

Egyetemi doktori (PhD) értekezés tézisei

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

Kádek Tamás

Témavezető:
Dr. Mihálydeák Tamás

Konzulens:
Dr. Várterész Magda



DEBRECENI EGYETEM
Informatikai Tudományok Doktori Iskola

Debrecen, 2017

Tartalomjegyzék

1. Bevezetés és motiváció	1
2. Eredmények	3
2.1. Parciális approximációs téren alapuló logikai rendszerek .	3
2.2. Parciális approximációs tér a mesterséges intelligenciában	9
3. Összefoglalás	15
4 Introduction and Motivation	18
5 Results	20
5.1 Logical Systems Based on Partial Approximation Space .	20
5.2 Partial Approximation Space in Artificial Intelligence . . .	26
6 Summary	32
Irodalomjegyzék / References	32
Publikációs lista / List of Publications	34

1. Bevezetés és motiváció

Az életlenhalmaz-elmélet – annak ellenére, hogy alapjait Pawlak már az 1980-as évek elején lerakta [6, 7] – mit sem veszített aktualitásából. Újra és újra felvetődik a kérdés, hogy miként dolgozzuk fel a rendelkezésre álló és folyamatosan keletkező adatokat, és hogy miként kezeljük az információ ábrázolásából vagy gyakran épp a hiányos információkból adódó pontatlanságot. A rendelkezésünkre álló hiányos tudás jelentette bizonytalanság kezelésére ad választ az életlen halmazok elmélete. Az így kialakuló halmazelméleti megközelítésben az *eleme* reláció válik bizonytalanná. Az egyes halmazokhoz bizonyosan tartozó vagy bizonyosan nem tartozó elemek mellett megjelenik egy köztes kategória is. A bizonytalanságot a rendelkezésre álló tudás korlátai generálják azáltal, hogy ezen korlátozott tudás alapján bizonyos objektumok megkülönböztethetetlenek egymástól.

Az utóbbi 30 évben az ekvivalenciarelációra épített Pawlak-féle rendszer több általánosítása is napvilágot látott [1, 8], amelyek közt megjelent a parciális approximációs terek használatának lehetősége is. Ez a kutatás számos szállal kötődik a Debreceni Egyetem Informatikai Karához. A klasszikus halmazelméletre épített elsőrendű logika mintájára az életlen halmazokra és a parciális approximációs térre alapozott parciális logikai rendszerek keletkeztek [5, 4]. A parciális logikai rendszer értékréses logikát jelent, ahol az értékrés (értékhiány) jeleníti meg a parcialitást. A doktori dolgozat egyik célja az így kialakult logikai rendszer vizsgálata. Ez a vizsgálat arra koncentrál, hogy a parciális logikai rendszerben kapott eredmények mennyiben csengnek össze a klasszikus logikában kaphatókkal. A cél, hogy feltárjuk a parciális logikai rendszer sajátosságait, és hogy felderítsük, mi lehet az előnye a közelítő számításokon alapuló rendszernek, és hol kell megfizetni ennek árát.

A különféle problémák mesterséges intelligenciával történő megoldása során az első lépés a probléma megfelelő modellezése. Ez gyakran elsőrendű logikai formulák formájában áll elő, különös tekintettel arra az esetre, ha magát a problémát állapottér-reprezentáció segítségével írjuk le és állapottéren alkalmazott megoldáskereső algoritmusok felhasználásával

szeretnénk megoldani. Innen adódik a kérdés, hogy miképp alakítsuk át a megoldáskereső algoritmusokat, ha a modellalkotás során fel szeretnénk használni a parciális logikai rendszer kínálta lehetőségeket.

Abban az esetben, ha már rendelkezésünkre áll egy, a probléma leírását megadó diszkrét véges állapottér, akkor adódik a lehetőség, hogy átalakítsuk azt ítéletlogikai SAT problémává. Innentől kezdve a probléma megoldása ítéletlogikai modellkeresést jelent, amelyre számtalan eszköz áll már rendelkezésre. Ugyanakkor a probléma átalakításával elvesz némi információ, ami egyértelmű hátrányt jelent a megoldás során. Az élethalmaz-elmélet ez esetben épp ezt a hátrányt segít leküzdeni.

