző esetet foglal magába – az összes szillogizmus vizsgálata nem került többé pár percnél.

A 2.7. ábra (47. oldal) a Java programnyelven készített implementációt illusztrálja. Ha már adott egy interpretáció (és az általa generált approximációs tér), akkor első lépésben a $\llbracket \downarrow P \rrbracket^+$ és $\llbracket \downarrow P \rrbracket^-$ halmazok kerülnek kiszámításra. Ezeket a halmazokat rendre a **ptd** és **pfd** egészeket tartalmazó tömbök reprezentálják. A tömb egy-egy eleme egy olyan bitvektor, amely az univerzum megfelelő részhalmazát ábrázolja, és a tömbnek annyi eleme van, ahány predikátumparamétert használunk. (A szillogizmusok esetében ez 3, ugyanakkor az alkalmazást úgy alakítottuk ki, hogy az angol ábécé összes kisbetűje használható legyen.) A szillogizmusokat postfix kifejezések

2.7. ábra. A szemantika Java nyelvű implementációja

```
int mask = -1; // each bit position contains 1

switch (next) { // next char of postfix formula
case 'A': // forall
    sp--; // pop an argument
    afd = sfd[sp];
    atd = std[sp];
    // push the result
    sfd[sp] = (afd & mask) != 0 ? mask : 0;
    std[sp] = (atd & mask) != 0
        && (afd & mask) == 0 ? mask : 0;
    sp++;
    break;

case '&': // conjunction
    sp--; // pop an argument
    int rfd = sfd[sp];
    int rtd = std[sp];
    sp--; // pop an argument
    int lfd = sfd[sp];
    int ltd = std[sp];
    // push the result
    sfd[sp] = (lfd & (rfd|rtd))
        | (rfd & (lfd|ltd));
    std[sp] = ltd & rtd;
    sp++;
    break;

// ... other connectives ...

default: // an atom
    sfd[sp] = pfd[next-'a'];
    std[sp] = ptd[next-'a'];
    sp++;
    break;
}
```

formájában rögzítettük egy-egy sztring literálban, megtartva a predikátumparaméterek neveit, és rendre $>$, $\&$, $-$ és A karakterekkel helyettesítve az $\supset$, $\wedge$, $\neg$ és $\forall$ logikai konstansokat. Például *Barbara* az alábbi formában jelent meg:

$$BARBARA = \text{"mp>Asm>A\&sp>A-\&"}$$

A postfix kifejezés kiértékelésére az algoritmus egy egész értékekből álló verempárt használt, elkülönítve kezelve a pozitivitási (*std*) és a negativitási (*sfd*) tartományokat. A példaként bemutatott kódrészletben a verem tetejét az *sp* változó tárolja. Az ítéletlogikai összekötő jelek implementálása során a formulák és a kvantormentes kifejezések között nem kellett különbséget tenni (az előbbi esetben mindössze annyi történt, hogy a bitvektor minden bitpozícióján azonos érték szerepelt).

2.4.4. Eredmények

Az elvégzett vizsgálatok eredményét a 2.6. és 2.7. táblázatokban foglaltuk össze. A táblázatokat lényegesen leegyszerűsítettük, mivel minden szillogizmus besorolható volt 4 kategóriába. Ugyanannyi interpretáció cáfolta a *Barbara* szillogizmust, mint a *Baroco* és *Bocardo* szillogizmusokat, továbbá egyforma eredményeket kaptunk *Celarent*, *Darii*, *Ferio*, *Cesare*, *Camestres*, *Festion*, *Disamnis*, *Datisi* és *Ferison* esetében is. Megjegyzendő, hogy azonos számokkal jellemzett szillogizmusok cáfolatai mögött nem feltétlenül ugyanazon interpretációk állnak. Hamar kiderült, hogy mindössze háromelemű univerzumon megkonstruálható mind a 14 szillogizmuscáfolat. Ezután a különböző tulajdonságokkal rendelkező approximációs tereket és interpretációkat külön-külön is megvizsgáltuk. A táblázatok felső öt sora az eszközök jellemzői alapján mutatja a különböző approximációs terek tulajdonságait, azon esetekben, ha

1. legfeljebb 3 különböző alaphalmaz létezik ($|\mathfrak{B}| \leq 3$);
2. legfeljebb 2 különböző alaphalmaz létezik ($|\mathfrak{B}| \leq 2$);
3. az alaphalmazok lefedik az univerzumot;
4. az alaphalmazok nem fedik le az univerzumot;
5. az alaphalmazaink páronként diszjunktak.

A következő 4 sorban az eszközök között nem szereplő predikátumparaméterek alapján különítettük el azon eseteket, ahol

6. bármely $P \in Con$ esetén $\left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^+ \neq \emptyset$, azaz egyetlen predikátumparaméterhez tartozó pozitivitási tartomány sem üres;
7. bármely $P \in Con$ esetén $\left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^- \neq \emptyset$, azaz egyetlen predikátumparaméterhez tartozó negativitási tartomány sem üres;
8. bármely $P \in Con$ esetén $\left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^+ \neq \emptyset$ és $\left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^- \neq \emptyset$, azaz a predikátumparaméterekhez tartozó pozitivitási és negativitási tartományok egyike sem üres;

2.6. táblázat. Cáfolatok száma, ha $|U| = 3$

	$\langle U, \varrho \rangle$	<i>GAS</i>	<i>Barbara</i>	<i>Celarent</i>	<i>Darapti</i>	<i>Felapton</i>
$ \mathfrak{B} \leq 3$	32 256	63	72	120	252	216
$ \mathfrak{B} \leq 2$	14 336	28	24	24	72	72
$\bigcup \mathfrak{B} = U$	55 808	109	120	228	354	354
$\bigcup \mathfrak{B} \neq U$	9 216	18	0	0	0	0
diszjunkt $\mathfrak{T}$	7 168	14	24	24	72	72
$[\varrho(P)]^+ \neq \emptyset$	20 731	127	84	168	162	240
$[\varrho(P)]^- \neq \emptyset$	3 115	127	36	42	102	108
$[\varrho(P)]^\pm \neq \emptyset$	834	22	0	12	18	18
$\varrho(P) \neq \varrho(T)$	4 896	100	0	0	0	0
Összesen	65 024	127	120	228	354	354

2.7. táblázat. Cáfolatok száma, ha $|U| = 4$

	$\langle U, \varrho \rangle$	<i>GAS</i>	<i>Barbara</i>	<i>Celarent</i>	<i>Darapti</i>	<i>Felapton</i>
$ \mathfrak{B} \leq 3$	2 355 200	575	4 728	9 576	22 368	18 816
$ \mathfrak{B} \leq 2$	491 520	120	912	912	3 024	3 024
$\bigcup \mathfrak{B} = U$	132 288 512	32 297	117 696	219 936	253 212	314 004
$\bigcup \mathfrak{B} \neq U$	1 925 120	470	3 840	7 296	11 328	11 328
diszjunkt $\mathfrak{T}$	208 896	51	1 104	1 104	3 456	3 456
$[\varrho(P)]^+ \neq \emptyset$	64 139 967	32 767	110 448	193 776	158 676	274 104
$[\varrho(P)]^- \neq \emptyset$	689 831	32 767	12 168	25 716	58 356	52 920
$[\varrho(P)]^\pm \neq \emptyset$	229 940	2 680	1 368	10 608	15 732	15 252
$\varrho(P) \neq \varrho(T)$	5 918 720	30 580	540	4 752	8 652	6 036
Összesen	134 213 632	32 767	121 536	227 232	264 540	325 332

2.8. táblázat. A szillogizmusok cáfolatai

<i>Barbara, Darii, Darapti, Disamnis, Datisi</i>										
	$\varrho(P)$	$\varrho(S)$	$\varrho(M)$	$\varrho(T_1)$	$\varrho(T_2)$	$\varrho(T_3)$	$\varrho(T_4)$	$\llbracket P \rrbracket$	$\llbracket S \rrbracket$	$\llbracket M \rrbracket$
u_1	0	1	1	0	1	0	1	2	1	1
u_2	1	0	1	1	1	0	0	1	2	1
u_3	0	1	0	0	0	1	1	0	1	2
u_4	0	0	0	0	0	0	0	2	2	2

<i>Celarent, Ferio, Cesare, Festion, Felapton, Bocardo, Ferison</i>										
	$\varrho(P)$	$\varrho(S)$	$\varrho(M)$	$\varrho(T_1)$	$\varrho(T_2)$	$\varrho(T_3)$	$\varrho(T_4)$	$\llbracket P \rrbracket$	$\llbracket S \rrbracket$	$\llbracket M \rrbracket$
u_1	0	1	1	0	1	0	1	2	1	1
u_2	1	1	0	1	1	0	0	1	1	2
u_3	0	0	1	0	0	1	1	0	2	1
u_4	0	0	0	0	0	0	0	2	2	2

<i>Camestres, Baroco</i>										
	$\varrho(P)$	$\varrho(S)$	$\varrho(M)$	$\varrho(T_1)$	$\varrho(T_2)$	$\varrho(T_3)$	$\varrho(T_4)$	$\llbracket P \rrbracket$	$\llbracket S \rrbracket$	$\llbracket M \rrbracket$
u_1	1	1	0	0	1	0	1	1	1	2
u_2	1	0	1	1	1	0	0	1	2	1
u_3	0	1	0	0	0	1	1	2	1	0
u_4	0	0	0	0	0	0	0	2	2	2

9. egyetlen $P \in Con$ esetén sem találunk olyan $T \in \mathfrak{T}$ eszközt, amely esetén $\varrho(P) = \varrho(T)$, azaz a predikátumparaméterek és az eszközök interpretálása különböző.

Ahogy az a 2.7. táblázatból leolvasható, mind a 9 külön vizsgált esetben mind a 14 szillogizmus megcáfolható volt legalább 4 elemű univerzumot feltételezve (a 3 elemű univerzum nem minden esetben bizonyult elegendőnek).

A 2.8. táblázatban három olyan interpretációt gyűjtöttünk össze, amelyek között a 14 szillogizmus mindegyikéhez találunk cáfolatot.

2.5. A közelítés relevanciája

A következőkben azt a kérdést vizsgáljuk meg, hogy milyen feltételek mellett helyettesítheti a közelítő számítás eredménye a klasszikus szemantika esetében számítottat.

Formálisan: ha $\tau \in \{0, 1\}$ egy klasszikus értelemben vett igazságérték, akkor

$$\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = \tau \quad \Rightarrow \quad |A|_v^{\langle U, \varrho \rangle} = \tau$$

milyen A formulákra teljesül. A cél egy olyan kalkulus kidolgozása, amely segítségével egy-egy formula esetében eldönthető, hogy helyettesíti-e a közelítő számítás eredménye a klasszikusat. (Abból a feltevésből indulunk ki, hogy a közelítő számítás lényegesen gyorsabban végezhető el, így amikor csak lehet, inkább ezt alkalmazzuk.)

2.29. Tétel. *Ha egy $P(x)$ atomi formula a parciális approximációs téren kiértékelve nem értékréses, akkor igazságértéke megegyezik a klasszikus logikában ugyanazon interpretációban számított igazságértékkel, azaz*

$$\llbracket P(x) \rrbracket_v^{\langle U, \varrho \rangle} = \tau \quad \Rightarrow \quad |P(x)|_v^{\langle U, \varrho \rangle} = \tau.$$

Bizonyítás.

- Ha $\llbracket P(x) \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 1$, akkor van olyan $T \in \mathfrak{T}$, hogy $[\varrho(T)]^+ \subseteq [\varrho(P)]^+$ és $(\varrho(T))(u) = 1$, ezért $u \in [P]^+$, tehát $|P(x)|_{x \mapsto u}^{\langle U, \varrho \rangle} = 1$;
- ha $\llbracket P(x) \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 0$, akkor van olyan $T \in \mathfrak{T}$, hogy $[\varrho(T)]^+ \subseteq U \setminus [\varrho(P)]^+$ és $(\varrho(T))(u) = 0$, ezért $u \in [P]^-$, tehát $|P(x)|_{x \mapsto u}^{\langle U, \varrho \rangle} = 0$.

■

Legyen ez a kalkulusunk axiómasémája a szokásos tört alakban, feltéve hogy $\tau \in \{0, 1\}$ és $P \in \text{Con}$:

$$\frac{\emptyset}{\llbracket P(x) \rrbracket_v^{\langle U, \varrho \rangle} = \tau \quad \Rightarrow \quad |P(x)|_v^{\langle U, \varrho \rangle} = \tau}$$

A vonal alatti állítás akkor teljesül, ha teljesülnek a vonal feletti feltételek. Nincs nehéz dolgunk az ítéletlogikai fő összekötő jellel rendelkező formulák esetében:

$$\frac{\frac{\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = 1 - \tau \Rightarrow |A|_v^{\langle U, \varrho \rangle} = 1 - \tau}{\llbracket \neg A \rrbracket_v^{\langle U, \varrho \rangle} = \tau \Rightarrow |\neg A|_v^{\langle U, \varrho \rangle} = \tau}}{\frac{\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = \tau \Rightarrow |A|_v^{\langle U, \varrho \rangle} = \tau; \quad \llbracket B \rrbracket_v^{\langle U, \varrho \rangle} = \tau \Rightarrow |B|_v^{\langle U, \varrho \rangle} = \tau}{\llbracket (A \wedge B) \rrbracket_v^{\langle U, \varrho \rangle} = \tau \Rightarrow |(A \wedge B)|_v^{\langle U, \varrho \rangle} = \tau}}$$

$$\frac{\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = 1 - \tau \Rightarrow |A|_v^{\langle U, \varrho \rangle} = 1 - \tau; \quad \llbracket B \rrbracket_v^{\langle U, \varrho \rangle} = \tau \Rightarrow |B|_v^{\langle U, \varrho \rangle} = \tau}{\llbracket (A \supset B) \rrbracket_v^{\langle U, \varrho \rangle} = \tau \Rightarrow |(A \supset B)|_v^{\langle U, \varrho \rangle} = \tau}$$

Az ítéletlogikai jelekhez tartozó szabályok egyenes következményei a hozzájuk rendelt szemantikának, így ezek bizonyításával részletesen nem foglalkozunk, példaként csak az első adjuk meg $\tau = 1$ esetében.

Bizonyítás. Indirekt módon tegyük fel a következőket:

1. Az A formula olyan, hogy ha $\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = 0$ teljesül, akkor $|A|_v^{\langle U, \varrho \rangle} = 0$,

2. $\llbracket \neg A \rrbracket_v^{\langle U, \varrho \rangle} = 1$, de $| \neg A |_v^{\langle U, \varrho \rangle} \neq 1$.

Ezen indirekt feltevésekből az alábbiak következnek:

1. mivel a klasszikus logika nem értékréses, az utóbbi feltevés következménye, hogy $| \neg A |_v^{\langle U, \varrho \rangle} = 0$, tehát $| A |_v^{\langle U, \varrho \rangle} = 1$;
2. a Kleene-féle gyenge szemantika szerint $\llbracket \neg A \rrbracket_v^{\langle U, \varrho \rangle} = 1$ csak akkor teljesülhetett, ha $\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = 0$;
3. az első indirekt feltevés szerint $\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = 0$ esetén $| A |_v^{\langle U, \varrho \rangle} = 0$;

ami $| A |_v^{\langle U, \varrho \rangle}$ értékére vonatkozó ellentmondás. ■

A fenti szabályok segítségével belátható, hogy a kvantormentes kifejezések alsó közelítés segítségével számított értéke egybeesik a klasszikus szemantika szerint számított értékkel, feltéve hogy értékrés nem keletkezett.

A kvantorok estében a következőket mondhatjuk (a vonal feletti állítás minden $u \in U$ esetén érvényes kell legyen):

$$\frac{\llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 0 \Rightarrow | A |_{x \mapsto u}^{\langle U, \varrho \rangle} = 0}{\llbracket \forall x A \rrbracket^{\langle U, \varrho \rangle} = 0 \Rightarrow | \forall x A |^{\langle U, \varrho \rangle} = 0}$$

$$\frac{\llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 1 \Rightarrow | A |_{x \mapsto u}^{\langle U, \varrho \rangle} = 1}{\llbracket \exists x A \rrbracket^{\langle U, \varrho \rangle} = 1 \Rightarrow | \exists x A |^{\langle U, \varrho \rangle} = 1}$$

Látható, hogy lényegesen gyengébb állításokat tudtunk megfogalmazni. Példaként most is csak az elsőt mutatjuk meg (a változatosság kedvéért ez esetben direkt módszerrel), hiszen itt is az összekötő jelek szemantikájának egyszerű következményéről van szó.

Bizonyítás.

1. Tegyük fel, hogy az A formula olyan, hogy ha $\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = 0$ teljesül, akkor $| A |_v^{\langle U, \varrho \rangle} = 0$ is fennáll.
2. Ha $\llbracket \forall x A \rrbracket^{\langle U, \varrho \rangle} = 0$, akkor a kvantorhoz tartozó szemantika szerint van olyan $u \in U$ és $v(x) = u$, amely esetén $\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = 0$. Ebből feltevésünk szerint az következik, hogy $| A |_v^{\langle U, \varrho \rangle} = 0$, ekkor viszont $| \forall x A |_v^{\langle U, \varrho \rangle} = 0$. ■

Abból viszont, hogy $\llbracket \forall x A \rrbracket^{\langle U, \varrho \rangle} = 1$, nem következik, hogy $| \forall x A |^{\langle U, \varrho \rangle} = 1$, még abban az esetben sem, ha minden v esetén igaz az, hogy $\llbracket A \rrbracket_v^{\langle U, \varrho \rangle} = \tau$ esetén $| A |_v^{\langle U, \varrho \rangle} = \tau$.

Bizonyítás. Az állítás igazolására vegyünk egy egyszerű nyelvet, ahol $Con = \{P\}$ és $\mathfrak{T} = \{T\}$, továbbá konstruáljunk olyan $\langle U, \varrho \rangle$ interpretációt, ahol

$$[\varrho(T)]^+ \subset [\varrho(P)]^+ \quad \text{és} \quad [\varrho(P)]^- \neq \emptyset,$$

azaz az eszközhöz tartozó pozitivitási tartomány valódi részhalmaza a predikátum-paraméterhez tartozó pozitivitási tartománynak, továbbá a predikátumparaméterhez nemüres negativitási tartomány tartozik. Mindezek legalább 3 különböző univerzum-elem létezését feltételezik (ne felejtjük el, hogy az eszközök pozitivitási tartománya nem lehet üres). Létezik tehát olyan $u_1, u_2, u_3 \in U$, hogy

$$\begin{aligned} u_1 &\in [\varrho(T)]^+ \\ u_2 &\in [\varrho(P)]^+ \setminus [\varrho(T)]^+ \\ u_3 &\in [\varrho(P)]^- \end{aligned}$$

Világos, hogy u_3 tulajdonságai miatt $|\forall x P(x)|^{\langle U, \varrho \rangle} = 0$. Tekintve, hogy $[\downarrow P]^-$ üres (mivel T az egyetlen eszköz a nyelvben), és $[\downarrow P]^+$ halmaznak eleme u_1 , így $\llbracket \forall x P(x) \rrbracket^{\langle U, \varrho \rangle} = 1$. Mivel $P(x)$ atomi formula, így esetében már bizonyítottuk, hogy $\llbracket P(x) \rrbracket_v^{\langle U, \varrho \rangle} = \tau$ következménye a $|P(x)|_v^{\langle U, \varrho \rangle} = \tau$ (ahol $\tau \in \{0, 1\}$). ■

Megjegyzendő, hogy a kalkulus, melyet definiáltunk, kizárólag a logikai jelek szemantikája miatt alakult ki, és olyan bővebb nyelvben is érvényes lenne, amelyet nem szűkítünk le egyrétegű kvantálásra és egyetlen változóra. Maga a kalkulus egy olyan eszközt ad a kezünkbe, amely lehetővé teszi tetszőleges formula minősítését aszerint, hogy a közelítéssel számított igazságértéke egybeesik-e a klasszikus értékkel. Aból a feltevésből kiindulva, hogy az approximációs tér segítségével végzett számítás gyorsabb, tetszőleges formula kiértékelését kezdhethetjük a közelítés szerinti érték meghatározásával, majd ellenőrizhetjük, hogy megspórolható-e a klasszikus számítás.

Pesszimista szemantika

A korábbi tételek alapján kialakult tapasztalatokat felhasználva újragondoltuk a szemantikai definíciókat. Ennek során két szempontot tartottunk szem előtt: ahol lehet, próbáljuk meg csökkenteni az értékrés előfordulásának lehetőségét, továbbá definiáljuk úgy a kvantorok szemantikáját, hogy az véges univerzumon a szokásos módon visszavezethető legyen ítéletlogikai műveletekre:

$$|\forall x P(x)|^{\langle U, \varrho \rangle} = \bigwedge_{u \in U} |P(x)|_{x \mapsto u}^{\langle U, \varrho \rangle}, \quad |\exists x P(x)|^{\langle U, \varrho \rangle} = \bigvee_{u \in U} |P(x)|_{x \mapsto u}^{\langle U, \varrho \rangle}, \quad (2.9)$$

amely tulajdonság az approximációs térre épített szemantikával nem teljesül. Az új szemantikát – manapság divatos szóhasználattal élve – nevezhetjük *pesszimistának*², mivel a hiányzó tudásra úgy tekint, hogy az akár minden eddigi ismeretünknek is ellentmondhat. (Eddig például az univerzálisan kvantált kifejezést elfogadtuk igaznak, ha a közvetlen részformula néhány – legalább egy – univerzumelemet a kötött változó helyébe helyettesítve igaz volt, és egyetlen esetben sem hamis.)

A *pesszimista* szemantika esetén az interpretáció és az approximációs tér korábban bevezetett fogalmait megtartjuk, csak a nyelv szavainak értékét definiáljuk másképp:

2.30. Definíció. Egy $\mathcal{F} \in QF \cup Form$ (pesszimista) értéke alatt $\langle U, \varrho \rangle$ interpretációban és kvantormentes kifejezések esetében szintén adott $x \mapsto u$ (ahol $u \in U$)

²Ez a megközelítés eltér a [25]-ben megadott pesszimista, optimista és átlagos szemantikáktól.

változókiértékelés mellett a következőképpen definiált és $\lfloor \mathcal{F} \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle}$ formában jelölt igazságértéket értjük:

1. Egy $P(x) \in QF$ atomi kifejezés értéke az előbbi szemantikához képest nem változik:

$$\lfloor P(x) \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} \stackrel{\text{def}}{=} \llbracket P(x) \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle}.$$

2. Az ítéletlogikai összekötő jelek esetében az értékrés megjelenésének esélyét azal csökkentjük, hogy Kleene erős konnektívumainak (2.5. táblázat, 44. oldal) segítségével definiáljuk őket:

$$\begin{aligned} \lfloor \neg A \rfloor_v^{\langle U, \varrho \rangle} &\stackrel{\text{def}}{=} \neg^s \lfloor A \rfloor_v^{\langle U, \varrho \rangle}, \\ \lfloor (A \supset B) \rfloor_v^{\langle U, \varrho \rangle} &\stackrel{\text{def}}{=} \lfloor A \rfloor_v^{\langle U, \varrho \rangle} \supset^s \lfloor B \rfloor_v^{\langle U, \varrho \rangle}, \\ \lfloor (A \wedge B) \rfloor_v^{\langle U, \varrho \rangle} &\stackrel{\text{def}}{=} \lfloor A \rfloor_v^{\langle U, \varrho \rangle} \wedge^s \lfloor B \rfloor_v^{\langle U, \varrho \rangle}, \end{aligned}$$

ahol A és B vagy mindketten kvantortmentes (QF -beli) kifejezések, és ekkor v egy $x \mapsto u$ alakú helyettesítést jelöl, vagy A és B mindketten formulák, mely esetben v elmarad (helyettesítést nem adunk meg, hiszen a Form halmaz elemei mind zártak).