2. Eredmények

2.1. Parciális approximációs téren alapuló logikai rendszerek

A parciális logikai rendszerek vizsgálatát az eszközalapú parciális elsőrendű logika áttekintésével kezdtük [3]. A közelítést a tudásunk alapján generált alaphalmazokkal végezzük; ezeket az alaphalmazokat tekintjük eszköznek. A rendszer maga rendkívül kifejező és összetett, ezért első lépésként egy egyszerűsített logikai nyelv került kialakításra, amely kezdetben csak ítéletlogikai, később egyszerű elsőrendű formulák és a belőlük alkotott következtetések vizsgálatára volt alkalmas. Az egyszerűsített nyelvben a parciális approximációs tér approximációs párját a formulák alsó és felső közelítésének meghatározására használtuk fel. Azt, hogy a formulák értékét alsó és felső közelítés szerint számítjuk, rendre $\downarrow$ és $\uparrow$ közelítő mondatfunktorokkal jelezzük. Az egyszerűsített nyelvben ez mindössze annyit jelent, hogy a formulák igazságértékének meghatározásakor az atomi formulák igazságértékét a megfelelő közelítés szerint számítjuk ki.

Az egyszerűsített nyelv kizárólag egyargumentumú predikátumszimbólumokat enged meg. Ez nagyban leegyszerűsíti az alsó és felső közelítés illusztrációját, amelyet a 2.1. ábra mutat be, felhasználva a logikai (a $\{0, 1\}$ halmazba képező) függvények pozitivitási tartományát.

2.1. Definíció. Egy $\varphi : U \rightarrow \{0, 1, 2\}$ függvény pozitivitási tartománya alatt a $[\varphi]^+$ -szal jelölt és a következőképpen definiált halmazt értjük:

$$[\varphi]^+ = \{ u : u \in U \text{ és } \varphi(u) = 1 \}.$$

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

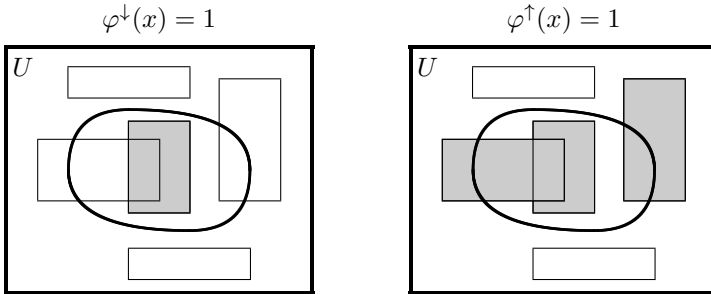
$$\varphi^\downarrow(u) = \begin{cases} 1 & \text{ha } u \in \mathfrak{l}([\varphi]^+), \\ 0 & \text{ha } u \in \mathfrak{l}(U \setminus \mathfrak{u}([\varphi]^+)), \\ 2 & \text{egyébként.} \end{cases}$$

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

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

A logikai függvények esetében a 2-es érték az igazságérték hiányát jelöli (míg a 0 és 1 rendre a *hamis* és *igaz* igazságérték jelölésére szolgál). A logikai rendszer parciaritását az igazságérték hiánya adja. A parciaritás következménye, hogy a logikai összekötő jelek szemantikája értékréses logikai rendszert alkot.

2.1. ábra. Atomi formulák alsó és felső közelítése



A 2.1. ábrán a φ logikai függvény pozitivitási tartományát a középén ábrázolt ovális terület szemlélteti, míg a vékonyabb vonallal határolt téglalapok a parciális approximációs tér alaphalmazait mutatják meg. Feltételezve, hogy $\langle \mathfrak{l}, \mathfrak{u} \rangle$ egy Pawlak-féle approximációs pár, a φ logikai függvény

alsó és felső közelítése a bal és jobb oldali ábrán sötét háttérrel kiemelt terület.

Bár az egyszerűsített logikai nyelv kifejezőereje messze elmarad az eredeti ([4]-ben definiált) rendszerétől, mégis alkalmas arra, hogy a parciális logikai rendszer sajátosságaira fényt derítsen. Minden olyan – a klasszikus logikában helytálló – állítás, amely az egyszerűsített nyelvet felhasználva cáfolható, az az eredeti rendszerben is cáfolható lesz.