3. A kvantált kifejezések értéke – a 2.9-ben leírt elvárásoknak megfelelően – a következőképpen definiált:

$$\begin{aligned} \lfloor \forall x A \rfloor^{\langle U, \varrho \rangle} &\stackrel{\text{def}}{=} \begin{cases} 0 & \text{ha van olyan } u \in U, \text{ hogy } \lfloor A \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} = 0, \\ 1 & \text{ha minden } u \in U \text{ esetén } \lfloor A \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} = 1, \\ 2 & \text{egyébként;} \end{cases} \\ \lfloor \exists x A \rfloor^{\langle U, \varrho \rangle} &\stackrel{\text{def}}{=} \begin{cases} 1 & \text{ha van olyan } u \in U, \text{ hogy } \lfloor A \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} = 1, \\ 0 & \text{ha minden } u \in U \text{ esetén } \lfloor A \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} = 0, \\ 2 & \text{egyébként.} \end{cases} \end{aligned}$$

A pesszimista szemantika alapján kiegészíthetjük a kalkulust (amelynek korábbi szabályai az új szemantika esetében is érvényesek) két további szabállyal:

$$\begin{array}{c} \frac{\lfloor A \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} = 1 \Rightarrow \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 1}{\lfloor \forall x A \rfloor^{\langle U, \varrho \rangle} = 1 \Rightarrow \llbracket \forall x A \rrbracket^{\langle U, \varrho \rangle} = 1} \\ \frac{\lfloor A \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} = 0 \Rightarrow \llbracket A \rrbracket_{x \mapsto u}^{\langle U, \varrho \rangle} = 0}{\lfloor \exists x A \rfloor^{\langle U, \varrho \rangle} = 0 \Rightarrow \llbracket \exists x A \rrbracket^{\langle U, \varrho \rangle} = 0} \end{array}$$

Lényeges következmény, hogy a pesszimista szemantika alapján értékelt formula igazságértéke (feltéve, hogy értékrés nem keletkezett) egybeesik a formula klasszikus értelemben vett igazságértékével.

2.31. Tétel. Minden A kvantormentes kifejezés vagy formula esetén

$$\lfloor A \rfloor_v^{\langle U, \varrho \rangle} = \tau \quad \Rightarrow \quad |A|_v^{\langle U, \varrho \rangle} = \tau,$$

ahol $\tau \in \{0, 1\}$ igazságérték, $\langle U, \varrho \rangle$ tetszőleges interpretáció, v pedig a kvantormentes kifejezések esetén egy $x \mapsto u$ alakú helyettesítést jelöl (ahol $u \in U$).

Bizonyítás. A tételt a kalkulus felhasználásával a kvantormentes kifejezések és a formulák szerkezeti indukciójával igazoljuk. (Mivel a bizonyítás nem túl komplikált, rövidítjük úgy, hogy $\wedge$ és $\exists$ logikai jeleket eltávolítjuk a nyelvből.)

1. Atomi ($P(x)$ alakú) kifejezések esetében az állítás a 2.29. tétel miatt teljesül.

2. Tegyük fel, hogy A és B esetén az állítás teljesül, ekkor

(a) ha $\lfloor \neg A \rfloor_v^{\langle U, \varrho \rangle} = \tau$, akkor $\lfloor A \rfloor_v^{\langle U, \varrho \rangle} = 1 - \tau$, ami feltevésünk szerint azt jelenti, hogy $|A|_v^{\langle U, \varrho \rangle} = 1 - \tau$, következésképpen $|\neg A|_v^{\langle U, \varrho \rangle} = \tau$;

(b) ha $\lfloor (A \supset B) \rfloor_v^{\langle U, \varrho \rangle} = 0$, akkor $\lfloor A \rfloor_v^{\langle U, \varrho \rangle} = 1$ és $\lfloor B \rfloor_v^{\langle U, \varrho \rangle} = 0$, ennél fogva $|A|_v^{\langle U, \varrho \rangle} = 1$ és $|B|_v^{\langle U, \varrho \rangle} = 0$, így $|(A \supset B)|_v = 0$;

(c) ha $\lfloor (A \supset B) \rfloor_v^{\langle U, \varrho \rangle} = 1$, akkor

- vagy $\lfloor A \rfloor_v^{\langle U, \varrho \rangle} = 0$, és ezért $|A|_v^{\langle U, \varrho \rangle} = 0$,
- vagy $\lfloor B \rfloor_v^{\langle U, \varrho \rangle} = 1$, és ezért $|B|_v^{\langle U, \varrho \rangle} = 1$,

melyek következménye mindkét esetben $|(A \supset B)|_v^{\langle U, \varrho \rangle} = 1$;

(d) ha $\lfloor \forall x A \rfloor_v^{\langle U, \varrho \rangle} = 0$, akkor létezik legalább egy olyan $u \in U$ amely esetében $\lfloor A \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} = 0$, tehát feltevésünk szerint $|A|_{x \mapsto u}^{\langle U, \varrho \rangle} = 0$, ami miatt $|\forall x A|_v^{\langle U, \varrho \rangle} = 0$;

(e) ha $\lfloor \forall x A \rfloor_v^{\langle U, \varrho \rangle} = 1$, akkor minden $u \in U$ esetén $\lfloor A \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} = 1$, tehát feltevésünk szerint $|A|_{x \mapsto u}^{\langle U, \varrho \rangle} = 1$, ami miatt $|\forall x A|_v^{\langle U, \varrho \rangle} = 1$.

■

Sajnos a kvantorok *pesszimista* szemantikájának megvan az ára, mivel az értékrés keletkezésének valószínűsége megnő. Abban az esetben, ha az approximációs tér parciális, akkor $\forall x A$ közelített értéke nem lehet igaz, épp úgy, ahogy $\exists x A$ értéke sem lehet hamis.

2.32. Tétel. Legyen $A \in QF$ egy tetszőleges kvantormentes kifejezés és az $\langle U, \varrho \rangle$ egy olyan interpretáció az A kifejezés nyelvéhez, amely által generált $\mathcal{GAS}$ approximációs tér valóban parciális, azaz van olyan $u^* \in U$, hogy minden $T \in \mathfrak{T}$ eszköz esetében $\varrho(T)(u^*) = 0$. Ekkor

$$\lfloor \forall x A \rfloor \neq 1 \quad \text{és} \quad \lfloor \exists x A \rfloor \neq 0.$$

Bizonyítás. A tételben szereplő $\langle U, \varrho \rangle$ interpretáció esetében tetszőleges $P \in Con$ predikátumparaméter értékére igaz, hogy

$$u^* \notin \left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^+ \cup \left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^-, \quad \text{ezért} \quad \lfloor P(x) \rfloor_{x \mapsto u^*}^{\langle U, \varrho \rangle} = \llbracket P(x) \rrbracket_{x \mapsto u^*}^{\langle U, \varrho \rangle} = 2.$$

Ennek oka, hogy az approximációs teret az eszközök pozitivitási tartományai generálják, az u^* azonban ezek egyikének sem eleme. A következmény, hogy tetszőleges kvantortmentes kifejezés esetén

$$\lfloor A \rfloor_{x \mapsto u^*}^{\langle U, \varrho \rangle} = 2,$$

hiszen – akár az erős, akár a gyenge konnektívumok segítségével definiáltuk az ítéletlogikai összekötőket – az értékrés megmarad, ha minden közvetlen részkifejezés értékréses. ■

Ebben a fejezetben kizárólag az interpretáció által generált approximációs tér alsó közelítését használtuk fel, a felső közelítéssel nem foglalkoztunk. Természetesen analóg módon készíthettünk volna olyan szemantikát is, amely a felső közelítésen alapul, azaz ahol az atomi $P(x) \in QF$ kifejezések értéke a $\lceil P(x) \rceil_{x \mapsto u}^{\langle U, \varrho \rangle}$ -val jelölt és a következőképpen definiált $\{0, 1, 2\}$ halmazbeli érték:

$$\lceil P(x) \rceil_{x \mapsto u}^{\langle U, \varrho \rangle} \stackrel{def}{=} \begin{cases} 1 & \text{ha } u \in u(\lceil \varrho(P) \rceil^+); \\ 0 & \text{ha } u \in l(U \setminus [u(\varrho(P))]^+); \\ 2 & \text{egyébként.} \end{cases}$$

Tekintettel arra, hogy a predikátumokhoz rendelt függvények definiálásakor a felső és alsó közelítés ($\lfloor P(x) \rfloor$ és $\lceil P(x) \rceil$) meghatározása során a negativitási tartományt azonos módon határoztuk meg, kijelenthető, hogy

$$\lfloor P(x) \rfloor_{x \mapsto u}^{\langle U, \varrho \rangle} = 0 \quad \Rightarrow \quad |P(x)|_{x \mapsto u}^{\langle U, \varrho \rangle} = 0.$$

Sajnos azonban az atomi kifejezések esetében ennél tovább nem mehetünk.

$$\lceil P(x) \rceil_{x \mapsto u}^{\langle U, \varrho \rangle} = 1 \quad \not\Rightarrow \quad |P(x)|_{x \mapsto u}^{\langle U, \varrho \rangle} = 1.$$

2.6. Összegzés

Arisztotelész kategorikus szillogizmusainak vizsgálatára kizárólag az alsó közelítésen alapuló szemantikát készítettünk. Úgy gondoltuk, hogy mivel az alsó közelítés reprezentálja a biztos tudásunkat, így ez esetben van remény arra, hogy a szillogizmusok a közelítés használata mellett is érvényesek maradjanak. Ennek ellenére még viszonylag kis méretű univerzum mellett is sikerült megcáfolni ezeket.

3. fejezet

Approximáció állapottereken

A fejezetben bemutatott ötletek és eredmények [16]-ban kerültek korábban publikálásra.

3.1. A bizonytalanság kezelése klasszikus megoldáskeresőkben

Feltételezve, hogy reprezentálható egy probléma olyan logikai rendszeren, amely szemantikájának háttérét parciális approximációs tér adja, fel kell készíteni a megoldáskereső rendszereket az ebből adódó sajátosságok kezelésére. A logikai rendszerek vizsgálata során azzal a feltételezéssel éltünk, hogy az approximációs rendszer használata mellett szólhat az egyes kifejezések kiértékelési idejének csökkentése. Az állapottér-reprezentációk esetében nem elhanyagolható annak a lehetőségnek a mérlegelése sem, hogy kevesebb attribútum alapján próbáljuk meg jellemezni az állapotokat, csökkentve ezzel az állapotleírás bonyolultságát. Jelen alfejezetben azt a kérdést járjuk körbe, hogy miként lehet egyszerűen átalakítani a klasszikus megoldáskereső algoritmusokat úgy, hogy figyelembe tudják venni a közelítő számítások eredményét és az ezekben rejlő bizonytalanságot.

3.1.1. Visszalépéses kereső

Első lépésben az operátoralkalmazási előfeltételekre koncentrálunk. Az operátoralkalmazási előfeltételek válogatják ki az $a \in \mathcal{A}$ valódi állapotok közül azokat, amelyekre adott $o \in \mathcal{O}$ alkalmazható. Formálisan:

$$\text{dom}(o) = \{ a : a \in \mathcal{A} \wedge \varphi_o(a) \},$$

ahol a $\varphi_o(a)$ formula írja le az o operátoralkalmazási előfeltételét adott $a \in \mathcal{A}$ állapotban. Az egyszerűség kedvéért tegyük fel, hogy a $\varphi_o(a)$ formula a 2.28. definícióban megadott egyrétegű elsőrendű nyelvnek formulája, és így kiértékelhető a 2.30. definícióban rögzített pesszimista szemantika szabályai szerint.

3.1. Definíció. Azon $a \in \mathcal{A}$ állapotok halmaza, amelyre az $o \in \mathcal{O}$ operátor bizonyosan alkalmazható, a $\text{dom}^\downarrow(o)$ -val jelölt és a következőképpen definiált halmaz:

$$\text{dom}^\downarrow(o) = \{ a : a \in \mathcal{A} \wedge \lfloor \varphi_o(a) \rfloor = 1 \}.$$

3.2. Definíció. Azon $a \in \mathcal{A}$ állapotok halmaza, amelyre az $o \in \mathcal{O}$ operátor lehetséges, hogy alkalmazható, a $\text{dom}^\uparrow(o)$ -val jelölt és a következőképpen definiált halmaz:

$$\text{dom}^\uparrow(o) = \{ a : a \in \mathcal{A} \wedge \lfloor \varphi_o(a) \rfloor \neq 0 \}.$$

Informálisan úgy tekintjük, hogy amennyiben az operátoralkalmazási előfeltétel felírható a 2.28. szerinti formula formájában, akkor $a \in \text{dom}^\downarrow(o)$ pontosan akkor teljesül, ha a pesszimista szemantika szerint az alkalmazási előfeltétel igaz a -ra, míg $a \in \text{dom}^\uparrow(o)$ akkor, ha az alkalmazási előfeltétel a -ra nem hamis. Ekkor $\text{dom}^\downarrow(o)$ és $\mathcal{A} \setminus \text{dom}^\uparrow(o)$ halmazok egy orthopairt alkotnak, továbbá

$$\text{dom}^\downarrow(o) \subseteq \text{dom}(o) \subseteq \text{dom}^\uparrow(o) \subseteq \mathcal{A}, \quad (3.1)$$

így tulajdonképpen az operátor értelmezési tartományának alsó és felső közelítését definiáltuk. Valójában a célnak tetszőleges, az $\mathcal{A}$ halmazon értelmezett (1.1) szerinti – erős approximációs tulajdonságot teljesítő – approximációs pár megfelel.

A klasszikus megoldáskereső algoritmusokat most úgy alakítjuk át, hogy az állapotgráfot – amíg ez lehetséges – az approximáció alapján biztosnak vélt tudásunk mentén járjuk be, majd csak ezt követően kerüljön sor a lehetséges esetek közül azokra, ahol az eredeti operátoralkalmazási előfeltétel alapján kell döntenie a továbblépés lehetőségéről. A motiváció arra a feltevésre vezethető vissza, hogy $\lfloor \varphi_o(a) \rfloor$ lényegesen gyorsabban meghatározható, mint a klasszikus szemantika szerint számítandó $\lfloor \varphi_o(a) \rfloor$ érték; illetve az $a \in \text{dom}^\downarrow(o)$ és $a \in \text{dom}^\uparrow(o)$ relációk könnyebben ellenőrizhetők, mint az $a \in \text{dom}(o)$. Természetesen ha kiderül, hogy $a \notin \text{dom}^\downarrow(o)$ és $a \in \text{dom}^\uparrow(o)$ fennáll, akkor ki kell számítani, hogy $a \in \text{dom}(o)$ teljesül-e. Ezt ugyanakkor törekszünk annyira elodázní, amennyire csak lehet.

A klasszikus visszalépéses megoldáskereső algoritmus szerkezetén lényeges módosítást nem kell elvégezni. A megoldáskereső tulajdonságait adatbázis, műveletek és vezérlő szerinti bontásban röviden áttekintjük.

Adatbázis

A visszalépéses kereső adatbázisában az állapotter feltárt részéből csupán egy, a kezdőállapotból induló utat tart nyilván, amelyet veremként szokás implementálni. A verem tartalmát nevezzük aktuális útnak, a verem tetején levő állapotot aktuális állapotnak. A veremben az egyes állapotokról tárolt információkat kiegészítjük az állapotot előállító operátorral és az adott állapotban még nem alkalmazott, de alkalmazható operátorok listájával. Ez utóbbi információt két listában tartjuk nyilván:

1. még nem alkalmazott, de bizonyosan alkalmazható operátorokat tartalmazó, kezdetben:

$$\left\{ o \in \mathcal{O} : a \in \text{dom}^\downarrow(o) \right\};$$

2. és a még nem alkalmazott, de lehet, hogy alkalmazható operátorokat tartalmazó, kezdetben:

$$\left\{ o \in \mathcal{O} : a \in \text{dom}^\uparrow(o) \setminus \text{dom}^\downarrow(o) \right\}$$

listákat (halmazokat), ahol $a \in \mathcal{A}$ az a veremben tárolt állapot, amelyhez csatolni akarjuk a kiegészítő információt.

Műveletek

A visszalépéses kereső adatbázisán két műveletet definiálunk:

1. az *operátoralkalmazás* művelet használata esetén az aktuális állapotra alkalmazunk egy kiválasztott operátort, és az eredményt (kiszámítva az új állapothoz tartozó kiegészítő információkat) hozzáadjuk a veremhez, amivel növeljük az aktuális út hosszát;
2. a *visszalépés* művelet eltávolítja a verem tetejéről az aktuális állapotot, így csökkenti a tárolt út hosszát.

Az algoritmus jellemzése

A keresés menete nem különbözik a szokványos visszalépéses algoritmustól, egyszerűen csak azoknak az operátoroknak az alkalmazását vesszük előre egy-egy állapotban, amelyek alkalmazhatósága az approximációs tér használatával meghatározható (az eredeti formula értékének kiszámítása nélkül).

3.1. ábra. A visszalépéses megoldáskereső algoritmus informális lépései

- (1) Inicializáljuk a vermet, elhelyezve benne a kezdőállapotot (és a hozzá tartozó kiegészítő információkat).
 - (2) Megvizsgáljuk, hogy az aktuális állapot teljesíti-e a célfeltételt,
 - ha igen, akkor az algoritmus sikeresen véget ér, és megoldással terminál (a megoldás a verem tartalmából leolvasható);
 - ha nem, akkor az algoritmus a (3)-as ponton folytatódik.
 - (3) Megvizsgáljuk, hogy az aktuális állapotban vannak-e még ki nem próbált, de bizonyosan alkalmazható operátorok,
 - ha igen, akkor – egyben ezzel a listából eltávolítva – kiválasztunk egyet és alkalmazzuk, majd az algoritmus a (2)-es ponton folytatódik;
 - ha nem, akkor az algoritmus a (4)-es ponton folytatódik.
 - (4) Megvizsgáljuk, hogy az aktuális állapotban vannak-e még ki nem próbált, de lehetséges, hogy alkalmazható operátorok,
 - ha igen, akkor – egyben ezzel a listából eltávolítva – kiválasztunk egyet, majd meghatározzuk, hogy az aktuális állapot teljesíti-e a választott operátor alkalmazási előfeltételét. Ez utóbbi esetben az operátor alkalmazásával, majd a (2)-es ponton folytatjuk az algoritmust, ellenkező esetben a (4)-es pontot ismétljük meg;
 - ha nem, akkor az algoritmus az (5)-ös ponton folytatódik.
 - (5) Alkalmazzuk a visszalépés műveletet, majd megvizsgáljuk, hogy a verem üres-e,
 - ha igen, akkor az algoritmus befejezi működését, felismerve, hogy a probléma nem megoldható;
 - ha nem, akkor folytatjuk a (3)-as ponton.
-

3.1.2. Keresőfával keresők

A keresőfával keresők átalakításának első lépéseként előbb az állapottér definíciójához nyúlunk vissza. Az $\mathcal{A}$ állapothalmazt az állapotok jellemzői segítségével határozzuk meg a

$$\mathcal{T} = T_1 \times T_2 \times \dots \times T_n$$

Descartes-szorzat segítségével, ahol T_i az egyes jellemzők lehetséges értékeit tartalmazza: $\mathcal{T}$ elemei közül a kényszerfeltételek segítségével válogatjuk ki az $\mathcal{A} \subseteq \mathcal{T}$ állapothalmaz elemeit (a valódi állapotokat). A kényszerfeltételek rejtve megjelennek az operátoralkalmazási előfeltételekben is. A kényszerfeltételeket kiemelve az operátorok alkalmazási előfeltételeiből, általánosíthatjuk azokat.

3.3. Definíció. Az $f : \mathcal{T} \rightarrow \mathcal{T}$ függvény az $o \in \mathcal{O}$ operátor általánosítása, ha

1. $\text{dom}(o) = \left\{ a : a \in \left(\widehat{\text{dom}}(f) \cap \mathcal{A} \right) \wedge f(a) \in \mathcal{A} \right\}$, ahol $\widehat{\text{dom}}(f) \subseteq \mathcal{T}$, és
2. minden $a \in \text{dom}(o)$ esetén $o(a) = f(a)$.

Informálisan: az o operátort általánosító f függvény megadja az operátor hatását; az pedig, hogy egy o operátor alkalmazható-e egy adott $a \in \mathcal{A}$ állapotban, úgy dönthető el, hogy megvizsgáljuk, hogy a hozzá tartozó f függvény a szerinti képe valódi állapot-e. Nem kizárt, hogy az f függvény is rendelkezik $\widehat{\text{dom}}(f)$ által definiált alkalmazási előfeltétellel, így nem kötelező (sok esetben nem is lehet) minden alkalmazási előfeltételt a kényszerfeltételben megjeleníteni.

Az általánosítás lehetővé teszi, hogy a megoldáskereső algoritmusokban az f függvények használatával nemvalódi állapotok is megjelenhessenek a keresés során. (Épp úgy, ahogy a visszalépéses kereső átalakított változatában is megengedtük olyan operátorok felbukkanását, amelyek alkalmazhatóságában nem lehettünk biztosak.) Megjegyzendő, hogy f nem lehet parciális függvény, vagyis $\text{dom}(f) = \mathcal{T}$.

3.4. Definíció. Az $a \in \mathcal{A}$ állapotból az $a' \in \mathcal{T}$ bizonyosan közvetlenül elérhető, ha van olyan f általánosított operátor, amely esetén $a \in \text{dom}(f)$ és $a' = f(a)$, továbbá

$$\left[a \in \widehat{\text{dom}}(f) \wedge f(a) \in \mathcal{A} \right] = 1,$$

azaz a pesszimista szemantika szerint meghatározva teljesül az alkalmazási előfeltétel az a állapotban, és teljesül a kényszerfeltétel az $a' = f(a)$ állapotban.

3.5. Definíció. Az $a \in \mathcal{A}$ állapotból az $a' \in \mathcal{T}$ lehetséges, hogy közvetlenül elérhető, ha van olyan f általánosított operátor, amely esetén $a \in \text{dom}(f)$ és $a' = f(a)$, továbbá

$$\left[a \in \widehat{\text{dom}}(f) \wedge f(a) \in \mathcal{A} \right] = 2,$$

azaz a pesszimista szemantika szerint nem határozható meg, hogy teljesül-e az alkalmazási előfeltétel az a állapotban és teljesül-e a kényszerfeltétel az $a' = f(a)$ állapotban.

Megjegyzés. A közvetlen elérhetőséget az $\mathcal{A}$ halmazon – azaz csak valódi állapotok esetében – definiáltuk, így ahhoz, hogy a közvetlen elérhetőség tekintetében nyilatkozni

- (1) Inicializáljuk az adatbázist, elhelyezve benne a kezdőállapotból készített bizonyosan nyílt csúcsot.
 - (2) Megvizsgáljuk, hogy vannak-e még nyílt csúcsok az adatbázisban, majd
 - ha van még bizonyosan nyílt csúcs, akkor kiválasztunk egyet kiterjesztésre és folytatjuk a (3)-as ponton;
 - ha nincs bizonyosan, de van még lehetségesen nyílt csúcs, akkor kiválasztunk egyet kiterjesztésre és folytatjuk a (3)-as ponton;
 - ha nincsenek nyílt csúcsok, akkor az algoritmus befejezi működését, felismerve, hogy a probléma nem oldható meg.
 - (3) Kiterjesztjük a kiválasztott nyílt csúcsot, és ha a kiterjesztés során az algoritmus nem terminált, akkor folytatjuk a (2)-es ponton.
-

tudjunk, már biztosnak kell lenni abban, hogy amire az általánosított f operátort alkalmazni szeretnénk, az valódi állapot. Például $(f_1 \circ f_2)(a)$ csak akkor lehet $f_2(a)$ -ból közvetlenül elérhető, ha $f_2(a) \in \mathcal{A}$. Ezzel azt szeretnénk elkerülni, hogy a megoldáskeresés során az állapottérgráf bejárása közben a keresőben nem létező, egynél több élből álló, összefüggő élsorozatok jelenjenek meg.