A vizsgálatokat az egyargumentumú predikátumparaméterekkel és közelítő mondatfunktorokkal kiegészített ítéletlogikai nyelvvel kezdtük. Formulahalmazok kielégíthetőségét vizsgálva sikerült megmutatni, hogy előállítható olyan szituáció, hogy (többek közt) a *modus ponens* és a *modus tollens* kétpremisszás következtetések nem teljesülnek.

1. Tézis. *A klasszikus logika legismertebb kétpremisszás következtetései – a modus ponens és a modus tollens – igaznak bizonyulnak a parciális logikai rendszerben, mindaddig, amíg nem használunk az összetett premisszák és a konklúzió vonatkozásában ellentétes irányú közelítéseket (közelítő mondatfunktorokat). Ellenkező esetben ezen következtetések cáfolhatók.*

Az előbbi állítást formulahalmazok kielégíthetőségét vizsgálva az alábbi két tétel bizonyításával sikerült igazolni.

2.4. Tétel. (2.15.) *Van olyan $A \in \text{Form}$ és $B \in \text{Form}$ formula megfelelő parciális logikai nyelven, amely esetében $\{\downarrow A, \downarrow(A \supset B), \uparrow \neg B\}$ kielégíthető.*

2.5. Tétel. (2.17.) *Van olyan $A \in \text{Form}$ és $B \in \text{Form}$ formula megfelelő parciális logikai nyelven, amely esetében $\{\uparrow(A \supset B), \uparrow \neg B, \downarrow \neg A\}$ kielégíthető.*

Mindezek mellett azt is sikerült megmutatni, hogy atomi formulákból képzett premisszák és konklúzió esetén a *modus ponens* és a *modus tollens* érvényes, még akkor is, ha ellentétes közelítő mondatfunktorok állnak a premisszák és a konklúzió előtt.

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

Annak érdekében, hogy gyakorlati példán keresztül is meg tapasztal-

hassuk a parciális logikai rendszer sajátosságait, elkészítettük a szemantika relációs adatbázison alapuló implementációját. Az univerzumot ez esetben egyetlen adatbázistábla rekordjai adják. Az univerzum részhalmozai (azaz rekordokat tartalmazó halmazok) segítségével adjuk meg az egyargumentumú predikátumok pozitivitási tartományait. A predikátumparaméterek szemantikai értékét tárolt függvények definiálják, amelyek értelmezési tartománya a tábla rekordjainak halmaza. A predikátumparaméterek közül a felhasználó szabadon állíthatja össze az approximációs tér alaphalmazait adó eszközöket.

Az eszközök kiválasztása generálja az approximációs teret. Feltéve, hogy a $P_1, \dots, P_n$ predikátumparaméter közül az első k darab került kiválasztásra, két mátrix kerül felépítésre:

$$A = (a_{i,j})_{k \times n} \quad \text{és} \quad B = (b_{i,j})_{k \times n}.$$

$$\begin{aligned} a_{i,j} &= \left| [\varrho(P_i)]^+ \cap [\varrho(P_j)]^+ \right| \\ b_{i,j} &= \left| [\varrho(P_i)]^+ \setminus [\varrho(P_j)]^+ \right| \end{aligned}$$

ahol $i = 1, \dots, k$ és $j = 1, \dots, n$; míg a $\varrho(P_i) = \varphi_i$ az i -edik predikátumparaméterhez tartozó logikai függvény. 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:

$$\varphi_j^\downarrow(u) = \begin{cases} 1 & \text{ha van olyan } i, \text{ hogy } u \in [\varphi_i]^+ \text{ és } b_{i,j} = 0, \\ 0 & \text{ha van olyan } i, \text{ hogy } u \in [\varphi_i]^+ \text{ és } a_{i,j} = 0, \\ & \text{és egyetlen } \ell \text{ esetén sem teljesül } a_{\ell,i} > 0 \text{ és } a_{\ell,j} > 0, \\ 2 & \text{egyébként;} \end{cases}$$