A keresőfával megoldást kereső algoritmusok (szélességi kereső, mélységi kereső, optimális kereső, stb.) a megoldáskeresés során az állapottérgráf előállított részéből egy részfat tárolnak. A részfa minden csúcsát címkézzük most az alábbiak szerint¹:

- a *bizonyosan nyílt csúcs* olyan csúcs, amely még nem került kiterjesztésre, és amely olyan állapotot reprezentál, amely bizonyosan közvetlenül elérhető a fában szülőjeként megadott csúcs szerinti állapotból;
- a *lehetséges nyílt csúcs* olyan csúcs, amely még nem került kiterjesztésre, és amely olyan (nem feltétlenül valódi) állapotot reprezentál, amely lehetséges, hogy közvetlenül elérhető a fában szülőjeként megadott csúcs szerinti állapotból;
- a *zárt csúcs* – szokás szerint – olyan csúcs, amelyre már alkalmaztuk a kiterjesztés műveletét.

Azért döntöttünk a zárt csúcsok tárolása mellett, hogy elkerüljük az útvonalak ismételt bejárását ([31] 77. oldal). A keresés során ismételt útvonalak megjelenésétől még alternatív utaktól és köröktől mentes állapottérgráfok esetén is tartani kell, mivel a közelítés miatt megjelenhetnek az eredeti reprezentációban nem szereplő élek.

¹A [3]-ban megadott gráfbejáró algoritmusban fehér, szürke és fekete csúcsokkal találkozhatunk. Jelenlegi terminológiánk szerint a fehér csúcsok még nem kerültek az adatbázisba, a fekete csúcsok a már kifejtett (zárt) csúcsok, a szürke csúcsok pedig a bizonyosan nyílt csúcsoknak feleltethetők meg.

3.3. ábra. A kiterjesztés műveletének lépései

- (1) Ha a kiterjesztésre kiválasztott csúcs címkéje „*lehetséges nyílt*”, akkor ellenőrizzük, hogy a csúcs elérhető-e a szülőjéből a megfelelő operátor szerint,
 - ha igen, akkor a (2)-es ponton folytatjuk,
 - ha nem, úgy a csúcsot az adatbázisból töröljük és a kiterjesztést befejezzük;ha a kiválasztott csúcs „*bizonyosan nyílt*”, úgy szintén a (2)-es ponton folytatjuk.
- (2) Megvizsgáljuk, hogy a kiterjesztésre kiválasztott csúcs olyan állapotot reprezentál-e, amely teljesíti a célfeltételt, és
 - ha igen, akkor az algoritmus sikeresen befejezi működését, és a megoldás a tárolt keresőfából kiolvasható;
 - ha nem, akkor a kiválasztott csúcs címkéjét „*zárt*”-ra cseréljük, majd az algoritmus a következő ponton folytatódik.
- (3) Az általánosított operátorok segítségével előállítjuk az összes bizonyosan közvetlenül elérhető és lehetséges közvetlenül elérhető állapotot, majd minden állapot esetében az alábbiak szerint járunk el:
 - ha az új állapot még nem szerepelt az adatbázisban, új nyílt csúcsot készítünk, amely pontosan akkor lesz „*bizonyosan nyílt*” címkével ellátva, ha az állapotot a kiterjesztés során bizonyosan közvetlenül elérhető állapotként kaptuk;
 - ha az új állapot már szerepelt az adatbázisban, akkor az alábbi táblázat szerint járunk el:

	bizonyosan elérhető	lehetséges elérhető
<i>zárt</i>	nincs módosítás	nincs módosítás
<i>bizonyosan nyílt</i>	nincs módosítás	nincs módosítás
<i>lehetséges nyílt</i>	frissítés	feltételes frissítés

ahol a frissítés az adatbázisban tárolt csúcs cseréjét jelenti; míg a feltételes frissítés esetében megvizsgáljuk, hogy a csúcs elérhető-e a szülőjéből a megfelelő operátor szerint, és nemleges válasz esetén frissítünk.

A megoldáskeresés menetének illusztrációját egy fiktív állapottér-reprezentációt felhasználva a B függelék tartalmazza.

3.2. Az állapothalmaz közelítése

Amikor összehasonlítjuk egy probléma lehetséges állapottér-reprezentációit, fontos szempont az állapothalmaz elemszáma. Természetes, hogy amennyiben sikerül ki-

sebb elemszámú állapotterrel modellezni a problémát, úgy a megoldáskereső algoritmusoknak kisebb (kevesebb csúcsból és élből álló) gráfot kell bejárnia a megoldás előállításához, ugyanakkor tipikusan összetettebb kényszerfeltétel szükséges ahhoz, hogy definiáljuk a valódi állapotokat. Ebben az alfejezetben két egyszerű és jól ismert probléma segítségével járjuk körbe, hogy miként segít minket az approximációs tér feltárni a kapcsolatot ugyanazon probléma különböző modelljei között.

Első lépésként tekintsük úgy az állapothalmazt definiáló kényszerfeltételt, illetve a kényszerfeltételt leíró klasszikus elsőrendű logikai formulát, mint egy konjunkcióláncot. Valójában ez a megközelítés az állapotterén eléggé jellemző, a kényszerfeltételt szokás szerint több egyszerre teljesítendő kritériummal határozzák meg. Jelöljön $\varphi_i(a)$ egy klasszikus elsőrendű formulát, amelynek paraméterei az a állapot attribútumai:

$$\mathcal{A} = \{ a \in \mathcal{T} : \varphi_1(a) \wedge \varphi_2(a) \wedge \dots \wedge \varphi_k(a) \}$$

A megközelítés előnye, hogy így az $\mathcal{A}$ halmaz az egyes formulák pozitivitási tartományának – jelölje ezt most S_i , ahol $i = 1, 2, \dots, k$ – metszeteként definiálható:

$$S_i = \{ a \in \mathcal{T} : \varphi_i(a) \} \quad \text{esetén} \quad \mathcal{A} = \bigcap_{i=1}^k S_i \quad \text{és} \quad \mathcal{A}^c = \bigcup_{i=1}^k S_i^c,$$

ahol a komplementerhalmazokat $\mathcal{T}$ alaphalmaz felett értelmezzük, így $S^c = \mathcal{T} \setminus S$.

3.2.1. Az n -vezér probléma állapotter-reprezentációi

3.4. ábra. Az n -vezér probléma

A feladat, hogy egy n sorból és n oszlopból álló táblán helyezzünk el n darab vezért úgy, hogy azok ne üssék egymást (azaz minden sorba, oszlopba és átlóba legfeljebb egy darab kerüljön).

A következőkben tekintsük át a jól ismert – és a 3.4. ábrán megadott – n -vezér probléma egy szemléletes reprezentációját, amelyben a tábla minden cellájához egy értéket rendelünk, amely megmutatja, hogy található-e vezér az adott cellában. Ekkor:

$$\mathcal{T} = \{0, 1\}^{n \times n}$$

Ebben a modellben a célunk az, hogy a teljesen üres (kizárólag 0 értékeket tartalmazó) táblából – mint a_0 kezdőállapotból – kiindulva n különböző operátort alkalmazva lerakjunk egymás után n darab vezért úgy, hogy azok ne üssék egymást. Az állapotteret ez utóbbi feltétel segítségével definiáljuk²:

$$\mathcal{A} = \{ a \in \mathcal{T} : \varphi_1(a) \wedge \varphi_2(a) \wedge \varphi_3(a) \}, \text{ ahol } a = (a_{i,j})_{n \times n}$$

²A formulákat most csak illusztráljuk egy olyan klasszikus elsőrendű logikai nyelven, amely többfajta [17] és egyenlőségjeles. Az elképzelt interpretációban azt, hogy az a állapotban a tábla i . sorában és j . oszlopában áll-e vezér vagy sem, rendre $a_{i,j} = 1$, illetve $a_{i,j} = 0$ jelzi, míg az i, j, k, ℓ változók az $1, \dots, n$ halmazból vehetnek fel értékeket.

$\varphi_1(a)$: az a állapotban tetszőleges két különböző vezér különböző sorban áll:

$$\forall i \forall j \forall k \forall \ell (a_{i,j} = 1 \wedge a_{k,\ell} = 1 \wedge j \neq \ell \supset i \neq k);$$

$\varphi_2(a)$: az a állapotban tetszőleges két különböző vezér különböző oszlopban áll:

$$\forall i \forall j \forall k \forall \ell (a_{i,j} = 1 \wedge a_{k,\ell} = 1 \wedge i \neq k \supset j \neq \ell);$$

$\varphi_3(a)$: az a állapotban tetszőleges két különböző vezér különböző átlóban áll:

$$\forall i \forall j \forall k \forall \ell (a_{i,j} = 1 \wedge a_{k,\ell} = 1 \wedge (i \neq k \vee j \neq \ell) \supset |i - k| \neq |j - \ell|).$$

A célállapotok olyan valódi állapotok, ahol n darab vezér van a táblán (ami figyelembe véve a kényszerfeltételt, pontosan akkor teljesül, ha minden sorban van vezér):

$$\mathcal{C} = \{ a \in \mathcal{A} : \forall i \exists j (a_{i,j} = 1) \}$$

Operátorokat $n \cdot n$ darabot adunk meg, a tábla minden i sorában és j oszlopában álló cellához egyet-egyet. Ezeket most az általánosított operátorok segítségével definiáljuk:

$$\widehat{\text{dom}}(f_{i,j}) = \{ a \in \mathcal{A} : a_{i,j} = 0 \}$$

$$f_{i,j}(a) = a' \quad \text{úgy, hogy} \quad a'_{k,\ell} = \begin{cases} 1 & \text{ha } i = k \wedge j = \ell; \\ a_{k,\ell} & \text{egyébként.} \end{cases}$$

Bár ez a modell egyszerű és könnyen áttekinthető, a gyakorlatban mégsem használható. A gond az, hogy egy-egy állapotban – különösen a keresés elején – nagyon sok alkalmazható operátorunk van (például a kezdőállapotban épp $n \cdot n$), így hatalmas állapottérgráf keletkezik. Felismerve, hogy a cél elérése szempontjából az operátoralkalmazások sorrendje nem számít, de több különböző operátoralkalmazás-sorozat segítségével is elérhető ugyanazon célállapot, javíthatunk a reprezentáción. A szokásos ötlet az, hogy oszlopról oszlopra haladjunk a vezérek lerakásával. Így mindjárt a kezdőállapotban $n \cdot n$ alkalmazható operátor helyett csak n darab marad (azok, amelyek az első oszlop valamely sorába helyeznék el a következő vezért). Ezt a megszorítást a kényszerfeltételt kiegészítve az alábbi $\varphi_4(a)$ formulával fejezhetjük ki:

$\varphi_4(a)$: az a állapotban igaz, hogy ha valamely oszlopban vezér áll, akkor tetszőleges megelőző oszlopban is:

$$\forall i (\exists k (a_{k,i} = 1) \supset \forall j (j < i \supset \exists k (a_{k,j} = 1))).$$

A későbbiekben ki fogjuk használni a $\varphi_4(a)$ formula azon tulajdonságát, hogy az eddigi kényszerfeltételtől abban az értelemben független, hogy mind az $\mathcal{A}$, mind az $\mathcal{A}^c$ halmaz esetében a $\varphi_4(a)$ formulát teljesítő elemek valódi és nemüres részhalmazát képezik $\mathcal{A}$ -nak, illetve $\mathcal{A}^c$ -nek (kivéve persze az $n = 1$ esetet, amellyel most nem foglalkozunk).

A $\varphi_4(a)$ formulával kiegészítve az eddigieket, egy új (lényegesen kisebb elemszámú) állapothalmazt kapunk:

$$\mathcal{A}' = \{ a \in \mathcal{T} : \varphi_1(a) \wedge \varphi_2(a) \wedge \varphi_3(a) \wedge \varphi_4(a) \}$$

A második reprezentáció esetén (ami így az előzőtől csak a kényszerfeltételek megfogalmazásában különbözött) tekinthetjük úgy, hogy az $\mathcal{A}$ állapothalmaz $\mathcal{A}'$ alsó közelítésével dolgozunk, ami az állapottér komplementerét felhasználva felírható egy Pawlak-féle approximációs párral rendelkező parciális approximációs tér segítségével. Ehhez legyen

1. $U = \mathcal{T}$;
2. $\mathfrak{B} = \{S_1^c, \dots, S_4^c\}$, ahol $S_i = \{a \in \mathcal{T} : \varphi_i(a)\}$;
3. $\mathfrak{D}_{\mathfrak{B}}$ olyan, hogy $\mathfrak{B} \subset \mathfrak{D}_{\mathfrak{B}}$, $\emptyset \in \mathfrak{D}_{\mathfrak{B}}$ és $\mathfrak{D}_{\mathfrak{B}}$ az unió műveletre nézve zárt;
4. $\langle \mathfrak{I}, \mathfrak{u} \rangle$ (2.1. és 2.2. szerinti) Pawlak-féle approximációs pár.

Természetesen feltételeztük, hogy S_i^c nem üres halmaz, hiszen ellenkező esetben $S_i = \mathcal{T}$, ami azt jelentené, hogy $\varphi_i(a)$ minden állapotban teljesül, azaz nem is valódi feltétel.

A $\langle \mathcal{T}, \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{I}, \mathfrak{u} \rangle$ approximációs téren az új $\mathcal{A}'$ állapothalmaz komplementere az eredeti $\mathcal{A}$ állapothalmaz komplementerének felső közelítése:

$$\mathcal{A}'^c = \mathfrak{u}(\mathcal{A}^c), \quad \text{vagyis} \quad \mathcal{A}' = (\mathfrak{u}(\mathcal{A}^c))^c.$$

Mindehhez csak annyit kell belátni, hogy minden $S_i^c \in \mathfrak{B}$ esetén $\mathcal{A}^c \cap S_i^c \neq \emptyset$, ami

1. az S_1^c, S_2^c, S_3^c halmazokra azért teljesül, mert $\mathcal{A}^c = S_1^c \cup S_2^c \cup S_3^c$;
2. S_4^c esetében pedig azért, mert $(\mathcal{A} \cup S_4)^c \neq \emptyset$, hiszen volt olyan elem $\mathcal{A}^c$ -ben, amelyre a $\varphi_4(a)$ formula nem teljesült.

Ezzel annyit értünk el, hogy a probléma két különböző állapottér-reprezentációja közti kapcsolatot halmazapproximáció segítségével fejeztük ki.

3.2.2. A zebra probléma állapottér-reprezentációi

A zebra probléma is olyan jól ismert feladvány, amelynek modellezésére több különböző állapottér-reprezentáció is alkalmas lehet. Maga a probléma több változatban is létezik, és hívják Einstein- vagy Kennedy-tesztnek is. A jelenleg tárgyalt változatot a 3.5. ábrán ismertetjük, amelyet a [20] 225. oldalán közzétett változat alapján készítettünk az 5. állítás („a zöld ház rögtön az elefántcsont színű után áll”) apró átfogalmazásával annak érdekében, hogy a feladatnak ne csak egy megoldása legyen.

Tulajdonképpen a feladat 5 ház 5 különböző tulajdonságának a meghatározása a rendelkezésre álló 14 igaz állítás alapján. Az egyes házak egyes tulajdonságait egy 5 sorból és 5 oszlopból álló táblázatban foglaljuk össze, ahol az oszlopok az egyes házakat, a sorok a különféle tulajdonságokat reprezentálják.

Az állapottér-reprezentáció alapjául egy 5×5 -ös mátrixokból álló állapothalmazt választunk. A mátrix egyes elemei a táblázatban megfelelő cella értékét mutatják, megengedve, hogy egy-egy cella ismeretlen (kitöltetlen) legyen. Ennél fogva minden cella 6 különböző értéket vehet fel. A célunk, hogy a kezdetben teljesen kitöltetlen táblázatot jelölő a_0 kezdőállapotból lépésenként egy-egy kitöltetlen cella értékét meghatározva haladjunk a teljesen kitöltött, $\mathcal{C}$ -ben ábrázolt táblázatokig. Valódi állapotnak viszont csak olyan mátrixot tekintünk, amely olyan táblázatot jelöl, melyben

3.5. ábra. A zebra probléma

Egy utcában áll öt ház, amelyek mind különböző színűek, és a lakóik különféle nemzetiségűek, akik más-más állatot tartanak, különböző italokat fogyasztanak, és eltérő márkájú autókat vezetnek. Tudjuk továbbá:

1. *Az angol a piros házban lakik.*
2. *A spanyolnak van egy kutya.*
3. *Kakaót a zöld házban isznak.*
4. *Az ukrán tejáslikőrt iszik.*
5. *A zöld ház rögtön az elefántcsont színű ház mellett van.*
6. *Az Oldsmobile tulajdonosának csigái vannak.*
7. *A Ford tulajdonosa a sárga házban lakik.*
8. *Tejet a középső házban isznak.*
9. *A norvég balról az első házban lakik.*
10. *A Chevrolet-tulajdonos szomszédjának róka van.*
11. *A lótulajdonos szomszédja Ford márkájú autót vezet.*
12. *A Mercedes-Benz gazdája narancslevet iszik.*
13. *A japán Volkswagent vezet.*
14. *A norvég a kék ház mellett lakik.*

A kérdés: kié a zebra, és melyik házban isznak vizet?

(a már ismert tulajdonságok alapján) a feladatban megadott 14 mondatunk még igaz lehet. Egy-egy operátor feladata, hogy egy még ismeretlen értéket lecseréljen. Így, mivel mind az 5 ház mind az 5 tulajdonsága 5 különböző értéket vehet fel, 125 darab operátorunk lesz.

Az állapottér-reprezentáció teljes kifejtésétől most eltekintünk (az alapötlet [14]-ben is fellelhető), most csak az állapothalmazt vizsgáljuk. A feladat 14 állítása egy-egy feltételformula formájában jelentkezik a kényszerfeltételben. Az első állítás megfelelő elsőrendű nyelven formalizálva az alábbi alakot ölti:

$$\varphi_1(a) \wedge \varphi_2(a) \wedge \varphi_3(a),$$

ahol az alábbi formulákban t_0 a nyelvben az üres cellát jelölő konstans, $t_{i,j}$ pedig az i . tulajdonsághoz tartozó egy lehetséges értéket reprezentál.

$\varphi_1(a)$: az a állapotban igaz, hogy ha az i . házban angol ($t_{2,1}$) lakik, akkor a ház színe piros ($t_{1,1}$) vagy még nem ismert (t_0):

$$\forall i(a_{2,i} = t_{2,1} \supset a_{1,i} = t_{1,1} \vee a_{1,i} = t_0);$$

3.6. ábra. A zebra probléma megoldása (eredeti)

1.	2.	3.	4.	5.
<i>sárga</i>	<i>kék</i>	<i>piros</i>	<i>elefántcsont</i>	<i>zöld</i>
<i>norvég</i>	<i>ukrán</i>	<i>angol</i>	<i>spanyol</i>	<i>japán</i>
<i>Ford</i>	<i>Chevrolet</i>	<i>Oldsmobile</i>	<i>Mercedes</i>	<i>Volkswagen</i>
<i>víz</i>	<i>tojáslikőr</i>	<i>tej</i>	<i>narancslé</i>	<i>kakaó</i>
<i>róka</i>	<i>ló</i>	<i>csiga</i>	<i>kutya</i>	<i>zebra</i>

3.7. ábra. A zebra probléma megoldása (módosított)

1.	2.	3.	4.	5.
<i>sárga</i>	<i>kék</i>	<i>piros</i>	<i>zöld</i>	<i>elefántcsont</i>
<i>norvég</i>	<i>ukrán</i>	<i>angol</i>	<i>japán</i>	<i>spanyol</i>
<i>Ford</i>	<i>Chevrolet</i>	<i>Oldsmobile</i>	<i>Volkswagen</i>	<i>Mercedes</i>
<i>víz</i>	<i>tojáslikőr</i>	<i>tej</i>	<i>kakaó</i>	<i>narancslé</i>
<i>róka</i>	<i>ló</i>	<i>csiga</i>	<i>zebra</i>	<i>kutya</i>

$\varphi_2(a)$: az a állapotban igaz, hogy ha az i . ház színe piros, akkor lakójának nemzetisége angol vagy még nem ismert (t_0):

$$\forall i(a_{1,i} = t_{1,1} \supset a_{2,i} = t_{2,1} \vee a_{2,i} = t_0);$$

$\varphi_3(a)$: az a állapotban igaz, hogy ha az i . ház színe piros és a j . ház lakója angol, akkor i és j ugyanaz az oszlop (ház):

$$\forall i \forall j(a_{1,i} = t_{1,1} \wedge a_{2,j} = t_{2,1} \supset i = j).$$

Megjegyzendő, hogy az első állítás formalizálásához a kényszerfeltételben a $\varphi_1(a)$, $\varphi_2(a)$ és $\varphi_3(a)$ feltételek mindegyikére egyszerre szükség van, annak ellenére hogyha például egy állapotban már egyetlen cella sem vehet fel ismeretlen – azaz t_0 – értéket az első két sorban, akkor $\varphi_3(a)$ következménye a $\varphi_1(a) \wedge \varphi_2(a)$ feltételnek, így ekkor már felesleges. Ezt a gondolatot folytatva, ha egyszerűsíteni akarjuk a probléma reprezentációját (csökkenteni szeretnénk a feltételformulák logikai összetettségét), akkor $\varphi_1(a)$, $\varphi_2(a)$ és $\varphi_3(a)$ közül csak egyet tartunk meg a kényszerfeltételben. Ennek persze az az ára, hogy megnő a kényszerfeltételt kielégítő mátrixok (valódi állapotok) száma (ugyanakkor a célállapotoké nem változik). Az így nyert állapottér ekkor nem más, mint

$$\mathcal{A}' = (\mathcal{I}(\mathcal{A}^c))^c,$$

ahol a $\mathfrak{B}$ alaphalmazt az egyszerűsítés után megmaradó $\varphi(a)$ formulákhoz tartozó S_i halmazok alkotják.

3.3. SAT probléma közelítő megoldása

A következőkben a mesterséges intelligencia területén klasszikusnak számító állapot-tér-reprezentációs technikával leírható problémákat ítéletlogikai SAT problémaként kezeljük. Az állapottér-reprezentációkból klózhalmazt készítve keressük a problémák megoldását.

3.3.1. Véges diszkrét állapotterek mint SAT problémák

A fejezetben megmutatjuk, hogy egy probléma állapottér-reprezentációja hogyan alakítható át ítéletlogikai klózhalmaz kielégíthetőségének vizsgálatává, feltéve hogy a tekintett állapottér véges és diszkrét.

Legyen a p probléma egy lehetséges állapottér-reprezentációja $\langle \mathcal{A}, a_0, \mathcal{C}, \mathcal{O} \rangle$, ahol

$$\mathcal{A} \subseteq T_1 \times T_2 \times \dots \times T_n \quad \text{és} \quad T_i = \{t_{i,1}, t_{i,2}, \dots, t_{i,k_i}\}.$$

Ekkor a probléma egy-egy állapotát n darab tulajdonság határozza meg. Tetszőleges i . tulajdonság egy véges k_i elemű T_i halmazból vehet fel értéket. Az állapotok maximális száma a Descartes-szorzat elemszáma:

$$|\mathcal{C}| \ll |\mathcal{A}| \leq \prod_{i=1}^n k_i.$$

A Descartes-szorzat elemei közül az $\mathcal{A}$ állapothalmaz elemeit a kényszerfeltétel, míg ez utóbbi elemei közül a $\mathcal{C}$ célállapotokat a célfeltétel válogatja ki. Általában a célállapotok száma nagyságrendekkel kisebb, mint az állapottér elemszáma, sőt gyakran az is kérdés, hogy célállapot egyáltalán létezik-e.

Az $\langle \mathcal{A}, a_0, \mathcal{C}, \mathcal{O} \rangle$ diszkrét és véges állapottér-reprezentációhoz rendeljünk hozzá egy $L^{(0)} = \langle LC, Con, Form \rangle$ ítéletlogikai nyelvet az alábbiak szerint:

1. $LC = \{\neg, \vee\}$, mivel a nyelv formuláit ítéletlogikai klózhalmazok összeállítására akarjuk használni, így további logikai jelekre nincs szükség.
2. A nem logikai konstansok Con halmazát úgy alakítjuk ki, hogy minden T_i halmazba tartozó $t_{i,j}$ értékhez egy $p_{i,j}$ állításparamétert rendelünk. Később az egyszerűség kedvéért, ahol ez nem feltétlenül szükséges, ott a kettős indexelést elhagyjuk az alábbi rövidítés bevezetésével:

$$p_\ell \Rightarrow p_{i,j}, \quad \text{ahol} \quad \ell = j + \sum_{e=1}^{i-1} k_e.$$

Ezzel az egyszerűsítéssel egy $Con = \{p_1, p_2, \dots, p_m\}$ halmazt kapunk, ahol

$$m = \sum_{i=1}^n k_i.$$

3. A $Form$ halmaz olyan diszjunkcióláncokat tartalmaz, amelyek Con halmazbéli állításparaméterekből alkotott literálokból állnak, azaz ítéletlogikai klózkod.

Az $L^{(0)}$ nyelv egy interpretációja alatt egyszerűen egy $\varrho \subseteq Con$ halmazt értünk, amely szerint a $p \in Con$ atomi formula értéke pontosan akkor igaz, ha $p \in \varrho$. A formulák (klózik) értékét a szokásos módon definiáljuk.

Minden $u = (u_1, u_2, \dots, u_n) \in T_1 \times T_2 \times \dots \times T_n$ vektorhoz az $L^{(0)}$ nyelv egy-egy interpretációját rendelhetjük hozzá az alábbiak szerint:

$$\varrho_u = \{ p_{i,j} \in Con : u_i = t_{i,j} \}.$$

Ebben a szemléletben tehát $p_{i,j}$ egy-egy ϱ_u interpretációban azt jelenti, hogy az u vektorban az i . koordináta értéke $t_{i,j}$. Ettől kezdve a probléma célállapotának előállítása azt jelenti, hogy olyan ϱ interpretációt keresünk, amely olyan, a kényszerfeltételt teljesítő vektorhoz – azaz állapothoz – tartozik, amely esetén a célfeltétel is teljesül. Ehhez az említett feltételek $L^{(0)}$ nyelven történő megfogalmazására van szükség, amely 3 részből tevődik össze:

1. A kényszer- és célfeltételektől függetlenül az $L^{(0)}$ nyelv csak olyan interpretációit lehet vektorhoz rendelni, amelyekben minden $i = 1, \dots, n$ esetén

(a) $p_{i,j} \in Con$ atomok közül legalább egy igaz:

$$p_{i,1} \vee p_{i,2} \vee \dots \vee p_{i,k_i}, \quad (3.2)$$

(b) tetszőleges két különböző $p_{i,j}$ és $p_{i,\ell}$ atom közül legalább egy hamis:

$$\neg p_{i,j} \vee \neg p_{i,\ell}, \quad \text{ahol } j \neq \ell \text{ és } j, \ell \in \{1, 2, \dots, k_i\}. \quad (3.3)$$

Ugyanis ha egy i -re minden $p_{i,j}$ atom hamis, u_i nem lenne eleme T_i -nek, továbbá ha van két különböző $p_{i,j}$ és $p_{i,\ell}$ atom úgy, hogy mindkettő igaz, akkor u_i a két egymástól különböző $t_{i,j}$ és $t_{i,\ell}$ értékkel is egyenlő kellene legyen.

2. A kényszerfeltételek esetében most csak a legjellemzőbb esetekre térünk ki. Az általunk vizsgált problémákban a kényszerfeltétel olyan konjunkciós lánc, amelynek elemei olyan konjunktív normálformák, amelyekben a $p_{i,j}$ és $\neg p_{i,j}$ literálok rendre azt fejezik ki, hogy $(u_i = t_{i,j})$ és $(u_i \neq t_{i,j})$. Az így kapott konjunktív normálformákat alkotó elemi diszjunkciók az $L^{(0)}$ nyelv formulái.
3. A célfeltételek megfogalmazását a kényszerfeltételekéhez hasonló módon kell elvégezni.

A p probléma célállapotai a fent leírt klózikból képzett klózhalmaz modelljei. A p probléma megoldható, ha a klózhalmaznak van modellje, azaz kielégíthető. Innentől kezdve a feladat egy SAT probléma megoldása, ami közismerten egy NP-teljes probléma.

3.3.2. SAT feladat megoldása approximációs térrel

Legyen $L^{(0)} = \langle LC, Con, Form \rangle$ az az ítéletlogikai nyelv, amelyen a probléma megfogalmazásra került. Jelölje $\mathfrak{A}$ az ítéletváltozókból alkotott diszjunkt halmazpárok halmazát:

$$\mathfrak{A} = \{ \langle P, Q \rangle : P \subseteq Con \wedge Q \subseteq Con \wedge P \cap Q = \emptyset \}$$

A párok bal oldali komponenseire pozitív, míg jobb oldali komponenseire negatív ítéletváltozókként fogunk tekinteni. A későbbiekben az $\mathfrak{A}$ halmaz elemeit kétféleképpen is fel fogjuk használni:

1. Az $L^{(0)}$ *ítéletlogikai nyelv klózai*: Az $L^{(0)}$ ítéletlogikai nyelv klózait ábrázoljuk olyan $\mathfrak{A}$ -beli párral, amelynek bal oldalán a klóz pozitív, jobb oldalán a klóz negatív literáljait soroljuk fel. Például:

$$p_1 \vee \neg p_2 \vee \neg p_3 \quad \rightsquigarrow \quad \langle \{p_1\}, \{p_2, p_3\} \rangle.$$

Ez persze csak olyan klózok esetén tehető meg, amelyek nem tartalmazzák egyszerre a p_i és $\neg p_i$ literálokat is. Az $L^{(0)}$ nyelv klózhalmazai felírhatók $\mathfrak{A}$ részhalmazaiként. Azokat a klózokat, ahol ez az átírás nem lehetséges, egyszerűen elhagyjuk. (Tehetjük ezt azért, mert ezen klózok érvényes formulák.)

2. Az $L^{(0)}$ *ítéletlogikai nyelv interpretációinak egy családja*: Legyen $\mathcal{M}_{\langle P, Q \rangle}$ olyan interpretációk halmaza, amelyek a P -beli ítéletváltozókhoz *igaz*, míg a Q -beli ítéletváltozókhoz *hamis* értéket rendelnek:

$$\mathcal{M}_{\langle P, Q \rangle} = \{ \varrho : \varrho \subseteq \text{Con} \wedge P \subseteq \varrho \wedge Q \cap \varrho = \emptyset \}.$$

A $\langle P, Q \rangle$ pár által leírt $\mathcal{M}_{\langle P, Q \rangle}$ interpretációcsalád elemeinek száma:

$$|\mathcal{M}_{\langle P, Q \rangle}| = 2^{|\text{Con}| - |P| - |Q|}$$

Tulajdonképpen $\langle P, Q \rangle$ nem más, mint egy elemi konjunkció pozitív és negatív literáljaiban szereplő ítéletváltozóinak halmazpárja, $\mathcal{M}_{\langle P, Q \rangle}$ pedig ezen elemi konjunkció modelljei.

Megjegyzés. Később, az interpretációcsaládokon definiált műveletek tulajdonságainak bizonyítása során kihasználjuk azt, hogy $\varrho \notin \mathcal{M}_{\langle P, Q \rangle}$ akkor és csak akkor teljesül, ha

- van olyan $p \in \varrho$, hogy $p \in Q$ (vagyis p *igaz* a ϱ interpretációban és *hamis* az interpretációcsalád minden elemében), vagy
- van olyan $p \notin \varrho$, hogy $p \in P$ (vagyis p *hamis* a ϱ interpretációban és *igaz* az interpretációcsalád minden elemében).

Az $\mathfrak{A}$ halmaz segítségével definiáljuk a $\mathcal{SAT}$ problémát megoldó $\mathcal{SAT}$ függvényt – $\mathcal{SAT}$ solver-t – a következőképpen:

$$\mathcal{SAT} : 2^{\mathfrak{A}} \rightarrow 2^{\mathfrak{A}}$$

A $\mathcal{SAT}$ függvény nemcsak azt a kérdést kell megválaszolja, hogy a klózhalmaz kielégíthető-e, hanem esetünkben azt is elvárjuk tőle, hogy megadja a klózhalmaz modelljeit (feltéve, hogy léteznek ilyenek). Úgy is megfogalmazhatjuk a feladatot, hogy a klózhalmazból képzett konjunktív normálformát kell diszjunktív normálformává átalakítani. Természetesen az előbbi feladat megoldása többféleképp is előállhat, a $\mathcal{SAT}$ solver implementációjától függően. Néhány lehetséges implementálási technika megtalálható [8]-ban.

Tekintsünk egy egyszerű példát az $L^{(0)} = \langle \{p_1, p_2, p_3\}, \{\neg, \vee\}, Form \rangle$ nyelven. Legyen adott egy probléma a nyelv klózainak alábbi halmazaként definiálva:

$$\{ p_1 \vee \neg p_2, \neg p_1 \vee \neg p_2 \vee p_3 \}$$

Ekkor a SAT probléma megoldása alatt az alábbi hozzárendelést értjük:

$$\underbrace{\{ \langle \{p_1\}, \{p_2\} \rangle, \langle \{p_3\}, \{p_1, p_2\} \rangle \}}_{\text{klózek / CNF}} \xrightarrow{SAT} \underbrace{\{ \langle \{p_1, p_3\}, \emptyset \rangle, \langle \emptyset, \{p_2\} \rangle \}}_{\text{interpretációcsaládok / DNF}}$$

$$(p_1 \vee \neg p_2) \wedge (\neg p_1 \vee \neg p_2 \vee p_3) \Leftrightarrow (p_1 \wedge p_3) \vee \neg p_2$$

A megoldás azon interpretációcsaládok halmaza, amelyek modelljei a $p_1 \wedge p_3$ vagy $\neg p_2$ elemi konjunkcióknak.

3.6. Definíció. Legyen $\langle P, Q \rangle, \langle P', Q' \rangle \in \mathfrak{A}$ olyan, hogy $P \cap Q' = \emptyset$ és $Q \cap P' = \emptyset$. Ekkor a $\sqcap$ parciális művelet alatt a következőt értjük:

$$\langle P, Q \rangle \sqcap \langle P', Q' \rangle \stackrel{def}{=} \langle P \cup P', Q \cup Q' \rangle.$$

3.7. Definíció. Legyen $\langle P, Q \rangle \in \mathfrak{A}$ és $\langle P', Q' \rangle \in \mathfrak{A}$ két tetszőleges halmazpár. Defináljuk ekkor az $\sqcup$ műveletet a következőképpen:

$$\langle P, Q \rangle \sqcup \langle P', Q' \rangle \stackrel{def}{=} \langle P \cap P', Q \cap Q' \rangle.$$

3.8. Tétel. Ha $\langle P, Q \rangle \sqcap \langle P', Q' \rangle$ definiált, akkor

$$\mathcal{M}_{\langle P, Q \rangle \sqcap \langle P', Q' \rangle} = \mathcal{M}_{\langle P, Q \rangle} \cap \mathcal{M}_{\langle P', Q' \rangle},$$

vagyis két $\mathfrak{A}$ -béli pár metszeteként meghatározott interpretációcsalád egybeesik a párok által meghatározott interpretációcsaládok metszetével.

Bizonyítás. Az állítást indirekt bizonyítjuk.

1. Tegyük fel, hogy van olyan $\varrho \in \mathcal{M}_{\langle P, Q \rangle \sqcap \langle P', Q' \rangle}$ de $\varrho \notin \mathcal{M}_{\langle P, Q \rangle} \cap \mathcal{M}_{\langle P', Q' \rangle}$. Ha ϱ nem eleme a két interpretációcsalád metszetének, akkor ϱ legalább az egyik interpretációcsaládnak nem eleme, legyen az most $\mathcal{M}_{\langle P, Q \rangle}$. Ekkor
 - (a) vagy van olyan $p_i \in \varrho$, ahol $p_i \in Q$, és ez esetben $p_i \in Q \cup Q'$, amelynek következtében $\varrho \notin \mathcal{M}_{\langle P, Q \rangle \sqcap \langle P', Q' \rangle}$;
 - (b) vagy van olyan $p_i \notin \varrho$, ahol $p_i \in P$, és ez esetben $p_i \in P \cup P'$, amelynek következtében $\varrho \notin \mathcal{M}_{\langle P, Q \rangle \sqcap \langle P', Q' \rangle}$.
2. Tegyük fel, hogy van olyan $\varrho \notin \mathcal{M}_{\langle P, Q \rangle \sqcap \langle P', Q' \rangle}$, ahol $\varrho \in \mathcal{M}_{\langle P, Q \rangle} \cap \mathcal{M}_{\langle P', Q' \rangle}$. Az előbbi kétféleképp teljesülhet:
 - (a) vagy van olyan $p_i \in \varrho$, ahol $p_i \in Q \cup Q'$, így viszont ha $p_i \in Q$, akkor $\varrho \notin \mathcal{M}_{\langle P, Q \rangle}$, vagy ha $p_i \in Q'$, akkor $\varrho \notin \mathcal{M}_{\langle P', Q' \rangle}$, következésképp ϱ nem lehet eleme az interpretációcsaládok metszetének;

- (b) vagy van olyan $p_i \notin \varrho$, ahol $p_i \in P \cup P'$, így viszont ha $p_i \in P$, akkor $\varrho \notin \mathcal{M}_{\langle P, Q \rangle}$, vagy ha $p_i \in P'$, akkor $\varrho \notin \mathcal{M}_{\langle P', Q' \rangle}$, következésképp ϱ nem lehet eleme az interpretációcsaládok metszetének.

■

3.9. Tétel. *Legyenek $\langle P, Q \rangle, \langle P', Q' \rangle \in \mathfrak{A}$ halmazpárok, ekkor*

$$\mathcal{M}_{\langle P, Q \rangle \sqcup \langle P', Q' \rangle} \supseteq \mathcal{M}_{\langle P, Q \rangle} \cup \mathcal{M}_{\langle P', Q' \rangle},$$

vagyis $\mathfrak{A}$ -béli párok uniójaként megadott interpretációcsalád bővebb lehet, mint a párok által leírt interpretációcsaládok uniója.

Bizonyítás. Legyen $L^{(0)}$ olyan ítéletlogikai nyelv, ahol $Con = \{p_1, p_2\}$. Tekintsük az alábbi két $\mathfrak{A}$ -béli párt, illetve ezek unióját:

$$\langle \{p_1\}, \emptyset \rangle \sqcup \langle \{p_2\}, \emptyset \rangle = \langle \emptyset, \emptyset \rangle,$$

továbbá legyen ϱ az $L^{(0)}$ nyelv olyan interpretációja, ahol $\varrho = \emptyset$. Ekkor

1. $\varrho \in \mathcal{M}_{\langle \emptyset, \emptyset \rangle}$, hiszen ez utóbbi az $L^{(0)}$ nyelv összes interpretációját tartalmazza,
2. $\varrho \notin \mathcal{M}_{\langle \{p_1\}, \emptyset \rangle}$, mivel $p_1 \notin \varrho$, továbbá $\varrho \notin \mathcal{M}_{\langle \{p_2\}, \emptyset \rangle}$, mivel $p_2 \notin \varrho$,
3. a fentiekből következik, hogy $\varrho \notin \mathcal{M}_{\langle \{p_1\}, \emptyset \rangle} \cup \mathcal{M}_{\langle \{p_2\}, \emptyset \rangle}$.

Ezzel mutattunk példát arra, hogy a párok uniójához tartozó interpretációk családjá tartalmazhat olyan interpretációt, amely a párokhoz tartozó interpretációcsaládok egyikében sem szerepel.

A tétel második felét indirekt bizonyítjuk. Tételezzük fel, hogy van olyan ϱ interpretáció, ami eleme az $\mathcal{M}_{\langle P, Q \rangle}$ interpretációcsaládnak (és ezen interpretációcsalád bármely interpretációcsaláddal vett uniójának), de $\varrho \notin \mathcal{M}_{\langle P, Q \rangle \sqcup \langle P', Q' \rangle}$. A feltevés második feléből az alábbiak valamelyike következik:

1. van olyan $p \in Con$, amely esetén $p \in \varrho$ és $p \in Q \cap Q'$, de ekkor $p \in Q$, amelyből következően $\varrho \notin \mathcal{M}_{\langle P, Q \rangle}$, vagy
2. van olyan $p \in Con$, amely esetén $p \notin \varrho$ és $p \in P \cap P'$, de ekkor $p \in P$, amelyből következően $\varrho \notin \mathcal{M}_{\langle P, Q \rangle}$.

Mivel mindkét esetben ellentmondás állt elő, így a tétel bizonyítást nyert.

■

A továbbiakban a 3.9. tételben megadott tulajdonságot fogjuk felhasználni az interpretációcsaládok felső közelítésének definiálására. Valójában ennek segítségével a problémák célállapotainak halmazát szeretnénk felülről közelíteni.

3.3.3. Halmazmetszeteken alapuló approximációs tér

A következőkben olyan felső közelítést szeretnénk definiálni, aminek segítségével az interpretációcsaládokat interpretációcsaládok uniójával helyettesíthetjük (közelíthetjük). Amint azt az előző fejezetben láttuk, az interpretációcsaládot definiáló halmazpár uniója alkalmasnak látszik a célra. Az unió művelet a halmazpárok tagonkénti

metszetével került definiálásra, innen jön az ötlet, hogy a következőkben olyan approximációs teret alkossunk, amely halmazmetszetek segítségével ad felső közelítést. A minimális célkitűzés, hogy olyan konstrukció kerüljön kialakításra, ami a felső közelítés normalitását és monotonitását garantálja.

Legyenek $B_1, B_2, \dots, B_k$ egy U alaphalmaz nemüres részhalmazai. Defináljuk az approximációs teret a következő módon³:

1. A $\mathfrak{B}$ alaphalmaz a B_i halmazcsalád alábbi kiegészítése:

$$\mathfrak{B} = \{B_1, B_2, \dots, B_k, \hat{B}\}, \quad \text{ahol} \quad \hat{B} = \bigcup_{i=1}^k B_i$$

Tekintettel arra, hogy a halmazcsalád tagjaira további megszorítást nem alkalmaztunk, könnyen előállhat, hogy $\bigcup \mathfrak{B} \subset U$, azaz az approximációs tér parciális. A későbbiekben tárgyalt alkalmazás során is ez lesz a tipikus eset.

2. $\mathfrak{D}_{\mathfrak{B}}$ tartalmazza a definiálható halmazokat, ami $\mathfrak{B}$ olyan kiterjesztése, hogy

- (a) $\emptyset \in \mathfrak{D}_{\mathfrak{B}}$
- (b) $\bigcap \mathcal{B} \in \mathfrak{D}_{\mathfrak{B}}$ minden nemüres $\mathcal{B} \subseteq \mathfrak{B}$ esetén.

3. Az approximációs teret csak felső közelítésre szeretnénk használni, ezért az $\langle \mathfrak{l}, \mathfrak{u} \rangle$ approximációs párból csak $\mathfrak{u}$ -t definiáljuk, mégpedig:

$$\mathfrak{u}(S) = \bigcap \{B : B \in (\mathfrak{B} \cup \{\emptyset\}) \wedge (S \cap \hat{B}) \subseteq B\}.$$

Ha valamilyen formális oknál fogva mégis szükség volna az alsó közelítésre (de nem szükséges észszerűbb alsó közelítést definiálni), és nem szeretnénk bővíteni a definiálható halmazok körét, akkor azt mondjuk, hogy $\mathfrak{l} = \mathfrak{u}$.

Megjegyzés. A felső közelítés metszetként való definiálásakor használt

$$\{B : B \in (\mathfrak{B} \cup \{\emptyset\}) \wedge (S \cap \hat{B}) \subseteq B\}$$

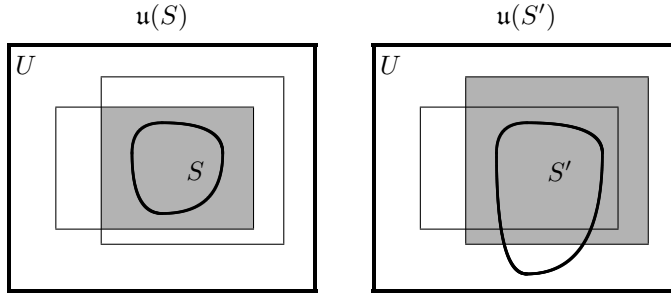
halmazcsalád garantáltan nem üres, hiszen $\hat{B} \in \mathfrak{B}$ halmaznak $S \cap \hat{B}$ bizonyosan részhalmaza. Ennél fogva $\mathfrak{u}(S) \subseteq \hat{B}$. Ha $S \cap \hat{B} = \emptyset$, akkor a halmazcsaládban $B \in \mathfrak{B} \cup \{\emptyset\}$ miatt az üres halmaz megjelenik, ezért ekkor $\mathfrak{u}(S) = \emptyset$.

A 3.8. ábra a felső közelítést illusztrálja egy egyszerű approximációs téren. Az ábrákon két vékonyabb kerettel szegélyezett téglalap (balról jobbra B_1 és B_2) segítségével definiáljuk az approximációs teret. Az illusztráció szerint $B_1 \cap B_2 \neq \emptyset$, továbbá $B_1 \not\subseteq B_2$ és $B_2 \not\subseteq B_1$. Az alaphalmazokat $\hat{B} = B_1 \cup B_2$ halmazzal kiegészítve a következő definiálható halmazokat kapjuk:

$$\mathfrak{B} = \{B_1, B_2, \hat{B}\}, \quad \mathfrak{D}_{\mathfrak{B}} = \{\emptyset, B_1 \cap B_2, B_1, B_2, \hat{B}\}$$

³Az egyértelműség kedvéért B és S következetesen mindig U egy részhalmazát jelölik, míg $\mathcal{B}$ egy U részhalmazaiából álló halmazcsaládot, $\mathfrak{B}$ pedig az approximációs tér alaphalmazát. Továbbá egy egyetlen elemből álló halmazcsalád esetében a halmazcsalád tagjainak metszete egybeesik a halmazcsalád egyetlen elemével.

3.8. ábra. A felső közelítés illusztrációja



A bal oldali ábrán olyan S halmaz közelítését jelöli a besatírozott terület, amely egyaránt részhalmaza B_1 -nek és B_2 -nek:

$$u(S) = \bigcap \{B_1, B_2, \hat{B}\}$$

A jobb oldali ábrán olyan S' halmaz közelítését illusztráltuk, amely $S \subset S'$ és $S \not\subseteq \hat{B}$. Bár S' egyik alaphalmaznak sem részhalmaza, $S' \cap \hat{B}$ több alaphalmaznak részhalmaza, így:

$$u(S') = \bigcap \{B_2, \hat{B}\}$$

Mivel $\hat{B} \subset U$, így parciális approximációs tér állt elő.

3.10. Tétel. Az u függvény teljesíti az alábbi tulajdonságokat,

1. u monoton, vagyis ha $S \subseteq S'$, akkor $u(S) \subseteq u(S')$,
2. $u(\emptyset) = \emptyset$ (u teljesíti a normalitási kritériumot),
3. minden $S \in \mathfrak{D}_{\mathfrak{B}}$ esetén $u(S) = S$.

Bizonyítás.

1. Indirekt tegyük fel, hogy van olyan S és S' , amely esetén

$$S \subseteq S', \quad \text{de} \quad u(S) \not\subseteq u(S').$$

Ekkor a következők teljesülnek:

- (a) Mivel $u(S) \not\subseteq u(S')$, így van olyan $x \in u(S)$, hogy $x \notin u(S')$.
- (b) $x \notin u(S')$ miatt van olyan $B \in \mathfrak{B} \cup \{\emptyset\}$, amely esetén $x \notin B$ és $S' \cap \hat{B} \subseteq B$.
- (c) Ha $S' \cap \hat{B} \subseteq B$, akkor $S \cap \hat{B} \subseteq B$, hiszen $S \subseteq S'$.
- (d) Mivel $u(S) \subseteq B$ és $x \notin B$, így $x \notin u(S)$.