$$\varphi_j^\uparrow(u) = \begin{cases} 1 & \text{ha van olyan } i, \text{ hogy } u \in [\varphi_i]^+ \text{ és } a_{i,j} > 0, \\ 0 & \text{ha van olyan } i, \text{ hogy } u \in [\varphi_i]^+ \text{ és } a_{i,j} = 0, \\ & \text{és egyetlen } \ell \text{ esetén sem teljesül } a_{\ell,i} > 0 \text{ és } a_{\ell,j} > 0, \\ 2 & \text{egyébként;} \end{cases}$$

ahol $i, \ell \in \{1, \dots, k\}$ és $j \in \{1, \dots, n\}$. Az alkalmazás segítségével összevethetjük egyszerű formulák igazságértékét a klasszikus és a parciális approximációs téren alapuló logikai rendszer esetében.

Arisztotelész szillogizmusai

A következőkben az eszközalapú parciális szemantikát Arisztotelész szillogizmusain keresztül vizsgáltuk meg. A szillogizmusokra azért esett a választás, mert úgy gondoltuk, ezek reprezentálják a leggyakoribb kvantifikált következtetéseket, amelyek egyrétű kvantifikáció segítségével formalizálhatóak. A vizsgálat során abból a naiv feltételezésből indulunk ki, hogy az approximációs térben az alsó közelítés a biztos tudásunkat testesíti meg, így érdemes megvizsgálni, hogy ebből a tudásból helyes következtetéseket vonhatunk-e le. Mivel a szillogizmusok a klasszikus elsőrendő logikában (néhány esetben a szükséges egzisztenciafeltevés mellett) helyes következtetések, következő vizsgálatunk tárgya az érvényességük tesztelése lett az approximációs téren. Az eddigi eredményekből az már világos volt, hogy a következtetések cáfolhatóak, így a kísérletezés célja, hogy különböző tulajdonságú approximációs terekben és interpretációkban állítsuk elő a cáfolatokat. A vizsgált esetek:

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

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

	$\langle U, \varrho \rangle$	<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
diszjunkt $\mathfrak{T}$	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
Összesen	134 213 632	121 536	227 232	264 540	325 332

Mind a 12 szillogizmus esetében hasonló eredményre jutottunk. Mind a 9 vizsgált esetben konstruálható cáfolat, legfeljebb a cáfolatok darabszáma különbözik. A *Barbara*, *Celarent*, *Darapti* és *Felapton* esetében kapott eredményeket foglalja össze a 2.1. táblázat. Az összes többi szillogizmus az előbbi 4 egyikével megegyező eredményt mutat.

2. Tézis. *Az eszközalapú parciális logikai rendszerben Arisztotelész kategorikus következtetései nem érvényesek, tekintet nélkül a választott approximációs tér, illetve az azt generáló interpretáció alapvető tulajdonságaira.*

A cáfolatok előállítására Java nyelven alkalmazás készült, amely lehetővé tette a több mint 134 millió interpretáció vizsgálatát. Az eredmények megmutatták, hogy a cáfolatok megléte nem függ az approximációs tér vagy az interpretáció tulajdonságaitól, hanem a logikai jelek szemantikájából adódik. Ezen eredményeket felhasználva két lehetőséget mutattunk arra, hogy miként helyettesítheti a közelítéssel végzett számítás a klasszikus módszert:

1. Bizonyos formulák alsó közelítés szerint számított értéke, feltéve hogy igazságérték (*igaz* vagy *hamis*) keletkezett, egybeesik a klasszi-

kus értelemben vett igazságértékkel. Hogy beazonosíthassuk az ilyen tulajdonságú formulákat, egy – a formulák szerkezetén alapuló – kalkulus került kidolgozásra.

2. A parciális logikai rendszer logikai összekötő jeleinek szemantikája átfogalmazható úgy, hogy a közelített számítás értéke – feltéve, hogy igazságérték keletkezett – egybeessen a klasszikus értelemben vett igazságértékkel. Ezt neveztük *pesszimista* szemantikának. Ennek egyik hátránya, hogy a keletkezett szemantika már nem felel meg az értékrésesnek nevezett logikai rendszerek követelményeinek. A másik hátrány pedig, hogy megnő azon formulák száma, amelyek értékelésének eredményeképpen nem áll elő igazságérték.