Az első és utolsó pontban $x \in u(S)$ és $x \notin u(S)$ ellentmondás.

2. Ha $S = \emptyset$, akkor minden $B \in \mathfrak{B} \cup \{\emptyset\}$ esetén igaz, hogy $S \cap \hat{B} \subseteq B$, így

$$u(\emptyset) = \emptyset \cap B_1 \cap \dots \cap B_k \cap \hat{B} = \emptyset.$$

3. Az $S \in \mathfrak{D}_{\mathfrak{B}}$ definiálható halmazok felső közelítéséről az alábbiakat mondhatjuk:

- (a) $S = \emptyset$ esetben az előző pontban foglaltak szerint $u(S) = \emptyset$.
- (b) Ha $S \in \mathfrak{D}_{\mathfrak{B}}$ nemüres halmaz, akkor előáll $S = \bigcap \mathcal{B}$ alakban, valamely $\mathcal{B} \subseteq \mathfrak{B}$ nemüres halmazcsalád tagjainak metszeteként. Ekkor:
 - i. $u(S) \subseteq S$, mert $\mathcal{B}$ minden eleme szerepel abban a halmazcsaládban, amely elemeinek metszeteként $u(S)$ előáll, hiszen minden $B \in \mathcal{B}$ teljesíti az $S \cap \hat{B} \subseteq B$ feltételt.
 - ii. $S \subseteq u(S)$ minden $S \subseteq \hat{B}$ halmaz esetében (így ez esetben is) fennáll. Ekkor ugyanis $S \cap \hat{B} \subseteq B$ pontosan akkor teljesül, ha $S \subseteq B$. Így S felső közelítését most olyan halmazok metszeteként definiáljuk, amelyek mindegyikében szerepel S minden eleme.

Összegezve: $S \subseteq u(S) \subseteq S$, tehát $S = u(S)$.

■

Megjegyzés. A 3.10. tétel következménye, hogy $\mathfrak{l} = u$ esetén az $\langle U, \mathfrak{B}, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{l}, u \rangle$ rendezett ötös egy parciális approximációs tér, hiszen ekkor

1. $\mathfrak{l}, u : 2^U \rightarrow 2^U$;
2. $\mathfrak{l}$ és u definiálható $(\mathfrak{l}(S), u(S) \subseteq \mathfrak{D}_{\mathfrak{B}})$;
3. $\mathfrak{l}$ és u monoton;
4. $u(\emptyset) = \emptyset$;
5. ha $S \in \mathfrak{D}_{\mathfrak{B}}$, akkor $\mathfrak{l}(S) = S$;
6. teljesül a gyenge approximációs tulajdonság (ha $S \in 2^U$, akkor $\mathfrak{l}(S) \subseteq u(S)$).

Jobb alsó közelítés definiálását azért nem tekintettük szükségesnek, mert ahogyan ez később bemutatásra kerül, az approximációs teret a problémák egyszerűsítésére használva, kizárólag a felső közelítés segítségével dolgozunk majd.

3.3.4. A megoldás közelítése

Az approximációs tér és a felső közelítés használatbavétele előtt kissé át kell formálni a probléma megoldásának leírását. A lényeg persze nem változik, $\mathcal{SAT}$ probléma megoldása egy diszjunktív normálforma. A diszjunktív normálformát alkotó elemi konjunkciókat eddig halmazpárként ábrázoltuk, szétválasztva a pozitív és negatív literálokat:

$$\mathcal{SAT}(\mathfrak{p}) = \{\langle P_1, N_1 \rangle, \dots, \langle P_k, N_k \rangle\},$$

ahol ha a probléma leírásának nyelve $L^{(0)} = \langle LC, Con, Form \rangle$, akkor minden P_i és N_i halmaz esetén $P_i, N_i \subseteq Con$. Most az approximációs tér univerzumát a nyelv literáljai adják:

$$U = \{ \ell_i : p_i \in Con \} \cup \{ \bar{\ell}_i : p_i \in Con \},$$

minden $p_i \in Con$ állításparaméterhez egy ℓ_i pozitív és egy $\bar{\ell}_i$ negatív literált rendelve. Ezzel minden elemi konjunkciót leíró halmazpár ábrázolható U részhalmazaként.

A $(2^{Con})^2$ halmazpárok és 2^U literálhalmazok között definiáljuk az alábbi kölcsönösen egyértelmű leképezést:

$$f_L : (2^{Con})^2 \rightarrow 2^U \quad \text{és} \quad f_L^{-1} : 2^U \rightarrow (2^{Con})^2$$

$$\begin{aligned} f_L(\langle P, Q \rangle) &= \{ \ell_j : p_j \in P \} \cup \{ \bar{\ell}_j : p_j \in Q \} \\ f_L^{-1}(S) &= \langle \{ p_i : \ell_i \in S \}, \{ p_i : \bar{\ell}_i \in S \} \rangle \end{aligned}$$

Az approximációs tér alaphalmazait a $SAT(\mathbf{p})$ megoldásból generáljuk, minden $SAT(\mathbf{p})$ -beli halmazpárhoz U megfelelő részhalmazát hozzárendelve az alábbi módon:

$$\begin{aligned} \mathfrak{B} &= \{ B_1, B_2, \dots, B_k, \hat{B} \} \\ B_i &= f_L(\langle P_i, Q_i \rangle), \quad i = 1, 2, \dots, k \\ \hat{B} &= B_1 \cup B_2 \cup \dots \cup B_k \end{aligned} \tag{3.4}$$

Definiáljuk továbbá az A halmazt a következőképpen:

$$\begin{aligned} A &= \left\{ \ell_j : \bigwedge_{i=1}^k (\ell_j \in B_i \vee \bar{\ell}_j \in B_i) \right\} \cup \\ &\cup \left\{ \bar{\ell}_j : \bigwedge_{i=1}^k (\ell_j \in B_i \vee \bar{\ell}_j \in B_i) \right\}, \end{aligned} \tag{3.5}$$

ahol így az A halmaz azokat a literálokat tartalmazza, amelyekben található állításparaméter (akár pozitív, akár negatív literálban) az alaphalmazok által reprezentált minden elemi konjunkcióban megjelenik.

A $\mathbf{p}$ probléma $SAT(\mathbf{p})$ megoldásának felső közelítése alatt a megoldásban szereplő halmazpárok által generált approximációs téren számított következő halmazt értjük:

$$\{ u(B_i \cap A) : i = 1, \dots, k \}$$

A cél továbbra is a probléma megoldását leíró halmaz egyszerűsítése, amely tekintetben kijelenthető, hogy

$$|\{ u(B_i \cap A) : i = 1, \dots, k \}| \leq k.$$

Valójában akkor sikerül egyszerűsíteni a leírást, ha léteznek olyan egymástól különböző i és j indexű elemek $\mathfrak{B}$ -ben, ahol $u(B_i \cap A) = u(B_j \cap A)$.

3.1. táblázat. Illusztráció: probléma megoldásának közelítése

Con	B_1	B_2	B_3	$\hat{B}$	A	$\uparrow B_1$	$\uparrow B_2$	$\uparrow B_3$
p_1	+	+	−	$\pm$	$\pm$	+	+	−
p_2	+	+	+	$\pm$	$\pm$	+	+	+
p_3	−	−	−	−	$\pm$	−	−	−
p_4	−	−	·	−	·	−	−	·
p_5	+	−	·	$\pm$	·	·	·	·
p_6	·	−	+	$\pm$	·	·	·	+

Példa

A következőkben egy egyszerű példán tekintjük át a közelítés mibenlétét. Ehhez tételezzük fel, hogy a probléma olyan ítéletlogikai nyelven van megfogalmazva, ahol $Con = \{p_1, p_2, \dots, p_6\}$, és valamely részproblémájának megoldásaként a következő diszjunktív normálforma állt elő:

$$(p_1 \wedge p_2 \wedge \neg p_3 \wedge \neg p_4 \wedge p_5) \vee (p_1 \wedge p_2 \wedge \neg p_3 \wedge \neg p_4 \wedge \neg p_5 \wedge \neg p_6) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3 \wedge p_6)$$

A három elemi konjunkciót rendre a következő három halmazpár reprezentálja:

$$\langle \{p_1, p_2, p_5\}, \{p_3, p_4\} \rangle, \quad \langle \{p_1, p_2\}, \{p_3, p_4, p_5, p_6\} \rangle, \quad \langle \{p_2, p_6\}, \{p_1, p_3\} \rangle$$

A Con halmaz minden eleméhez két univerzumelemet – egy pozitív és egy negatív literált – rendelünk, így az $U = \{\ell_1, \ell_2, \dots, \ell_6\} \cup \{\bar{\ell}_1, \bar{\ell}_2, \dots, \bar{\ell}_6\}$ univerzumhalmazt kapjuk. A három elemi konjunkció, mint U három részhalmaza, a következőképpen ábrázolható:

$$\begin{aligned} p_1 \wedge p_2 \wedge \neg p_3 \wedge \neg p_4 \wedge p_5 &\rightsquigarrow B_1 = \{\ell_1, \ell_2, \bar{\ell}_3, \bar{\ell}_4, \ell_5\}, \\ p_1 \wedge p_2 \wedge \neg p_3 \wedge \neg p_4 \wedge \neg p_5 \wedge \neg p_6 &\rightsquigarrow B_2 = \{\ell_1, \ell_2, \bar{\ell}_3, \bar{\ell}_4, \bar{\ell}_5, \bar{\ell}_6\}, \\ \neg p_1 \wedge p_2 \wedge \neg p_3 \wedge p_6 &\rightsquigarrow B_3 = \{\bar{\ell}_1, \ell_2, \bar{\ell}_3, \ell_6\}. \end{aligned}$$

Ezeket – a korábbiak szerint – a $\hat{B} = B_1 \cup B_2 \cup B_3$ halmazzal kiegészítve kapjuk meg az alaphalmazokat ($\mathfrak{B}$ elemeit). A B_1 , B_2 és B_3 halmazok egyike sem tartalmaz ellentétesen negált literálpárt. A 3.1. táblázat segítségével illusztráljuk az approximációs teret. A táblázat egy-egy sora megmutatja, hogy mely halmazban mely literálok találhatóak meg. Rendre +, − és $\pm$ jelöli, hogy az adott sornak megfelelő állításparaméter a tekintett oszlophoz tartozó halmazban csak mint pozitív, vagy csak mint negatív literál szerepel, vagy pozitív és negatív literálban egyaránt megjelenik.

Az A halmazt a (3.5) szerint kapjuk úgy, hogy az alaphalmazok mindegyikében megjelenő p_1 , p_2 és p_3 ítéletváltozókból pozitív és negatív literálokat képzünk:

$$A = \{\ell_1, \ell_2, \ell_3\} \cup \{\bar{\ell}_1, \bar{\ell}_2, \bar{\ell}_3\}$$

A táblázat utolsó három oszlopa tartalmazza a generáló halmazok A -val való metszetének felső közelítését. Mivel ez $\uparrow B_1$ és $\uparrow B_2$ tekintetében egybeesik, így a közelítés

eredményeképp kevesebb különböző halmazt kaptunk.

$$\begin{aligned}\uparrow B_1 = \uparrow B_2 = \{\ell_1, \ell_2, \bar{\ell}_3, \bar{\ell}_4\} &\rightsquigarrow p_1 \wedge p_2 \wedge \neg p_3 \wedge \neg p_4, \\ \uparrow B_3 = \{\bar{\ell}_1, \ell_2, \bar{\ell}_3, \ell_6\} &\rightsquigarrow \neg p_1 \wedge p_2 \wedge \neg p_3 \wedge p_6.\end{aligned}$$

Ezzel a probléma megoldásának leírása a következő két elemi konjunkcióra egyszerűsödött (és ahogy ezt később látni fogjuk, pontatlanná is vált):

$$\langle \{p_1, p_2\}, \{p_3, p_4\} \rangle, \quad \langle \{p_2, p_6\}, \{p_1, p_3\} \rangle.$$

A következőkben néhány tétel segítségével pontosítjuk a közelítés tulajdonságait.

3.11. Tétel. *Legyen $\mathfrak{B} = \{B_1, \dots, B_k, \hat{B}\}$ egy parciális approximációs tér alaphalmazainak halmaza az U univerzum felett, valamint $E_1, \dots, E_n$ halmazok U nemüres részhalmazai. Ha minden $B \in \mathfrak{B}$ esetén igaz, hogy*

$$|B \cap E_i| \leq 1, \quad \text{ahol } i = 1, \dots, n,$$

akkor tetszőleges $S \subseteq U$ halmaz felső közelítése esetén is igaz, hogy

$$|u(S) \cap E_i| \leq 1, \quad \text{ahol } i = 1, \dots, n,$$

azaz a felső közelítés megtartja az alaphalmazok fent megadott tulajdonságát.

Bizonyítás. A tétel egyszerűen annak a következménye, hogy az $u(S)$ halmaz $\mathfrak{B}$ -beli halmazok segítségével (azok metszeteként) definiált. Mivel a $\mathfrak{B}$ -beli halmazok között nem szerepel olyan, amelyben bármely E_i halmaznak egynél több eleme is megjelenne, így $u(S)$ -ben sem lesz ez másként. ■

A 3.11. tétel abban segít minket, hogy belássuk, a közelítés eredményeképpen előállt halmazok nem fognak ellentétes literálpárokat tartalmazni, amennyiben a definiálható halmazok sem tartalmaztak ilyet. (Ekkor az E_i halmazokat $E_i = \{\ell_i, \bar{\ell}_i\}$ alakban állítjuk elő minden $i = 1, \dots, k$ esetén.) Sajnos azonban a tételt nem tudjuk alkalmazni már a 3.1. táblázatban bemutatott példa esetében sem, mivel a $\hat{B}$ halmaz nem teljesíti a tételben megkövetelt feltételt. Ugyanakkor valójában nem tetszőleges halmazok, hanem csak a $B_1, \dots, B_k$ halmazok részhalmazainak közelítését kell előállítani. Ehhez kevésbé általános formában fogalmazzuk át a tételt.

3.12. Tétel. *Legyen $\mathfrak{B} = \{B_1, \dots, B_k, \hat{B}\}$ egy parciális approximációs tér alaphalmazainak halmaza az U univerzum felett, valamint $E_1, \dots, E_n$ halmazok U nemüres részhalmazai. Ha van olyan $B \in \mathfrak{B}$, amely esetén igaz, hogy*

$$|B \cap E_i| \leq 1, \quad \text{ahol } i = 1, \dots, n,$$

akkor tetszőleges $S \subseteq B$ halmaz felső közelítése esetén is igaz, hogy

$$|u(S) \cap E_i| \leq 1, \quad \text{ahol } i = 1, \dots, n.$$

Bizonyítás. Tekintettel arra, hogy B eleme lesz az $u(S)$ közelítésére használt halmazsaládnak, így $u(S)$ -nek nem lehet olyan eleme, amely B -ben ne szerepelt volna, innentől viszont a tételben megfogalmazott állítás nyilvánvaló. ■

A tétel következménye, hogy ha egy interpretációcsalád leírását felülről közelítjük, akkor eredményül szintén egy interpretációcsalád keletkezik, vagyis egyszerűen visszaalakíthatjuk az $S' = u(S)$ felső közelítéseket halmazpárokká úgy, hogy $f_L^{-1}(S') \in \mathfrak{A}$.

3.13. Tétel. *Legyen adott egy, a $\mathcal{SAT}(\mathfrak{p})$ segítségével generált approximációs tér $\mathfrak{B} = \{B_1, B_2, \dots, B_k, \hat{B}\}$ alaphalmazokkal. Ekkor tetszőleges $B \in \mathfrak{B} \setminus \{\hat{B}\}$ és $A \subseteq U$ halmaz esetén teljesül, hogy $\mathcal{M}_{f_L^{-1}(B)} \subseteq \mathcal{M}_{(f_L^{-1} \circ u)(B \cap A)}$. Másképp szólva, az $f_L^{-1}(B)$ pár segítségével leírt interpretációk megtalálhatók a $(f_L^{-1} \circ u)(B \cap A)$ felső közelítésben is.*

Bizonyítás. Tekintettel arra, hogy $B \cap A \subseteq B$, így a 3.12. tétel miatt $f \circ u(B \cap A) \in \mathfrak{A}$. Ráadásul $u(B \cap A)$ metszetsorozat segítségével definiált, melynek tagjai közt ott találjuk B -t, így a 3.9. tétel miatt teljesül az állítás. ■

3.3.5. Közelítő megoldáskereső algoritmus

Legyen adott egy p probléma, amelynek diszkrét és véges állapotterét n jellemző írja le $\mathcal{A} \subseteq T_1 \times T_2 \times \dots \times T_n$ alakban. Készítsük el a probléma leírását (kényszer- és célfeltételeit) az $L^{(0)} = \langle \{\neg, \vee\}, Con, Form \rangle$ nyelven, ahol minden T_i halmaz minden eleméhez pontosan egy ítéletváltozót rendeltünk Con -ban. Jelölje $\mathfrak{p}$ a probléma kényszer- és célfeltételeit tartalmazó klózhalmazt. Ekkor a $\mathfrak{p}$ probléma megoldása alatt a $\mathcal{SAT}(\mathfrak{p})$ halmazt értjük. A $\mathcal{SAT}(\mathfrak{p})$ közelítő megoldására használjuk az alábbi algoritmust.

1. Bontsuk fel a p problémát (klózhalmazt) $\mathfrak{r}_1, \mathfrak{r}_2, \dots, \mathfrak{r}_n$ részproblémára (klózhalmazokra) úgy, hogy

$$\mathfrak{r}_i = \left\{ \langle P, Q \rangle \in \mathfrak{p} : \bigvee_{j=1}^{|T_i|} p_{i,j} \in (P \cup Q) \right\}. \quad (3.6)$$

Ez azt jelenti, hogy a probléma n darab jellemzője szerint egy-egy klózhalmazt készítünk, melyekbe olyan klózok kerülnek, amelyeknek legalább egy literálja a tekintett jellemzőhöz tartozó ítéletváltozót is tartalmaz.

2. Oldjuk meg a részproblémákat, azaz készítsük el a $\mathcal{SAT}(\mathfrak{r}_i)$ halmazokat. Ha bármelyik részprobléma kielégíthetetlen, akkor a probléma nem megoldható.
3. Minden $\mathcal{SAT}(\mathfrak{r}_i)$ halmazt egyszerűsítsünk az alábbiak szerint:
 - (a) a (3.4) szerint alakítsuk ki $\mathcal{SAT}(\mathfrak{r}_i)$ -hez az approximációs teret, majd
 - (b) a (3.5) szerint állítsuk elő az A halmazt, végezetül
 - (c) helyettesítsük a $\mathcal{SAT}(\mathfrak{r}_i)$ halmazt önmaga felső közelítésével:

$$\uparrow \mathcal{SAT}(\mathfrak{r}_i) = \left\{ (f_L^{-1} \circ u)(B_j \cap A) : j = 1, \dots, k \right\}.$$

4. Keressük meg a részproblémák közös megoldását, azaz gyűjtsünk össze olyan ϱ interpretációkat – mint lehetséges megoldásokat –, amelyekre teljesül, hogy

$$\varrho \in \mathcal{M}_{\langle P_1, Q_1 \rangle} \cap \dots \cap \mathcal{M}_{\langle P_i, Q_i \rangle}, \quad \text{ahol} \quad \langle P_i, Q_i \rangle \in \uparrow \text{SAT}(\mathbf{r}_i).$$

Ehhez felhasználhatjuk a korábban definiált $\sqcap$ műveletet. A keresési algoritmus lehet egyszerű visszalépéses kereső.

5. A lehetséges ϱ megoldásokon tesztelve a kényszer- és célfeltételt, válogassuk ki a probléma valódi megoldásait.

3.3.6. A közelítő megoldáskeresés jellemzése

Az algoritmus jellemzését érdemes a motiváció tisztázásával kezdeni. Nyilvánvaló, hogy az ítéletlogikai klózhalmazok segítségével reprezentált problémák megoldására léteznek módszerek, hogy mást ne mondjunk, a SAT solver maga is számos jól bevált implementációval rendelkezik, ráadásul a bemutatott új algoritmus is feltételezi, hogy rendelkezésünkre áll egy ilyen implementáció, amelyet a részproblémák megoldásánál kell használni. A probléma valójában a feladat ábrázolásával van. Azzal, hogy a reprezentációt egy állapottér alapján úgy állítjuk össze, hogy minden jellemző minden lehetséges értékéhez egy ítéletváltozót rendelünk, jócskán megnövekszik a leírás bonyolultsága. (Ezt a következő fejezetben egy példa segítségével igyekszünk majd számszerűen is kifejezni.) Ennek eredményeként könnyen előfordulhat, hogy egy olyan probléma esetében, amit állapottéren keresőfával megoldást kereső vagy akár visszalépéses algoritmussal megoldani nem jelent kihívást, akkora keresési teret kapunk az ítéletlogikai reprezentációban, amivel egy egyszerű SAT implementáció nem boldogul (vagy legalábbis az utóbbi csúfosan alulmarad, ami a megoldás előállításához szükséges időt illeti). A jelenség oka, hogy az a tudás, ami a probléma jellemzőiről az állapottérben implicit benne volt, a SAT probléma leírásában elvész. Természetesen a klózhalmazban ott vannak azok a feltételek, amelyek a jellemzők értékének egyediségére vonatkoznak, de ezek belesimulnak a reprezentációba. Ha adott egy n jellemzőből álló, jellemzőnként m lehetséges értékkel rendelkező állapottér, akkor a keresési tér mérete az alábbiak szerint alakul:

$$\underbrace{m^n}_{\text{állapotok}} \ll \underbrace{2^{m*n}}_{\text{interpretációk}}.$$

Az előző fejezetben bemutatott algoritmus célja a keresési tér méretének kordában tartása. Szemléletesen az algoritmus maga egy visszalépéses kereséssel zárul, amely egy olyan fát kell bejárjon, amelynek mélységét a részproblémákra bontás, szélességét pedig az approximációs terek segítségével végrehajtott felső közelítés csökkenti. A részproblémákat úgy alakítjuk ki, hogy ezek SAT feladatként történő megoldása nagyságrendekkel kisebb feladat legyen, mint a teljes probléma megoldása. Tulajdonképpen azt a tudást csempésszük vissza a SAT problémába, ami az ábrázolás miatt implicit megvolt az állapottér-reprezentációban.

Az algoritmusnak két kritikus paramétere van:

1. Általában a részproblémákra bontás nem triviális feladat, úgy kell ugyanis összeválogatni a részproblémákat, hogy megoldásaik segítségével az eredeti probléma megoldása összeállítható maradjon.
2. Az approximációs tereken végzett felső közelítés szempontjából az A halmaz megválasztása lényeges kérdés. Minél kevesebb elemet tartalmaz ugyanis az A halmaz, annál jobban csökken a részproblémák megoldásait leíró elemi konjunkciók darabszáma. (Tekintettel arra, hogy az elemi konjunkciók A -val való metszetének keressük a felső közelítését, így minél kevesebb elemet tartalmaz A , annál valószínűbb, hogy különböző elemi konjunkciók metszete A -val egybeesik, és így egybeesik a felső közelítésük is.) Ugyanakkor ezzel párhuzamosan növekszik a közelítés miatt felmerült hibák száma (növekszik az olyan megoldásjelölt interpretációk száma, amelyek valójában nem megoldásai a részproblémának).

E két paraméter megválasztása egymással összefügg. A részproblémákra bontás tekintetében az segít minket, hogy ismerjük a probléma állapotter-reprezentációját, amelyből a probléma leírásának nyelvét kialakítottuk. A részproblémákat a (3.6) szerint azért a probléma jellemzőit felhasználva készítettük, mert így a megoldásokban az egy-egy jellemzőhöz tartozó összes ítéletváltozó értéke ismert lesz, azaz ezek az ítéletváltozók meg fognak jelenni az elemi konjunkciókban. Az A halmaz a (3.5) pontban szereplő definíciója miatt megtartja ezeket az ítéletváltozókat, így ezek részei lesznek a felső közelítésnek is.

3.3.7. A zebra probléma közelítő megoldása

A zebra probléma állapotter-reprezentációi közül válasszuk kiinduló alapnak azt, amely az 5 ház lakójának 5 különböző tulajdonságát rendezett elemhuszonötös formájában ábrázolja. Az egyes tulajdonságokhoz tartozó tartományok elemeit az egyszerűség kedvéért egy-egy vektorban rögzítjük, így definiálva sorrendet köztük. A választott sorrendnek nincs jelentősége, és esetünkben megegyezik azzal a sorrenddel, ahogy az egyes értékek a problémát leíró 14 állításban megjelentek:

$$\begin{aligned} t_1 &= \langle \text{piros, zöld, elefántcsont, sárga, kék} \rangle \\ t_2 &= \langle \text{angol, spanyol, ukrán, norvég, japán} \rangle \\ t_3 &= \langle \text{Oldsmobile, Ford, Chevrolet, Mercedes, Volkswagen} \rangle \\ t_4 &= \langle \text{kakaó, tojáslikőr, tej, narancslé, víz} \rangle \\ t_5 &= \langle \text{kutya, csiga, róka, ló, zebra} \rangle \end{aligned}$$

Az állapotter-reprezentáció A halmaza a jellemzők segítségével a következő alakban definiálható:

$$\begin{aligned} T_i &= \{t_{i,1}, t_{i,2}, \dots, t_{i,5}\}, \quad \text{ahol } i = 1, 2, \dots, 5 \\ A &\subseteq T_1^5 \times T_2^5 \times T_3^5 \times T_4^5 \times T_5^5 \end{aligned}$$

Az elemhuszonötöst az átláthatóság kedvéért mátrix alakba rendezzük (úgy, ahogy a megoldásokat a 3.6. és 3.7. táblázatok foglalják össze). Ha $\langle a_1, a_2, \dots, a_n \rangle \in A$, és a

hozzá tartozó mátrix

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,5} \\ a_{2,1} & a_{2,2} & \dots & a_{2,5} \\ \vdots & \vdots & \ddots & \vdots \\ a_{5,1} & a_{5,2} & \dots & a_{5,5} \end{pmatrix}, \text{ ahol } a_{i,j} \in T_i,$$

akkor $a_{i,j} = a_{5 \cdot (i-1) + j}$. Bár a Descartes-szorzat elemszáma 5^{25} , a kényszerfeltételnek ennél jóval kevesebb állapot felel meg. Kényszerfeltételnek most azt tekintjük, hogy minden lehetséges érték pontosan egyszer szerepel a táblázatban. Így egy $a = (a_{i,j})_{5 \times 5}$ mátrix akkor eleme az A halmaznak, ha

$$\bigwedge_{i=1}^5 \bigwedge_{j=1}^5 \bigwedge_{k=1}^5 (i \neq j \supset a_{k,i} \neq a_{k,j}),$$

azaz a mátrix minden k . sorában tetszőleges két különböző i . és j . oszlop értéke különböző. A keresési tér ekkor összesen $(5!)^5$ elemet tartalmaz (ami még mindig közel 25 milliárd állapotot jelent), amelyek közül mindössze azt a kettőt tekintjük célállapotnak, ahol mind a 14 állítás teljesül.

A probléma megfogalmazására alkalmas $L^{(0)} = \langle \{\neg, \vee\}, Con, Form \rangle$ ítéletlogikai nyelv ítéleváltozóinak száma 125 (ahol mind a 25 tulajdonsághoz 5 ítéleváltozó tartozik):

$$Con = \{ p_{i,j,k} : i, j, k \in \{1, 2, \dots, 5\} \},$$

továbbá minden $a \in T_1^5 \times \dots \times T_5^5$ elemhuszonötöshöz a nyelv olyan ϱ_a interpretációját rendeljük, ahol

$$\varrho_a = \{ p_{i,j,k} : a_{i,j} = t_{i,k} \}.$$

Kényszerfeltétel

A nyelv kialakításából következően (ahogy azt a (3.2) és (3.3) pontokban megadtuk) az állapotot leíró mátrixokban minden i . sor és j . oszlop esetén az alábbi klózok kiegészítik a probléma leírását:

$$\{ p_{i,j,1} \vee p_{i,j,2} \vee \dots \vee p_{i,j,5} \} \cup \{ \neg p_{i,j,k} \vee \neg p_{i,j,\ell} : k, \ell \in \{1, \dots, 5\} \wedge k \neq \ell \},$$

ami szerint egy-egy tulajdonság egy-egy állapotban pontosan egy értéket vehet fel. Nagyon hasonlít ehhez az állapottérben megadott kényszerfeltétel átírása:

$$\{ p_{i,1,k} \vee p_{i,2,k} \vee \dots \vee p_{i,5,k} \} \cup \{ \neg p_{i,j,k} \vee \neg p_{i,\ell,k} : j, \ell \in \{1, \dots, 5\} \wedge j \neq \ell \},$$

ami szerint a mátrix tetszőleges i . sorában tetszőleges t_k érték pontosan egyszer szerepel. Az ítéletlogikai átírásból következően és a kényszerfeltétel átfogalmazásával is $525 -$ eddig így összesen $1050 -$ klózt kaptunk.

Célfeltétel

Célfeltételnek most a probléma leírását adó 14 állításban foglaltakat tekintjük. Mivel ezek nagyon hasonlóak egymáshoz és hasonlóképp formalizálhatjuk őket, most csak az első mondatra koncentrálunk:

„Az angol a piros házban lakik.”

Ennek megfelelően az állapottér-reprezentációban azt mondjuk, hogy minden $a = (a_{i,j})_{5 \times 5} \in A$ célállapot esetén igaz a következő formula:

$$\bigwedge_{i=1}^5 \left(a_{1,i} = \underbrace{t_{1,1}}_{\text{angol}} \supset a_{2,i} = \underbrace{t_{2,1}}_{\text{piros}} \right) \rightsquigarrow \bigwedge_{i=1}^5 (p_{1,i,1} \supset p_{2,i,1}).$$

Az állapottér-reprezentáció célfeltételéhez $L^{(0)}$ nyelvű formulát készítünk (a korábban bemutatott módszer szerint), amit eztán már csak klózhalmazzá (konjunktív normálformává) kell alakítani. Hasonló logikával összesen 8 mondatot készíthetünk:

1. $\{ \neg p_{1,i,1} \vee p_{2,i,1} : i \in \{1, 2, \dots, 5\} \}$
2. $\{ \neg p_{2,i,2} \vee p_{5,i,1} : i \in \{1, 2, \dots, 5\} \}$
3. $\{ \neg p_{4,i,1} \vee p_{1,i,2} : i \in \{1, 2, \dots, 5\} \}$
4. $\{ \neg p_{2,i,3} \vee p_{4,i,2} : i \in \{1, 2, \dots, 5\} \}$
6. $\{ \neg p_{3,i,1} \vee p_{5,i,2} : i \in \{1, 2, \dots, 5\} \}$
7. $\{ \neg p_{5,i,3} \vee p_{1,i,4} : i \in \{1, 2, \dots, 5\} \}$
12. $\{ \neg p_{3,i,4} \vee p_{4,i,4} : i \in \{1, 2, \dots, 5\} \}$
13. $\{ \neg p_{2,i,5} \vee p_{3,i,5} : i \in \{1, 2, \dots, 5\} \}$

További négy állítás formalizálásának eredménye is egymáshoz hasonló (csak a felhasznált ítéletváltozók indexeiben különböző) klózhalmaz. A tömörebb leírás végett használjuk az I jelölést az $\{1, 2, \dots, 5\}$ halmazra.

5. $\left\{ \neg p_{1,i,2} \vee \left(\bigvee_{j \in J} p_{1,j,3} \right) : (i \in I) \wedge (J = \{i-1, i+1\} \cap I) \right\}$
10. $\left\{ \neg p_{3,i,3} \vee \left(\bigvee_{j \in J} p_{5,j,3} \right) : (i \in I) \wedge (J = \{i-1, i+1\} \cap I) \right\}$
11. $\left\{ \neg p_{3,i,2} \vee \left(\bigvee_{j \in J} p_{5,j,4} \right) : (i \in I) \wedge (J = \{i-1, i+1\} \cap I) \right\}$
14. $\left\{ \neg p_{1,i,5} \vee \left(\bigvee_{j \in J} p_{2,j,4} \right) : (i \in I) \wedge (J = \{i-1, i+1\} \cap I) \right\}$

Végezetül a 8-as és 9-es mondatok felírhatóak egyetlen ítéletváltozó segítségével (egyelemű klózhalmazként) az alábbi alakban:

8. $\{p_{4,3,3}\}$

9. $\{p_{2,1,4}\}$

Az elkészült formulák jól mutatják, hogy a 14 állítás három csoportba sorolható, amelyek formalizált alakja csupán az ítéletváltozók indexeiben tér el egymástól. A 8-as és 9-es mondatok kivételével minden állítást 5-5 klóz segítségével írhatunk le, így összesen 62 klózzal formalizálhattuk a célfeltételt, és mindösszesen 1112 klózzal és 125 változóval a teljes SAT problémát.

A SAT probléma leírását DIMACS formátumban állítottuk elő, így a reprezentációt meglévő SAT solver implementációk felhasználásával is ellenőrizni tudtuk. Tekintettel arra, hogy számunkra valójában a SAT probléma nemcsak a klózhalmaz kielégíthetőségének eldöntését jelentette (a kielégíthetőség bizonyítására egyetlen modellt elég megadni), hanem minden megoldás előállítását (a konjunktív normálformából diszjunktív normálforma készítését), ezért a feladat megoldására saját implementáció készült.

A részproblémák

A probléma leírását adó klózhalmazt (3.6) szerint 25 klózhalmazra bontva állíthatjuk elő a részproblémák leírásait. A *zebra probléma* esetében az alábbiakat tapasztaltuk:

1. 69-74 klóz alkot egy-egy részproblémát, amelyekben
2. 28-33 különböző ítéletváltozó szerepel,
3. 808-38 088 darab elemi konjunkció alkotja a megoldást, melyeket
4. 1-5 elemi konjunkcióra redukál a közelítés.

Az első két pont jól mutatja, hogy a részproblémák (mind a klózhalmaz elemszáma, mind a változók darabszáma tekintetében) lényegesen egyszerűbbek az eredetnél. Míg az utóbbi két pontból érzékelhetjük, hogy miért is van szükség az approximációra.

Megoldásjelöltek és megoldások

Az elemi konjunkciók közös modelljeinek megkeresésére az algoritmusban egyszerű visszalépéses keresőt alkalmaztunk. Tekintettel arra, hogy a részproblémák megoldásait felülről közelítettük, a keresés eredményeként olyan interpretációk is keletkezhetnek, amelyek nem megoldásai az eredeti feladatnak. A konkrét példában ez 26 modellt (modelljelöltet) eredményezett, amelyből 2 darab valódi megoldást kaptunk.

3.3.8. Az n -vezér probléma közelítő megoldása

Az n -vezér probléma állapottér-reprezentációi közül kiindulópontnak a probléma állapotainak azon ábrázolását választjuk, ahol egyetlen vektor segítségével jelöljük, hogy

a tábla egy-egy sorában melyik az az oszlop, ami a vezért tartalmazza (tulajdonképpen n darab egyforma jellemző határoz meg egy állapotot):

$$\begin{aligned} T_i &= \{1, 2, \dots, n\}, \quad \text{ahol } i = 1, 2, \dots, n \\ A &= T_1 \times T_2 \times \dots \times T_n \end{aligned}$$

A probléma megfogalmazására alkalmas $L^{(0)} = \langle \{\neg, \vee\}, Con, Form \rangle$ ítéletlogikai nyelv ítéletváltozóinak száma $n \cdot n$ (ahol n tulajdonsághoz egyenként n lehetséges érték tartozik):

$$Con = \{ p_{i,j} : i, j \in \{1, 2, \dots, n\} \},$$

továbbá minden $a = (a_1, a_2, \dots, a_n) \in A$ állapothoz a nyelv egy olyan ϱ_a interpretációját rendeljük, ahol

$$\varrho_a = \{ p_{i,j} : a_i = j \}.$$

Kényszerfeltétel

Az állapottéren ez esetben kényszerfeltétel nem került definiálásra. A klózhalmazba így mindösszesen a (3.2) és (3.3) pontból következő feltételeket kell megjeleníteni. Így minden i . tulajdonság esetén:

$$\{p_{i,1} \vee \dots \vee p_{i,n}\} \cup \{ \neg p_{i,j} \vee \neg p_{i,k} : j, k \in \{1, \dots, n\} \wedge j \neq k \}$$

Célfeltétel

A célfeltétel – miszerint az egyes sorok vezérei különböző oszlopban és átlóban állnak – állapotokon megfogalmazott alakját

$$\bigwedge_{i=1}^n \bigwedge_{j=1}^n (i \neq j \supset a_i \neq a_j \wedge |i - j| \neq |a_i - a_j|)$$

az $L^{(0)}$ nyelv következő klózaival helyettesíthetjük:

$$\{ \neg p_{i,j} \vee \neg p_{k,\ell} : i, j, k, \ell \in \{1, \dots, n\} \wedge (i \neq k) \wedge (j = \ell \vee |i - k| = |j - \ell|) \}.$$

A 4-vezér probléma esetén így egy 16 ítéletváltozóból álló nyelv és egy 80 elemű klózhalmaz alakul ki.

Részproblémák

A 8-vezér probléma leírását adó klózhalmazt (3.6) szerint 8 klózhalmazra bontva állíthatjuk elő a részproblémák leírásait a mindösszesen 736 darab és összesen 64 ítéletváltozóból álló klózból.

1. 141-165 klóz alkot egy-egy részproblémát, amelyekben
2. 64 különböző ítéletváltozó szerepel,
3. 28 052-155 132 darab elemi konjunkció alkotja a megoldást, melyeket
4. 8 elemi konjunkcióra redukál a közelítés.

Sajnos a részproblémákban nem csökkent a változók száma, csak a klózoké.

4. fejezet

Összefoglaló

A dolgozatban elsőként egy, a bevezetőben ismertetett parciális rendszer vizsgálatát folytattuk le. A kibővített elsőrendű logikai nyelv rendkívül bő eszköztárat kínált fel, így a feladat egyszerűsítése végett szűkített elsőrendű nyelv tulajdonságainak feltárásával kezdtünk. Már a szűkített elsőrendű nyelven felírható formulák esetében és Pawlak-féle approximációs teret választva is megmutatkoztak a szemantika sajátosságai. Megmutattuk, hogy ha egy kielégíthetetlen formulahalmaz formuláit eltérő mondatfunktorokkal egészítjük ki, az megváltoztathatja a formulahalmaz szemantikai tulajdonságait. Így például egyszerűen igazolható volt a modus ponens és modus tollens mintájára készített alábbi két állítás is: $\{\downarrow(A \supset B), \downarrow A\} \not\models_w B$, illetve $\{\downarrow(A \supset B), \downarrow \neg B\} \not\models_w A$. De olyan egyszerű esetben is, mint $\{\downarrow A, \uparrow \neg A\}$ kielégíthetősége, hasonló eredményt kaptunk.

A szemantika bemutatására olyan alkalmazást készítettünk, ahol összehasonlítható a közelítés segítségével, illetve nélküle számított eredmény úgy, hogy az alkalmazás szabad kezdet ad a felhasználó számára az approximációs tér kialakításához. Az implementáció alapját egy relációs adatbázis adta, amelyben SQL függvények segítségével készítettük el a szemantika implementációját, tesztelve ezzel a gyakorlati felhasználási lehetőségeket.

Az alkalmazás segítségével egyszerű példával illusztráltuk, hogy a hiányzó háttértudásban rejlő bizonytalanság miként hat a formulák kiértékelésének eredményére, és esetenként hogyan mond ellent a klasszikus szemantika szerint elvárhatónak.

A következő lépésben Arisztotelész szillogizmusain keresztül vizsgáltuk az új szemantikát. Az már nem volt kérdés, hogy a parciális kétértékű logikában cáfolhatóak-e a szillogizmusok, az azonban igen, hogy az approximációs tér tulajdonságai meghatározzák-e ezen cáfolatok létét. Mivel még viszonylag kis elemszámú univerzum esetén is temérdek interpretáció és approximációs tér létezik, így a cáfolatok keresését egy erre a célra kialakított alkalmazásra bíztuk. Ennek segítségével hamar nyilvánvalóvá vált, hogy a cáfolhatóság okait nem az approximációs tér tulajdonságai közt kell keresni. Az alkalmazás minden vizsgált esetben előállította a szillogizmusok alapján készített következtetések cáfolatait.

A további vizsgálatok már azt a célt szolgálták, hogy megmutassuk, milyen formulák esetében várható el, hogy a közelítés segítségével számított igazságérték egybeesik

a klasszikus szemantika szerint számított igazságértékkel, majd javaslatot tettünk a szemantika átalakítására. Mindkét út arra vezetett, hogy megmutassuk, miként lehet felhasználni a közelített számítás értékét arra, hogy megjósolja a klasszikus esetben kapható eredményt.

Feltételezve, hogy egy mesterséges intelligencia feladatként felmerülő probléma állapottér-reprezentációja a korábban bemutatott szűkített elsőrendű logikai nyelven megfogalmazható, végignéztük a szokásos megoldáskereső algoritmusokat, és úgy formáltuk át őket, hogy ki tudják használni azt az előnyt, amit a közelítő számítás hozhat. Ennek kapcsán előbb a visszalépéses keresőt alakítottuk át, majd a keresőfával megoldást keresők algoritmusának közös vázát terjesztettük ki, így készítve elő a terepet új megoldáskereső algoritmusok számára. Olyan algoritmusokat készítettünk, amelyek – amennyiben erre lehetőség adódik – a közelítő számítás eredményére támaszkodnak, azaz a megoldáskereső algoritmusok abban az irányban tárják fel előbb az állapottérgráfot, amerre az egyszerűbben (gyorsabban) ellenőrizhető feltételek vezetnek.

Az approximáció nem csupán a formulákon, hanem az állapothalmazon is értelmezhető, amit arra tudtunk felhasználni, hogy bemutassuk (és formalizáljuk) egy-egy közismert probléma állapottér-reprezentációi közti kapcsolatot. Ugyan a bemutatott reprezentációk már ismertek voltak, így új reprezentáció nem született, de a reprezentációk közti kapcsolatot ki tudtuk fejezni a halmazapproximáció eszközével. Amennyiben a cél, hogy minél egyszerűbb (minél kisebb logikai összetettségű) feltételrendszert használva modellezzük a problémát, úgy az állapothalmaz felső közelítése nyújt segítséget, míg ha a problémareprezentáció redundanciáit akarjuk megszüntetni, az állapothalmaz alsó közelítése (így az állapottérgráf méretének csökkentése) kínál megoldást.

Véges és diszkrét állapotterek esetében a probléma leírásához állításlogika is elegendő. Amennyiben már rendelkezésre áll egy állapottér-reprezentáció, akkor abból az állításlogikai modell egyszerűen (viszonylag mechanikusan) elkészíthető. Az így kapott modellben megoldani a problémát nem jelent mást, mint megoldani egy állításlogikai SAT feladatot. A tapasztalat ugyanakkor az, hogy a szokásos SAT solver implementációk alulmaradnak az állapottérhez köthető megoldáskereső algoritmusokkal szemben. Az az információ, ami az állapottér-reprezentációban implicit benne volt, de az állításlogikai modellben elveszett, az a részproblémákra bontás révén csempészhető vissza a megoldáskeresésbe. A részproblémák megoldásain a halmazapproximáció eszközt felhasználva tudtuk tovább redukálni a feladatot. Ehhez előbb egy halmazmetszeteken alapuló approximációs teret állítottunk elő, és megvizsgáltuk a felső közelítés tulajdonságait, majd két példán keresztül mutattuk meg a kidolgozott új algoritmus működését.

Chapter 5

Summary

Rough set theory hasn't lost its actuality till now, despite it was founded by Pawlak at the beginning of the 1980s. The question arises over and over again how the available and continuously generated data are processed and how the inaccuracy coming from information representation or often from the lack of information is managed. Rough set theory provides a solution for managing the uncertainty resulting from the incomplete knowledge being at our disposal. In the evolving set-theoretic approach, the *membership relation* becomes uncertain. Besides objects certainly being members of some set and objects certainly not being members of the set, a new intermediate category appears. Uncertainty is generated by the limits of the available knowledge in such a way that some objects are indistinguishable from one another based on this limited knowledge.

In the last 30 years, more generalisations of the equivalence-relation-based Pawlakian system have been created [6, 34]. Some of them make it possible to use partial approximation spaces. This research is connected to the Faculty of Informatics of the University of Debrecen in several ways. As with first-order logic based on classical set theory, new partial logical systems were built that are based on rough sets and the partial approximation space [26, 23]. Partial logical system means truth-value-gapped logic, where the value gap (lack of truth value) represents partiality. One of the goals of this doctoral thesis to investigate this kind of logical systems. This research focuses on how much the results generated by the partial logical system resemble those generated by classical logic. The goal is to discover the peculiarities of the partial logical system and to determine what the benefits of the system based on approximative computations are and where the price is to be paid.

When solving different problems using artificial intelligence, the first step is to appropriately model the problem. The model often appears in the form of first-order-logic formulae, especially if the problem itself is described using state-space representation and we want to solve it using search algorithms based on state spaces. The question arises how search algorithms should be transformed if – during modelling – we would like to use the possibilities provided by the partial logical system.

In case we already have a discrete finite state space giving the description of the problem, we have the chance to transform it into a SAT problem in propositional

logic. From this point on, solving the problem means searching for a model in propositional logic, for which we already have numerous methods at our disposal. However, by transforming the problem, some information is lost, which is definitely disadvantageous during the process of solving the problem. In this case, rough set theory helps cope with this very drawback.

5.1 Logical Systems Based on Partial Approximation Space

We began the investigation of the partial first-order logical systems with the overview of tool-based partial first-order logic (TbPFoL) [21]. Approximation is done using base sets generated by our knowledge; these base sets are considered as tools. The system itself is very expressive and complex, thus a simplified logical language has been created as a first step, which was initially capable of investigating only propositional but later also simple first-order logical formulae and the conclusions made from them. In the simplified language, the approximation pair of the partial approximation space was used for defining the lower and upper approximations. We indicate the computation of the lower and upper approximations of truth values by $\downarrow$ and $\uparrow$ approximative sentence functors. In the simplified language, this merely means that when evaluating the truth values, we compute the truth values of atomic formulae according to the appropriate approximation.

The simplified language allows only one-argument predicate parameters. This makes the illustration of the lower and upper approximation rather easy, as we can see in Figure 5.1, using the positivity domain of the logical functions (functions with range $\{0, 1\}$).

Definition 5.1. *The positivity domain of a $\varphi : U \rightarrow \{0, 1, 2\}$ function is denoted by $[\varphi]^+$ and defined as the following set:*

$$[\varphi]^+ = \{ u : u \in U \text{ and } \varphi(u) = 1 \}.$$

Definition 5.2. *Let $\langle l, u \rangle$ be an approximation pair. The lower approximation of a $\varphi : U \rightarrow \{0, 1, 2\}$ function is a $\varphi^\downarrow : U \rightarrow \{0, 1, 2\}$ mapping such that*

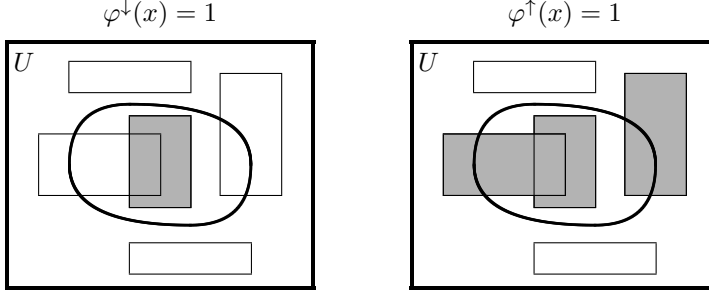
$$\varphi^\downarrow(u) = \begin{cases} 1 & \text{if } u \in l([\varphi]^+), \\ 0 & \text{if } u \in l(U \setminus u([\varphi]^+)), \\ 2 & \text{otherwise;} \end{cases}$$

Definition 5.3. *Let $\langle l, u \rangle$ be an approximation pair. The upper approximation of a $\varphi : U \rightarrow \{0, 1, 2\}$ function is a $\varphi^\uparrow : U \rightarrow \{0, 1, 2\}$ mapping such that*

$$\varphi^\uparrow(u) = \begin{cases} 1 & \text{if } u \in u([\varphi]^+), \\ 0 & \text{if } u \in l(U \setminus u([\varphi]^+)), \\ 2 & \text{otherwise.} \end{cases}$$

In case of logical functions, 2 represents the lack of truth value, while 0 and 1 represent *false* and *true*, respectively (as usual). The partiality of the logical system results from the lack of truth value. As a consequence of the system's partiality, logic with truth value gap is used.