2.2. Parciális approximációs tér a mesterséges intelligenciában

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

2.7. Definíció. Egy p probléma állapotter-reprezentációja alatt a következőképpen definiált $\langle \mathcal{A}, a_0, \mathcal{C}, \mathcal{O} \rangle$ rendezett négyest értjük [2]:

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

A vizsgálat során abból a feltevésből indulunk ki, hogy a probléma állapotter-reprezentációjában szereplő operátoralkalmazási előfeltételek és kényszerfeltételek (legalább részben) megfogalmazhatóak a parciális logika korábban bevezetett nyelvén.

Visszalépéses és keresőfával megoldást keresők

Első lépésben az operátoralkalmazási előfeltételekre koncentráltunk. Az operátoralkalmazási előfeltételek válogatják ki az $a \in \mathcal{A}$ valódi állapotok közül azokat, amelyekre adott $o \in \mathcal{O}$ alkalmazható, azaz definiáljuk a $\text{dom}(o) \subseteq \mathcal{A}$ halmazt. A $\text{dom}(o)$ halmaz közelítéseként az alábbi halmazokat vezethetjük be:

1. $\text{dom}^\downarrow(o) \subseteq \mathcal{A}$ olyan állapotokat tartalmazó halmaz, amelyekre az o operátor bizonyosan alkalmazható;
2. $\text{dom}^\uparrow(o) \subseteq \mathcal{A}$ olyan állapotokat tartalmazó halmaz, amelyekre az o operátor lehetséges, hogy alkalmazható; ahol

$$\text{dom}^\downarrow(o) \subseteq \text{dom}(o) \subseteq \text{dom}^\uparrow(o) \subseteq \mathcal{A}.$$

Az operátorok értelmezési tartományának alsó és felső közelítését felhasználva adott $a \in \mathcal{A}$ állapot esetében ekkor bevezethető a

1. még nem alkalmazott, de bizonyosan alkalmazható operátorok halmaza, kezdetben:

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

2. és a még nem alkalmazott, de lehet, hogy alkalmazható operátorok halmaza, kezdetben:

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

A visszalépéses kereső algoritmusok ekkor átalakíthatóak úgy, hogy amíg erre lehetőség van, a bizonyosan alkalmazható operátorok irányába tárják fel az állapottérgráfot. Az alkalmazási előfeltétel klasszikus módszer szerinti (éles) kiértékelése elhalasztható mindaddig, amíg el nem fogynak a tekintett állapotban a bizonyosan alkalmazható operátorok. A közelítés így kvázi heurisztikaként irányítja a keresés menetét. A visszalépéses algoritmus átdolgozása a dolgozatban került részletesen kifejtésre.

Gyakran alkalmazott technika, hogy a valódi állapotok $\mathcal{A}$ halmazát egy bővebb $\mathcal{T} = T_1 \times \dots \times T_n$ halmaz részének tekintjük, ahol a $\mathcal{T}$ halmaz elemei közül a kényszerfeltételek segítségével válogatjuk ki $\mathcal{A}$ elemeit. Ekkor a probléma egy-egy állapotát n darab jellemzővel írjuk le, ahol az i . jellemző lehetséges értékeit sorolja fel a T_i halmaz. A kényszerfeltételekre általában azért van szükség, mert a jellemzők egymástól nem függetlenek,

nem minden kombinációjuk alkot valódi állapotot. A klasszikus megoldáskereső algoritmusok esetén a kényszerfeltételeket implementálni nem kell, az operátoralkalmazási előfeltételeket ugyanakkor igen, ezért amit csak lehet, igyekszünk a kényszerfeltételben megszorítani.

2.8. Definíció. Az $f : \mathcal{T} \rightarrow \mathcal{T}$ függvény az $o \in \mathcal{O}$ operátor általánosítása, ha minden $a \in \text{dom}(o)$ esetén $o(a) = f(a)$ és

$$\text{dom}(o) = \left\{ a : a \in \left(\widehat{\text{dom}}(f) \cap \mathcal{A} \right) \wedge f(a) \in \mathcal{A} \right\};$$

ahol a $\widehat{\text{dom}}(f) \subseteq \mathcal{T}$ halmaz az f általánosított operátor alkalmazási előfeltételeinek megfelelő elemeket tartalmazza.