Figure 5.1: The lower and upper approximation of an atomic formula



In Figure 5.1, the oval area in the middle represents the positivity domain of the logical function φ . The rectangles with thin edges show the base sets of the partial approximation space. Supposing that $\langle l, u \rangle$ is a Pawlakian approximation pair, the lower and upper approximation of the φ logical function is the area highlighted by dark background in the left and right image.

Although the expressive power of the simplified logical language is far behind that of the original system (defined in [23]), it is still capable of enlightening the peculiarities of the partial logical system. Each statement that is true in classical logic and refutable using the simplified language is also refutable in the original partial system.

We began the investigation of the partial first-order logical systems with the overview of propositional logic extended with one-argument predicate parameters and approximative sentence functors. By analysing the satisfiability of formula sets, we managed to show that we can generate situations where (among others) the *modus ponens* and *modus tollens* rules of inference do not hold.

We managed to prove the above statement by investigating the satisfiability of formula sets as the following two theorems show.

Theorem 5.4. *There exists an $A \in \text{Form}$ and a $B \in \text{Form}$ in an appropriate partial logical language such that $\{\downarrow A, \downarrow(A \supset B), \uparrow \neg B\}$ is satisfiable.*

Theorem 5.5. *There exists an $A \in \text{Form}$ and a $B \in \text{Form}$ in an appropriate partial logical language such that $\{\uparrow(A \supset B), \uparrow \neg B, \downarrow \neg A\}$ is satisfiable.*

Additionally, we also managed to show that, with premises and conclusion constructed from atomic formulae, *modus ponens* and *modus tollens* are valid, even if both lower and upper approximations are used together before the premises and the conclusion.

Theorem 5.6. *The $\{\uparrow P(x), \uparrow(P(x) \supset Q(x)), \downarrow \neg Q(x)\}$ set is unsatisfiable, supposing that $P, Q \in \mathcal{P}(1)$ are predicate parameters and $x \in \text{Var}$ is a variable.*

In order to discover the peculiarities of the partial logical system through a practical example, we have created an implementation of the semantics based on a relational database. In this case, the universe is given by the tuples of a single database table. The positivity domains of the one-argument predicates are given by subsets of the universe (i.e., by sets containing tuples). The semantic value of each predicate parameter is defined by a stored logical function. The domain of these functions is the set of tuples. The user can set up the tools giving the base sets of the approximation space using a custom collection of predicate parameters.

The approximation space is generated by the selection of tools. Assuming that the first k of the $P_1, \dots, P_n$ predicate parameters are selected, two matrices will be built:

$$A = (a_{i,j})_{k \times n} \quad \text{and} \quad B = (b_{i,j})_{k \times n}.$$

$$\begin{aligned} a_{i,j} &= \left| [\varrho(P_i)]^+ \cap [\varrho(P_j)]^+ \right| \\ b_{i,j} &= \left| [\varrho(P_i)]^+ \setminus [\varrho(P_j)]^+ \right| \end{aligned}$$

where $i = 1, \dots, k$ and $j = 1, \dots, n$; while $\varrho(P_i) = \varphi_i$ is the logical function that belongs to the i th predicate parameter. Using the matrices and the positivity domains of the tools, the values of the lower and upper approximations of the formulae can be determined without having to evaluate the stored logical functions:

$$\begin{aligned} \varphi_j^\downarrow(u) &= \begin{cases} 1 & \text{if there exists an } i \text{ such that } u \in [\varphi_i]^+ \text{ and } b_{i,j} = 0, \\ 0 & \text{if there exists an } i \text{ such that } u \in [\varphi_i]^+ \text{ and } a_{i,j} = 0, \\ & \text{and there is no } \ell \text{ such that } a_{\ell,i} > 0 \text{ and } a_{\ell,j} > 0, \\ 2 & \text{otherwise;} \end{cases} \\ \varphi_j^\uparrow(u) &= \begin{cases} 1 & \text{if there exists an } i \text{ such that } u \in [\varphi_i]^+ \text{ and } a_{i,j} > 0, \\ 0 & \text{if there exists an } i \text{ such that } u \in [\varphi_i]^+ \text{ and } b_{i,j} = 0, \\ & \text{and there is no } \ell \text{ such that } a_{\ell,i} > 0 \text{ and } a_{\ell,j} > 0, \\ 2 & \text{otherwise;} \end{cases} \end{aligned}$$

where $i, \ell \in \{1, \dots, k\}$ and $j \in \{1, \dots, n\}$. Using the application, we can compare the truth values of simple formulae in the classical and partial logical system.

5.1.1 Aristotle's Syllogisms

After this, we investigated the tool-based partial semantics via Aristotle's syllogisms. We chose these syllogisms because we thought that they represent the most frequent quantified statements, which can be formalised using single-level quantification. The investigation was based on the assumption that the lower approximation in the approximation space represents our certain knowledge, therefore it is worth to examine whether we can make right conclusions based on this knowledge. As syllogisms are valid statements in classical first-order logic (in some cases, assuming the necessary

existential preconditions), the subject of our next investigation was to test the validity in the partial logic. It was clear from the previous results that we were able to refute the syllogisms, so the goal of the investigation was to create the refutations using approximation spaces and interpretations with different properties. The considered cases:

1. there exist at most 3 different base sets ($|\mathfrak{B}| \leq 3$);
2. there exist at most 2 different base sets ($|\mathfrak{B}| \leq 2$);
3. the union of base sets covers the universe;
4. the union of base sets does not cover the universe;
5. the base sets are pairwise disjoint;
6. $\left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^+ \neq \emptyset$ for all $P \in \text{Con}$, i.e., the positivity domain for each predicate parameter is not empty;
7. $\left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^- \neq \emptyset$ for all $P \in \text{Con}$, i.e., the negativity domain for each predicate parameter is not empty;
8. $\left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^+ \neq \emptyset$ and $\left[\llbracket P \rrbracket^{\langle U, \varrho \rangle} \right]^- \neq \emptyset$ for all $P \in \text{Con}$, i.e., neither the positivity nor the negativity domain is empty for each predicate parameter;
9. there is no such $T \in \mathfrak{T}$ tool for any $P \in \text{Con}$ such that $\varrho(P) = \varrho(T)$, i.e., the interpretations of the predicate parameters and the tools are different.

We achieved similar results in case of all 12 statements. In all 9 investigated cases, refutations could be constructed, only the number of refutations could be different. Table 5.1 summarises the results in case of *Barbara*, *Celarent*, *Darapti*, and *Felapton*. All other syllogisms showed the same results as one of the four mentioned.

To construct the refutations, a **Java** application was created, which made it possible to examine the more than 134 million interpretations. The results showed that the existence of refutations does not depend on the properties of the approximation space or the interpretation, but it comes from the semantics of logical connectives. Based on these results, we presented two possibilities for how the computation made with approximation can replace the classic method:

1. The lower-approximation-based computed value of some formulae coincides with the truth value in the classic sense, supposing that a truth value (*true* or *false*) resulted. To identify formulae with this property, a calculus was developed based on the syntactical structure of formulae.
2. The semantics of the logical connectives of the partial logical system can be transformed so that the value of the approximate computation coincides with the truth value in the classic sense, supposing that a truth value resulted. This is called *pessimistic* semantics. One of its drawbacks is that the resulting semantics does not conform to the requirements of logical systems with truth value gap. The other drawback is that the number of formulae having no truth value after evaluation increases.

Table 5.1: Number of refutations if $|U| = 4$

	$\langle U, \varrho \rangle$	<i>Barbara</i>	<i>Celarent</i>	<i>Darapti</i>	<i>Felapton</i>
$ \mathfrak{B} \leq 3$	2,355,200	4,728	9,576	22,368	18,816
$ \mathfrak{B} \leq 2$	491,520	912	912	3,024	3,024
$\bigcup \mathfrak{B} = U$	132,288,512	117,696	219,936	253,212	314,004
$\bigcup \mathfrak{B} \neq U$	1,925,120	3,840	7,296	11,328	11,328
disjoint $\mathfrak{T}$	208,896	1,104	1,104	3,456	3,456
$[\varrho(P)]^+ \neq \emptyset$	64,139,967	110,448	193,776	158,676	274,104
$[\varrho(P)]^- \neq \emptyset$	689,831	12,168	25,716	58,356	52,920
$[\varrho(P)]^\pm \neq \emptyset$	229,940	1,368	10,608	15,732	15,252
$\varrho(P) \neq \varrho(T)$	5,918,720	540	4,752	8,652	6,036
Total	134,213,632	121,536	227,232	264,540	325,332

5.2 Partial Approximation Space in Artificial Intelligence

We investigated the possible uses of the partial logical systems based on partial approximation space in the field of artificial intelligence in connection with the search algorithms defined on state space. The reason for this is that in the description of problems modelled using state-space representation, first-order logic plays a distinguished role.

Definition 5.7. *The state-space representation of a problem p is defined as an ordered 4-tuple of the form $\langle \mathcal{A}, a_0, \mathcal{C}, \mathcal{O} \rangle$ [7], where*

1. $\mathcal{A}$ is a nonempty set of states;
2. $a_0 \in \mathcal{A}$ is the initial state of the problem;
3. $\mathcal{C} \subset \mathcal{A}$ is the set of goal states;
4. $\mathcal{O}$ is a nonempty set containing the operators of the state space, where for all $o \in \mathcal{O}$, o is a mapping such that $\text{dom}(o) \subseteq \mathcal{A}$ and $\text{rng}(o) \subseteq \mathcal{A}$.

During the investigation, we assumed that in the state-space representation of the problem, the preconditions of operator applications and the state conditions can (at least in part) be formalised with the partial logic language introduced earlier.

5.2.1 Backtracking and Searching with Tree

First we focused on the preconditions of operator applications. These preconditions select those states from $\mathcal{A}$ to which a given $o \in \mathcal{O}$ can be applied, i.e., we define $\text{dom}(o) \subseteq \mathcal{A}$. To approximate the set $\text{dom}(o)$, we introduce the following sets:

1. $\text{dom}^\downarrow(o) \subseteq \mathcal{A}$: a set containing states to which o can be *certainly applied*;
2. $\text{dom}^\uparrow(o) \subseteq \mathcal{A}$: a set containing states to which o *may be applied*.

This means that $\text{dom}^\downarrow(o) \subseteq \text{dom}(o) \subseteq \text{dom}^\uparrow(o) \subseteq \mathcal{A}$.

Using the lower and upper approximations of the operators' domains, we can introduce for a given state $a \in \mathcal{A}$

1. the set of operators that haven't yet been applied but can certainly be applied, initially:

$$\left\{ o \in \mathcal{O} : a \in \text{dom}^\downarrow(o) \right\};$$

2. the set of operators that haven't yet been applied but may be applied, initially:

$$\left\{ o \in \mathcal{O} : a \in \left(\text{dom}^\uparrow(o) \setminus \text{dom}^\downarrow(o) \right) \right\}.$$

The backtrack algorithms can then be transformed so that they discover the state-space graph preferring the certainly applicable operators until it is possible. The evaluation of the preconditions of operator applications in the classic (crisp) way can be deferred until we run out of the certainly applicable operators for the given state. The approximation controls the search such as heuristics do.

It is a frequently used technique that the set of (real) states $\mathcal{A}$ is considered as a subset of a larger set $\mathcal{T} = T_1 \times \dots \times T_n$, where the elements of $\mathcal{A}$ are selected from the elements of $\mathcal{T}$ using the state conditions. In this case, the states of a problem are described by n properties, where the possible values of property i come from T_i . Usually, state conditions are necessary because the properties are not independent from one another, not all combinations of them form real states. In classic search algorithms, state conditions do not have to be implemented, but the preconditions of operator applications do; therefore, whatever restrictions we can do, we prefer to do it in the state conditions.

Definition 5.8. *The function $f : \mathcal{T} \rightarrow \mathcal{T}$ is the generalisation of the operator $o \in \mathcal{O}$, if $o(a) = f(a)$ for all $a \in \text{dom}(o)$ and*

$$\text{dom}(o) = \left\{ a : a \in \left(\widehat{\text{dom}}(f) \cap \mathcal{A} \right) \wedge f(a) \in \mathcal{A} \right\};$$

where the set $\widehat{\text{dom}}(f) \subseteq \mathcal{T}$ represents those states which satisfy the precondition of f .

When using the generalised operators, application preconditions being independent from the state conditions are included in $\widehat{\text{dom}}(f)$. The generalised operators make it possible to extend the concept of accessibility.

Definition 5.9. *The state $a' \in \mathcal{T}$ is certainly directly accessible from $a \in \mathcal{A}$ if there exists a generalised operator f such that $a \in \text{dom}(f)$ and $a' = f(a)$, moreover,*

$$\left[a \in \widehat{\text{dom}}(f) \wedge f(a) \in \mathcal{A} \right] = 1,$$

that is, according to pessimistic semantics, the application precondition is satisfied in state a , and the state condition is satisfied in $a' = f(a)$.

Definition 5.10. *The state $a' \in \mathcal{T}$ is possibly directly accessible from $a \in \mathcal{A}$ if there exists a generalised operator f such that $a \in \text{dom}(f)$ and $a' = f(a)$, moreover,*

$$\left[a \in \widehat{\text{dom}}(f) \wedge f(a) \in \mathcal{A} \right] = 2,$$

that is, according to pessimistic semantics, it cannot be determined whether both the application precondition is satisfied in state a and the state condition is satisfied in $a' = f(a)$.

Nodes stored during the search are given open or closed label by tree search algorithms (breadth-first search, depth-first search, uniform-cost search etc.). Open nodes can be divided into certainly open and possibly open categories depending on whether they are certainly directly accessible or possibly directly accessible from the parent node. As with backtracking algorithms, tree search algorithms have also been transformed so that they prefer certainly open nodes during the search.

5.2.2 Approximated Solutions Using Finite and Discrete State Spaces

In case of finite and discrete state spaces, propositional logic is sufficient to describe the problem. If we already have a state-space representation at our disposal, a model in propositional logic can be simply created. Solving the problem with the resulting model means solving a SAT problem in propositional logic. However, according to our experiences, the performance of the usual SAT solver implementations falls below that of search algorithms related to state spaces. The information implicitly involved in the state-space representation but missing from the model in propositional logic can be included in the search by dividing the problem into subproblems. In this case, the steps of the search are the following:

- (1) Construct the subproblems as subsets of the clause set describing the problem.
- (2) Solve the subproblems independently. If any subproblem (as a clause set) is unsatisfiable, then the problem is not solvable.
- (3) Find the common solution of the subproblems (e.g., using a simple backtracking algorithm).

A serious problem arises regarding the above algorithm. Depending on how we divide the problem into subproblems, the number of solutions (as sets of interpretations) may be much greater than the number of solutions of the problem. Because

of this, finding the common solutions of the subproblems may become much more difficult.

Set approximation may give us a helping hand to simplify the description of solutions.

Definition 5.11. Let $L^{(0)} = \langle LC, Con, Form \rangle$ be a language in propositional logic and $\mathfrak{A}$ be the set of disjoint set pairs formed by the elements of the *Con* set:

$$\mathfrak{A} = \{ \langle P, Q \rangle : P \subseteq Con \wedge Q \subseteq Con \wedge P \cap Q = \emptyset \}.$$

Then, each $\langle P, Q \rangle \in \mathfrak{A}$ defines an interpretation family, denoted by $\mathcal{M}_{\langle P, Q \rangle}$, containing interpretations in which propositional parameters in P are true, while propositional parameters in Q are false. Formally:

$$\mathcal{M}_{\langle P, Q \rangle} = \{ \varrho : \varrho \subseteq Con \wedge P \subseteq \varrho \wedge Q \cap \varrho = \emptyset \}.$$

The solution of a subproblem is a disjunctive normal form, in which each elementary conjunction can be associated with an interpretation family. This way, the solution of a subproblem is given in the form of an interpretation family.

Definition 5.12. Let $\langle P, Q \rangle \in \mathfrak{A}$ and $\langle P', Q' \rangle \in \mathfrak{A}$ be two arbitrary set pairs. Then, let us define the operation $\sqcup$ as follows:

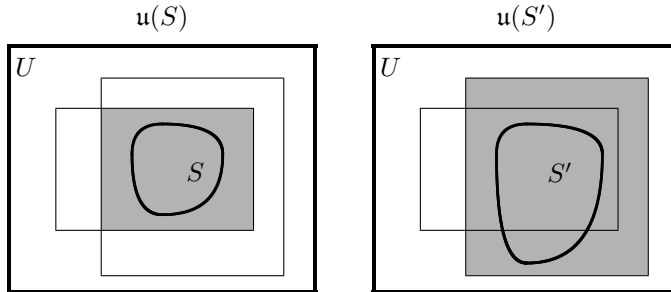
$$\langle P, Q \rangle \sqcup \langle P', Q' \rangle \stackrel{\text{def}}{=} \langle P \cap P', Q \cap Q' \rangle.$$

Theorem 5.13. Let $\langle P, Q \rangle \in \mathfrak{A}$ and $\langle P', Q' \rangle \in \mathfrak{A}$ be two arbitrary set pairs. Then, $\langle P, Q \rangle \sqcup \langle P', Q' \rangle \in \mathfrak{A}$, and

$$\mathcal{M}_{\langle P, Q \rangle \sqcup \langle P', Q' \rangle} \supseteq \mathcal{M}_{\langle P, Q \rangle} \cup \mathcal{M}_{\langle P', Q' \rangle}.$$

The $\sqcup$ -union of two interpretation families is constructed with the help of the intersection of the members of set pairs describing the interpretation families. The set of interpretations that belong to the unified interpretation family is the upper approximation of the union of interpretation sets that belong to the interpretation families. There comes the idea that set-intersection-based upper approximation simplifies the description of solutions of the subproblems.

Figure 5.2: Illustration of set-intersection-based upper approximation



Using upper approximation for the solutions of the subproblems, we can transform the steps of the search algorithm as follows:

- (1) Construct the subproblems as subsets of the clause set describing the problem.
- (2) Solve the subproblems independently. If any subproblem (as a clause set) is unsatisfiable, then the problem is not solvable.
- (3) Replace the solutions of the subproblems with their upper approximations.
- (4) Find the common solution of the subproblems (e.g., using a simple backtracking algorithm). As now we are working with the approximate solutions, only solution candidates will appear instead of real solutions.
- (5) Select the real solutions from the solution candidates by checking the state conditions and the goal condition.

The revised algorithm and the set-intersection-based upper approximation was illustrated via two well-known problems: the *zebra problem* and the *8-queens problem*. The results showed that, using this method, the SAT problem could be solved in nearly the same time as with the search algorithms based on the original state-space representation.

Do not forget, however, that a given problem may have a number of different state-space representations (which are all excellent in different aspects). It is the modeller's task to select the appropriate search algorithm or create the most appropriate model for the selected algorithm, taking into account the properties of the state-space graph.

5.3 Conclusion

In the partial approximation space, which was created as a generalisation of the Pawlakian system, the basis of approximate computations is the acquiescence of the uncertainty coming from the limitations of our available knowledge. The partial first-order logic, which includes the approximate computation in its semantics (with all its presented pros and cons), makes us possible to form conclusions based on this knowledge being at our disposal (or maybe the quickly accessible knowledge). All this can be introduced into the world of state-space representations via the formulae contained in the representation, whether it is about the extension of search algorithms or about solving a propositional SAT problem defined by the state-space representation.

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, dr. Mihálydeák Tamásnak mindazon szakmai iránymutatásért és támogatásért, amely döntő szerepet játszott a dolgozat létrejöttében. Köszönöm dr. Várterész Magdának, hogy az egyetemi tanulmányaim megkezdése óta folyamatosan figyelemmel kísérte és számtalan módon segítette előrehaladásomat. Szintén köszönettel tartozom kollégáimnak, kiemelten dr. Pánovics Jánosnak, továbbá családomnak a dolgozat végső formába öntéséhez nyújtott felbecsülhetetlen segítségért.

A. Függelék

Halmazelméleti jelölések

A következő függelék célja, hogy egyértelműsítse a dolgozatban használt jelöléseket. A jelölések megválasztásakor (zömében) a [27]-ben található azonos című függelékben választott jelöléseket vettük alapul.

Halmazok

Egy *halmaz* alatt objektumok kollekcióját vagy gyűjteményét értjük. Azt, hogy egy x objektum *elem*e az S halmaznak, $x \in S$ -sel jelöljük, míg ennek ellenkezőjét $x \notin S$ -sel. Egy halmazt megadhatunk elemeinek felsorolásával, kapcsos zárójelben, az elemeket vesszővel elválasztva. Így például az oktális számjegyek halmaza $\{0, 1, 2, 3, 4, 5, 6, 7\}$, amelyet esetleg $\{0, 1, \dots, 7\}$ alakban rövidíthetünk. Az üres halmazt $\emptyset$ jelöli. A természetes számok halmazát $\mathbb{N}$ -nel jelöljük és az $\{1, 2, \dots\}$ halmazt értjük alatta, az egész számok halmaza esetében a $\mathbb{Z}$ jelölést használjuk. A halmazokat megadhatjuk feltétel segítségével is, például az oktális számjegyek halmaza $\{x : x \in \mathbb{Z} \text{ és } 0 \leq x < 8\}$ alakban is felírható.

Az S_1 halmazt az S_2 halmaz *részalmazának* mondjuk, és ezt $S_1 \subseteq S_2$ formában jelöljük, ha S_1 minden eleme az S_2 halmaznak is eleme. Az S_1 és S_2 halmazok *egyenlőek* ($S_1 = S_2$), ha $S_1 \subseteq S_2$ és $S_2 \subseteq S_1$. Az S_1 halmaz az S_2 halmaz *valódi részalmaz*a, és ezt $S_1 \subset S_2$ formában jelöljük, ha $S_1 \subseteq S_2$, de $S_1 \neq S_2$. Egy véges S halmaz elemszáma $|S|$.

Az S_1 és S_2 halmazok *metszetét*, *unióját* és *különbségét* rendre $S_1 \cap S_2$, $S_1 \cup S_2$ és $S_1 \setminus S_2$ jelöli. Az S halmaz *hatványhalmazára* a 2^S jelölést alkalmazzuk. A rögzített U alaphalmaz mellett egy $S \subseteq U$ halmaz komplementere $S^c = U \setminus S$. Az $S_1, S_2, \dots, S_n$ halmazok *Descartes-szorzatát* $S_1 \times S_2 \times \dots \times S_n$ -nel jelöljük, és azt a halmazt értjük alatta, amely az összes olyan $\langle a_1, a_2, \dots, a_n \rangle$ alakú rendezett n -est tartalmazza, ahol $a_i \in S_i$, minden $i \in \{1, \dots, n\}$ esetén. Az U^n jelöléssel ($n \in \mathbb{N}$) az U halmaz n -szeres önmagával vett Descartes-szorzatát rövidítjük.