Az általánosított operátorok használata során a kényszerfeltételtől független alkalmazási előfeltételeket emeljük ki a $\widehat{\text{dom}}(f)$ halmazba. Az általánosított operátorok használatával nyílik meg a lehetőség az elérhetőség fogalmának kiterjesztésére.

2.9. 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.

2.10. 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.

A keresőfával megoldást kereső algoritmusok (szélességi kereső, mélységi kereső, optimális kereső stb.) a keresés során nyilvántartott csúcsokat nyílt vagy zárt címkével látják el. A nyílt csúcsok – attól függően, hogy a

szülő csúcsból bizonyosan közvetlenül elérhető vagy lehetséges, hogy közvetlenül elérhető állapotokhoz tartoznak – bizonyosan nyílt, illetve lehetséges nyílt kategóriákra választhatók szét. A visszalépéses keresők min-tájára ezen algoritmusokat is átalakítottuk úgy, hogy a bizonyosan nyílt csúcsokat részesítse előnyben a keresés során.

3. Tézis. *A klasszikus visszalépéses és keresőfával megoldást kereső algoritmusok kiterjeszthetők úgy, hogy az állapottérgráfot – amíg ez lehetséges – az approximáció alapján biztosnak vélt tudásunk mentén járjuk be. Így a közelítés kvázi heurisztikaként működhet.*

Közelítő megoldás véges diszkrét állapottereken

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 modellel megoldani a problémát nem jelent mást, mint megoldani egy állításlogikai SAT feladatot. A tapasztalat ugyanakkor az, hogy a szokásos SAT solver implementációk alulmarnak 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 megoldáskeresőbe. Ekkor a megoldás menete a következő:

- (1) Állítsuk elő a részproblémákat, amelyek a problémát leíró klózhalmaz részhalmazai lesznek.
- (2) Oldjuk meg a részproblémákat külön-külön. Ha van kielégíthetetlen részprobléma (klózhalmaz), akkor a probléma nem oldható meg.
- (3) Keressük meg a részproblémák közös megoldását (akár egyszerű visszalépéses algoritmussal).

A felvázolt algoritmussal kapcsolatban felmerül egy komoly probléma. Attól függően, hogy miként végezzük el a részproblémákra bontást (a dolgozatban erre bemutatunk egy észszerű módszert), a részproblémák modelljeinek száma nagyságrendekkel meghaladhatja a probléma megoldásainak számát, és így megnehezedik a részproblémák közös megoldásainak

keresése. A megoldások leírásának egyszerűsítésében a halmazközelítés nyújthat segítséget.

2.11. Definíció. Legyen $L^{(0)} = \langle LC, Con, Form \rangle$ egy ítéletlogikai nyelv, továbbá jelölje $\mathfrak{A}$ a nyelv ítéletváltozóiból (a Con halmaz elemeiből) alkotott diszjunkt halmazpárok halmazát:

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

Ekkor minden $\langle P, Q \rangle \in \mathfrak{A}$ meghatároz egy interpretációcsaládot, amelyet $\mathcal{M}_{\langle P, Q \rangle}$ -val jelölünk, és ami olyan interpretációkat tartalmaz, amelyekben a P -beli ítéletváltozók igaz, míg a Q -beli ítéletváltozók hamis értéket kapnak. Formálisan: $\mathcal{M}_{\langle P, Q \rangle} = \{ \varrho : \varrho \subseteq Con \wedge P \subseteq \varrho \wedge Q \cap \varrho = \emptyset \}$.

Egy-egy részprobléma megoldásaként diszjunktív normál forma áll elő, amelynek minden elemi konjunkciója megfeleltethető egy-egy interpretációcsaládnak. Így a részprobléma megoldása interpretációcsaládok formájában adott.