Binér relációk

Az $R \subseteq S_1 \times S_2$ halmazt $S_1 \times S_2$ feletti *binér relációnak* hívjuk. Az R binér reláció *értelmezési tartománya* a $\text{dom}(R)$ -rel jelölt $\{x : \langle x, y \rangle \in R\}$ halmaz, *értékkészlete* pedig a $\text{rng}(R)$ -rel jelölt $\{y : \langle x, y \rangle \in R\}$ halmaz. Az R binér reláció *balról teljes*, ha $\text{dom}(R) = S_1$. Az R binér reláció *funkcionális*, ha minden $\langle x, y \rangle \in R$ és $\langle x, z \rangle \in R$ esetén $y = z$.

Függvények

Egy $f \subseteq S_1 \times S_2$ funkcionális és balról teljes binér relációt *függvénynek* (vagy *leképezésnek*) nevezünk, valamint azt mondjuk, hogy az f függvény az S_1 halmazt képezi az S_2 halmazra. Jelölése: $f : S_1 \rightarrow S_2$. Ha f funkcionális, de nem balról teljes, akkor *parciális függvénynek* hívjuk. V^U -val azon függvények halmazát jelöljük, amelyek az U halmazt képezik V -re, valamint $f \in V^U$ akkor és csak akkor teljesül, ha $f : U \rightarrow V$ leképezés. Az $f : S_2 \rightarrow S_3$ és $g : S_1 \rightarrow S_2$ *függvények kompozícióját* $f \circ g$ jelöli, ahol $f \circ g : S_1 \rightarrow S_3$ leképezés olyan, hogy minden $x \in S_1$ esetén $(f \circ g)(x) = f(g(x))$.

Logikai függvény alatt a $\{0, 1\}$ halmazba képező függvényt értünk. 0-val a *hamis*, míg 1-gyel az *igaz* igazságértéket jelöljük. Az U halmazt leképező logikai függvények halmaza ezért $\{0, 1\}^U$. Minden f parciális logikai függvényt helyettesíthetünk a $\{0, 1, 2\}$ halmazba képező, következésképpen definiált f' (totális) függvénnyel:

$$f'(u) = \begin{cases} f(u) & \text{ha } u \in \text{dom}(f), \\ 2 & \text{egyébként;} \end{cases}$$

ahol a 2-es az értékhiány reprezentálására szolgál (a 2-est nem is tekintjük igazságértéknek). Ennél fogva a $\{0, 1, 2\}^U$ halmazba tartozó függvényeket fogjuk felhasználni az U halmazt leképező parciális logikai függvények ábrázolására.

B. Függelék

A keresőfával kereső működésének illusztrációja

A 3.2. ábrán található algoritmus illusztrálására egy fiktív $\langle \mathcal{A}, s_0, \mathcal{C}, \mathcal{O} \rangle$ állapottér-reprezentációt használunk fel, ahol

- $\mathcal{A} = \{s_0, s_1, s_2, s_3, s_6\}$,
- $\mathcal{C} = \emptyset$, és
- $\mathcal{O} = \{o_1, o_2, o_3, o_4\}$.

A célállapotok halmazának azért választjuk most az üres halmazt, hogy így a keresés során a – kezdőállapotból elérhető – teljes állapottérgráf bejárásra kerüljön.

Az állapotok halmazát a $\mathcal{T} = \mathcal{A} \cup \{s_4, s_5\}$ halmaz terjeszti ki, ahol s_4 és s_5 nem valódi állapotok. Az operátorok definícióját az alábbi három táblázat segítségével adjuk meg:

<i>Tényleges</i>					<i>Bizonyos</i>					<i>Lehetséges</i>				
	o_1	o_2	o_3	o_4		o_1	o_2	o_3	o_4		o_1	o_2	o_3	o_4
s_0	s_2	.	s_3	.	s_0	s_2	.	.	.	s_0	s_2	s_1	s_3	s_5
s_1	.	.	s_3	.	s_1	.	.	s_3	.	s_1	.	.	s_3	s_4
s_2	s_1	s_6	.	.	s_2	s_1	.	.	.	s_2	s_1	s_6	.	.
s_3	.	.	.	.	s_3	.	.	.	.	s_3	.	.	.	.
s_6	.	.	.	.	s_6	.	.	.	.	s_6	.	.	.	.

Az első táblázat s_0 címkéjű sorában az o_1 operátorhoz tartozó oszlopban az s_2 annyit jelent, hogy $s_0 \in \text{dom}(o_1)$ és $o_1(s_0) = s_2$, míg az o_4 operátorhoz tartozó oszlopból az olvasható le, hogy $s_0 \notin \text{dom}(o_4)$. A következő két hasonló felépítésű táblázat azt mutatja meg, hogy melyek a bizonyosan alkalmazható és melyek a lehetségesen alkalmazható operátorok, továbbá leolvasható az operátoralkalmazások eredménye is. Mivel $s_4 \notin \mathcal{A}$ és $s_5 \notin \mathcal{A}$, így ezek az első két táblázatban operátoralkalmazás eredményeként nem állhatnak elő. Míg s_4 és s_5 az eredeti állapottérgráfban nem létező csúcsok, a harmadik táblázatban található o_2 alkalmazása s_0 -ra egy nem létező él.

Az algoritmus működését a tekintett állapotér-reprezentációt felhasználva a következő táblázatban követhetjük nyomon. A táblázat minden sora a kiterjesztés előtti állapotot mutatja, ahol a kiterjesztésre kiválasztott nyílt csúcsot kiemeltük. Egy-egy cella megmutatja, hogy az oszlophoz tartozó állapotot tároljuk-e a kereső adatbázisában, és ha igen, akkor leolvasható, hogy melyik állapotból melyik operátor segítségével kaptuk, és az is, hogy milyen címkével láttuk el.

Nº	s_0	s_1	s_2	s_3	s_4	s_5	s_6
1. kezdő							
1.	$\emptyset$						
2.	zárt $\emptyset$	lehetséges s_0, o_2	bizonyos s_0, o_1	lehetséges s_0, o_3		lehetséges s_0, o_4	
3.	zárt $\emptyset$	bizonyos s_2, o_1	zárt s_0, o_1	lehetséges s_0, o_3		lehetséges s_0, o_4	lehetséges s_2, o_2
4.	zárt $\emptyset$	zárt s_2, o_1	zárt s_0, o_1	bizonyos s_1, o_3	lehetséges s_1, o_4	lehetséges s_0, o_4	lehetséges s_2, o_2
5.	zárt $\emptyset$	zárt s_2, o_1	zárt s_0, o_1	zárt s_1, o_3	lehetséges s_1, o_4	lehetséges s_0, o_4	lehetséges s_2, o_2
6.	zárt $\emptyset$	zárt s_2, o_1	zárt s_0, o_1	zárt s_1, o_3	lehetséges s_1, o_4		lehetséges s_2, o_2
7.	zárt $\emptyset$	zárt s_2, o_1	zárt s_0, o_1	zárt s_1, o_3	lehetséges s_1, o_4		zárt s_2, o_2
8.	zárt $\emptyset$	zárt s_2, o_1	zárt s_0, o_1	zárt s_1, o_3			zárt s_2, o_2

Ha több azonos címkéjű csúcs is kiválasztható lenne kiterjesztésre, akkor a szélességi kereső stratégiáját követve most azt választjuk, ami korábban jelent meg az adatbázisban (például az 5. sorban azért esik s_5 -re a választás, mert az már a második sorban – az első kiterjesztést követően – ismertté vált).

A keresés során két helyen történik frissítés (két csúcshoz találunk az eredetileg tárolt útnál jobbat), amelyek egyike sem feltételes frissítés: s_2 kiterjesztésével (2-es sorról 3-as sorra) s_1 kerül frissítésre, s_1 kiterjesztésével (3-as sorról 4-es sorra) s_3 kerül frissítésre. Egyik frissítés sem volt feltételes frissítés.

A keresés végére maradnak a lehetséges nyílt csúcsok, amelyek közül kettőt – s_4 -et és s_5 -öt – el kell távolítani az adatbázisból, mivel valójában nem vezet hozzájuk út az állapotérgráfban, sőt nem is csúcsai az állapotérgráfnak. Az s_0 -ból s_1 -be vezető o_2 operátorral címkézett él, amely valójában szintén nem része az állapotérgráfnak, s_1 frissítésével kerül ki a kereső adatbázisából. Az algoritmus minden zárt csúcshoz valódi – az eredeti állapotérgráfban szereplő – utat talál. Mivel nem volt célállapot, így a keresést a nyílt csúcsok elfogyása szakítja meg.

Irodalomjegyzék

- [1] CIUCCI, D. Orthopairs and granular computing. *Granular Computing* 1, 3 (Sep 2016), 159–170.
- [2] CIUCCI, D., MIHÁLYDEÁK, T., AND CSAJBÓK, Z. E. *On Definability and Approximations in Partial Approximation Spaces*. Springer International Publishing, Cham, 2014, pp. 15–26.
- [3] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., AND STEIN, C. *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.
- [4] CSAJBÓK, Z. E., AND MIHÁLYDEÁK, T. Partial approximative set theory: A generalization of the rough set theory. *International Journal of Computer Information System and Industrial Management Applications* 4 (2012), 437–444.
- [5] CSAJBÓK, Z. E., AND MIHÁLYDEÁK, T. *A General Set Theoretic Approximation Framework*. Springer, Berlin, Heidelberg, 2012, pp. 604–612.
- [6] DÜNTSCH, I., AND GEDIGA, G. *Approximation Operators in Qualitative Data Analysis*. Springer, Berlin, Heidelberg, 2003, pp. 214–230.
- [7] FUTÓ, I., Ed. *Mesterséges Intelligencia*. Aula Kiadó, Budapest, 1999.
- [8] GOMES, C. P., KAUTZ, H. A., SABHARWAL, A., AND SELMAN, B. Satisfiability solvers. In *Handbook of Knowledge Representation*, F. van Harmelen, V. Lifschitz, and B. Porter, Eds. Elsevier, Amsterdam, 2008, ch. 2, pp. 89–134.
- [9] JÄRVINEN, J., KONDO, M., MATTILA, J. K., AND RADELECZKI, S. Hidden modalities in algebras with negation and implication. *Math. Appl.* 2, 1 (2013), 5–20.
- [10] JÄRVINEN, J., PAGLIANI, P., AND RADELECZKI, S. Information completeness in Nelson algebras of rough sets induced by quasiorders. *Studia Logica* 101, 5 (2013), 1073–1092.
- [11] KÁDEK, T., AND MIHÁLYDEÁK, T. *Some Fundamental Laws of Partial First-Order Logic Based on Set Approximations*. Springer International Publishing, Cham, 2014, pp. 47–58.

- [12] KÁDEK, T., AND MIHÁLYDEÁK, T. On (in)validity of Aristotle's syllogisms relying on rough sets. *Annals of Computer Science and Information Systems* 7 (2015), 35–40.
- [13] KÁDEK, T., AND PÁNOVICS, J. Extended breadth-first search algorithm. *International Journal of Computer Science Issues* 10, 6 (2013), 78–82.
- [14] KÁDEK, T., KÓSA, M., AND PÁNOVICS, J. Klasszikus állapottér-reprezentációk tisztán matematikai logikai megközelítésben. In *SzámOkt XXVI. Nemzetközi Számítástechnika és Oktatás Konferencia* (Kolozsvár, Románia, 2016), K.-g. Biró and G. Sebestyén-Pál, Eds., Erdélyi Magyar Műszaki Tudományos Társaság (EMT), pp. 226–230.
- [15] KÁDEK, T., AND MIHÁLYDEÁK, T. Some unexpected properties of partial propositional logic based on partial approximation space. In *Proceedings of the 9th International Conference on Applied Informatics* (Eger, 2014), E. Kovács, G. Kusper, R. Kunkli, and T. Tómacs, Eds., vol. 1, pp. 51–58.
- [16] KÁDEK, T., AND MIHÁLYDEÁK, T. Az életlen halmazok elméletének néhány lehetséges alkalmazása a mesterséges intelligenciában. In *SzámOkt 2015 XXV. Nemzetközi Számítástechnika és Oktatás Konferencia* (Arad, Románia, 2015), K.-Á. Biró and G. Sebestyén-Pál, Eds., Erdélyi Magyar Műszaki Tudományos Társaság (EMT), pp. 251–256.
- [17] KÁDEK, T., ROBU, J., AND VÁRTERÉSZ, M. *Matematikai logika példatár*. Presa Universitara Clujeana, Cluj-Napoca, 2010.
- [18] KÓSA, M. *Korszerű információtechnológiai módszerek bevezetése a mesterséges intelligencia oktatásába*. PhD thesis, Matematika- és Számítástudományok Doktori Iskola, Debrecen, 2009.
- [19] MAIENBORN, C., VON HEUSINGER, K., AND PORTNER, P. *Semantics*. No. 3. k. in *Handbücher zur Sprach- und Kommunikationswissenschaft / Handbooks of Linguistics and Communication Science (HSK)*. De Gruyter, 2013.
- [20] MICHALEWICZ, Z., AND FOGEL, D. *How to Solve It: Modern Heuristics*. Springer, Berlin, Heidelberg, 2004.
- [21] MIHÁLYDEÁK, T. *Partial First-order Logic with Approximative Functors Based on Properties*. Springer, Berlin, Heidelberg, 2012, pp. 514–523.
- [22] MIHÁLYDEÁK, T. *Aristotle's Syllogisms in Logical Semantics Relying on Optimistic, Average and Pessimistic Membership Functions*. Springer International Publishing, Cham, 2014, pp. 59–70.
- [23] MIHÁLYDEÁK, T. Partial first-order logical semantics based on approximations of sets. *Non-classical Modal and Predicate Logics* (2011), 85–90.
- [24] MIHÁLYDEÁK, T. Generált szemantikai parcialitás kezelése az elsőrendű logikában. In *Nehogy érvgyűlölők legyünk: tanulmánykötet Máté András 60. születésnapjára*, Z. Zsófia and M. Péter, Eds. L'Harmattan K., 2013, pp. 193–204.

- [25] MIHÁLYDEÁK, T. Partial first-order logic relying on optimistic, pessimistic and average partial membership functions. In *Proceedings of the 8th conference of the European Society for Fuzzy Logic and Technology* (2013), G. Pasi, J. Montero, and D. Ciucci, Eds., pp. 334–339.
- [26] MIHÁLYDEÁK, T. First-order logic based on set approximation: A partial three-valued approach. In *2014 IEEE 44th International Symposium on Multiple-Valued Logic* (May 2014), pp. 132–137.
- [27] NILSSON, U., AND MAŁUSZYŃSKI, J. *Logic, programming and Prolog*. Wiley Chichester, 1990.
- [28] PAWLAK, Z. Rough sets. *International Journal of Computer & Information Sciences* 11, 5 (1982), 341–356.
- [29] PAWLAK, Z. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Theory and Decision Library D:. Springer Netherlands, 1991.
- [30] PÁNOVICS, J. *Appearance of Multiparadigm Programming Languages in the Teaching of Artificial Intelligence*. PhD thesis, Matematika- és Számítástudományok Doktori Iskola, Debrecen, 2013.
- [31] RUSSELL, S., AND NORVIG, P. *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall Press, Upper Saddle River, NJ, USA, 2009.
- [32] RUZSA, I., AND MÁTÉ, A. *Bevezetés a modern logikába*. Osiris Kiadó, Budapest, 1997.
- [33] SMITH, R. Aristotle’s logic. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., winter 2016 ed. Metaphysics Research Lab, Stanford University, 2016.
- [34] YAO, Y. Y. *On Generalizing Rough Set Theory*. Springer, Berlin, Heidelberg, 2003, pp. 44–51.

Publikációs lista

Referált közlemények

1. Adamkó, A., Kádek, T., Kollár, L., Kósa, M., Tóth, R.: *Cluster and discover services in the smart campus platform for online programming contests*. In: 6th IEEE Conference on Cognitive Infocommunications CogInfoCom 2015 : Proceedings, October 19–21, 2015, Széchenyi István University Győr, Hungary, IEEE, Danvers, 385–389, 2015.
Scopus EID: `eid=2-s2.0-84966641584`
2. Adamkó, A., Kádek, T., Kósa, M.: *Intelligent and adaptive services for a smart campus: Visions, concepts and applications*. In: 5th IEEE International Conference on Cognitive Infocommunications CogInfoCom 2014 : Proceedings, Vietri sul Mare, Italy / ed. Péter Baranyi, IEEE, Danvers, 505–509, 2014.
Scopus EID: `2-s2.0-84961968215`
3. Kádek, T., Mihálydeák, T.: *Some fundamental laws of partial first-order logic based on set approximations*. In: Rough Sets and Current Trends in Soft Computing : 9th International Conference, RSCTC 2014, Granada and Madrid, Spain, July 9–13, 2014, Proceedings / eds. Cornelis, C., Kryszkiewicz, M., Slezak, D., Menasalvas Ruiz, E., Bello, R., Shang, L, Springer, Cham, 47–58, 2014.
Scopus EID: `2-s2.0-84903761279`, io-port: `06349924`, dblp key: `conf/rsctc/KadekM14`
4. Szabó, R., Farkas, K., Ispány, M., Benczúr, A., Bátfa, N., Jeszenszky, P., Laki, S., Vágner, A., Kollár, L., Sidló, C., Besenczi, R., Smajda, M., Kövér, G., Szincsa, T., Kádek, T., Kósa, M., Adamkó, A., Lendák, I., Wiandt, B., Tomás, T., Nagy, A., Fehér, G.: *Framework for smart city applications based on participatory sensing*. In: 4th IEEE International Conference on Cognitive Infocommunications CogInfoCom 2013 : Proceedings, December 2–5, 2013 Budapest, Hungary / ed. Péter Baranyi, IEEE, Danvers, 295–300, 2013.
Scopus EID: `2-s2.0-84894178241`
5. Kádek, T., Pánovics, J.: *Extended breadth-first search algorithm*. Int. J. Comput. Sci. Issues 10 (6), 78–82., 2013.
dblp key: `journals/corr/KadekP14`
6. Kádek, T.: *Press-ready deduction trees in classical logic using point-plus-expressions*. Annales Univ. Sci. Budapest., Sect. Comp 37 229–238., 2012.
Zbl.: `1249.03003`, io-port: `06054976`

További közlemények

7. Tóth, R., Kádek, T., Kósa, M., Pánovics, J.: *Feladatajánlás döntéstámogató algoritmusokkal*. In: ENELKO 2016 XVII. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2016 XXVI. Nemzetközi Számítástechnika és Oktatás Konferencia, Kolozsvár, 2016. október 6–9. / szerk. Biró Károly-Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 300–305, 2016.
8. Kádek, T., Kósa, M., Pánovics, J.: *Klasszikus állapottér-reprezentációk tisztán matematikai logikai megközelítésben*. In: ENELKO 2016 XVII. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2016 XXVI. Nemzetközi Számítástechnika és Oktatás Konferencia, Kolozsvár, 2016. október 6–9. / szerk. Biró Károly-Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 226–230, 2016.
9. Kádek, T., Mihálydeák, T.: *On (in)validity of Aristotle's syllogisms relying on rough sets*. In: Proceedings of the LQMR 2015 Workshop / eds. Tomasz Lechowski, Przemysław Wałęga; Michał Zawidzki, Polskie Towarzystwo Informatyczne, Warszawa, 35–40, 2015.
10. Kádek, T., Mihálydeák, T.: *Az életlen halmazok elméletének néhány lehetséges alkalmazása a mesterséges intelligenciában*. In: ENELKO 2015 XVI. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2015 XXV. Nemzetközi Számítástechnika és Oktatás Konferencia, Arad, 2015. október 8–11. / szerk. Biró Károly-Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 251–256, 2015.
11. Adamkó, A., Kádek, T., Kollár, L., Kósa, M., Pánovics, J.: *New challenges in smart campus applications*. In: Recent Advances in Computer Science : Proceedings of the 19th International Conference on Computers (part of CSCC '15) / ed. Xiaodong Zhuang, Technical University of Sofia, Zakyntos Island, Greece, 44–50, 2015.
12. Fábián, R., Molnár, I., Kósa, M., Kádek, T., Pánovics, J.: *Hogyan születnek az intelligens tesztek a SmartCampusban?*. In: ENELKO 2015 XVI. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2015 XXV. Nemzetközi Számítástechnika és Oktatás Konferencia, Arad, 2015. október 8–11. / szerk. Biró Károly-Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 208–212, 2015.
13. Tóth, R., Kósa, M., Kádek, T., Pánovics, J.: *ProgContR: mentor alkalmazás a SmartCampusban*. In: ENELKO 2015 XVI. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2015 XXV. Nemzetközi Számítástechnika és Oktatás Konferencia, Arad, 2015. október 8–11. / szerk. Biró Károly-Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 328–333, 2015

14. Kádek, T., Pánovics, J.: *Extended breadth-first search algorithm in practice*. In: Proceedings of the 9th International Conference on Applied Informatics January 29 - Februar 1, 2014. Eger, Hungary Volume I. / eds. Emőd Kovács, Gábor Kusper, Roland Kunkli, Tibor Tómacs, Eszterházy Károly Főiskola, Eger, 59–66, 2014
15. Kádek, T., Mihálydeák, T.: *Some unexpected properties of partial propositional logic based on partial approximation space*. In: Proceedings of the 9th International Conference on Applied Informatics January 29 - Februar 1, 2014. Eger, Hungary Volume I. / eds. Emőd Kovács, Gábor Kusper, Roland Kunkli, Tibor Tómacs, Eszterházy Károly Főiskola, Eger, 51–58, 2014.
16. Kádek, T., Kósa, M., Pánovics, J.: *Programozási ismeretek mérése a ProgCont rendszerrel*. In: ENELKO 2014 XV. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2014 XXIV. Nemzetközi Számítástechnika és Oktatás Konferencia, Székelyudvarhely, 2014. október 9–12. / szerk. Biró Károly Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 188–191, 2014.
17. Kádek, T., Pánovics, J.: *Some improvements of the extended breadth-first search algorithm*. Studia Univ. Babes-Bolyai. Ser. Inf 59 (Special Issue), 165–173., 2014.
18. Kádek, T., Kósa, M.: *Intelligens campus Debrecenben*. In: ENELKO 2013 XIV. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2013 XXIII. Nemzetközi Számítástechnika és Oktatás Konferencia, Nagyszeben, 2013. október 10–13. / szerk. Biró Károly Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 210–213, 2013.
19. Kádek, T., Pánovics, J.: *Általános állapotter modell*. In: ENELKO 2013 XIV. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2013 XXIII. Nemzetközi Számítástechnika és Oktatás Konferencia, Nagyszeben, 2013. október 10–13. / szerk. Biró Károly Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 214–218, 2013.
20. Kádek, T., Kósa, M., Pánovics, J.: *A ProgCont szoftverrel támogatott programozó versenyek tapasztalatai*. In: New Technologies in Science, Research and Education / szerk. Veronika Stoffová, János Selye University, Komárno, Slovakia, 152–157, 2012.
21. Kádek, T.: *JavaScript függvénykönyvtár matematikai logika oktatásához*. In: ENELKO 2012 XIII. Nemzetközi Energetika-elektrotechnika Konferencia, SzámOkt 2012 XXII. Nemzetközi Számítástechnika és oktatás Konferencia, Gyulafehérvár, 2012. október 11–14. / szerk. Biró Károly Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 259–262, 2012.
22. Kádek, T., Kósa, M., Pánovics, J.: *Programozó versenyek támogatása webes alkalmazással*. In: ENELKO 2011 XII. Nemzetközi Energetika-Elektrotechnika

Konferencia, SzámOkt 2011 XXI. Nemzetközi Számítástechnika és Oktatás Konferencia, Kolozsvár, 2011. október 6–9. / szerk. Biró Károly-Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 184–187, 2011.

23. Kádek, T., Robu, J., Várterész, M.: *Matematikai logika példatár*. Presa Universitara Clujeana, Cluj-Napoca, 2010.
24. Kádek, T.: *Gentzen-levezetések generálása oktatási célú számítógépes támogatással*. In: ENELKO 2010 XI. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2010 XX. Nemzetközi Számítástechnika és Oktatás Konferencia, Szatmárnémeti, 2010. október 7–10. / szerk. Biró Károly-Ágoston, Sebestyén-Pál György, Erdélyi Magyar Műszaki Tudományos Társaság, Cluj-Napoca, 136–139, 2010.