2.12. 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:

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

2.13. Tétel. (3.9.) Legyenek $\langle P, Q \rangle \in \mathfrak{A}$ és $\langle P', Q' \rangle \in \mathfrak{A}$ tetszőleges halmazpárok, ekkor $\langle P, Q \rangle \sqcup \langle P', Q' \rangle \in \mathfrak{A}$, továbbá

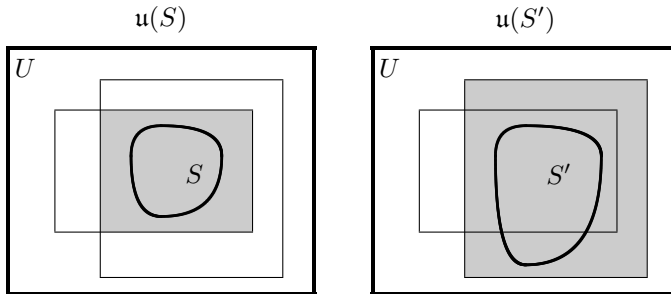
$$\mathcal{M}_{\langle P, Q \rangle \sqcup \langle P', Q' \rangle} \supseteq \mathcal{M}_{\langle P, Q \rangle} \cup \mathcal{M}_{\langle P', Q' \rangle}.$$

Két interpretációcsalád $\sqcup$ szerinti uniója (összevontja) tulajdonképp az interpretációcsaládokat leíró halmazpárok tagjainak metszete segítségével alakul ki. Az összevont interpretációcsaládba tartozó interpretációk halmaza a 2.13. tétel miatt felülről közelíti az interpretációcsaládokhoz tartozó interpretációk halmazainak unióját. Innen adódik az ötlet, hogy halmazmetszeteken alapuló felső közelítéssel egyszerűsítsük a részproblémák megoldásainak leírását.

A részproblémák megoldásait felülről közelítve a következőképpen alakíthatjuk át a probléma megoldásának menetét:

- (1) Állítsuk elő a részproblémákat, amelyek a problémát leíró klózhal-
maz részalmazai lesznek.
- (2) Oldjuk meg a részproblémákat külön-külön. Ha van kielégíthetetlen

2.2. ábra. Halmazmetszeteken alapuló felső közelítés illusztrációja



részprobléma (klózhalmaz), akkor a probléma nem oldható meg.

- (3) Helyettesítsük a részproblémák megoldásait azok felső közelítésével.
- (4) Keressük meg a részproblémák közös megoldását (akár egyszerű visszalépéses algoritlussal). Mivel ekkor már a közelített eredménnyel dolgozunk, nem valódi megoldások, csak megoldásjelöltek állnak elő.
- (5) A kényszer- és célfeltételek ellenőrzése segítségével válogassuk ki a valódi megoldásokat a megoldásjelöltek közül.

Az átdolgozott algoritmust és a halmazmetszeteken alapuló felső közelítést a dolgozatban két közismert példán, a *zebra-* és a *8-vezér problémán* keresztül mutattuk be. Bemutattuk, hogy ezzel a módszerrel közel azonos idő alatt lehet megoldani a SAT problémát, mint az eredeti, állapottér-reprezentáción alapuló klasszikus keresők esetében.

4. Tézis. A véges diszkrét állapotterekből generált SAT problémák esetében van létjogosultsága a részproblémák közelítő megoldásán alapuló stratégiáknak.

Ugyanakkor nem szabad megfeledkezni arról, hogy egy-egy probléma számos különböző – más-más szempontok szerint mind kiváló – állapottér-reprezentációval rendelkezhet. A modellalkotó feladata, hogy az állapottérgráf tulajdonságainak ismeretében megválassza, melyik megoldáskereső algoritmust használja, vagy hogy a kiválasztott algoritmushoz leginkább illeszkedő modellt alkossa meg.

3. Összefoglalás

A pawlaki rendszer általánosításaként létrejött parciális approximációs téren a közelített számítások alapja annak a bizonytalanságnak a tudomásulvétele, ami a rendelkezésünkre álló tudás korlátaiból fakad. A parciális elsőrendű logika, amely szemantikájába beemeli a közelítő számítást (ennek minden bemutatott előnye és hátránya mellett), lehetőséget kínál arra, hogy ezen rendelkezésünkre álló (vagy esetleg a gyorsan elérhető) tudás alapján alkossunk következtetéseket. Mindez átültethető az állapottér-reprezentációk világába a reprezentációt kialakító formulákon keresztül, legyen szó akár a megoldáskereső algoritmusok kiterjesztéséről, vagy az állapottér-reprezentáció által meghatározott ítéletlogikai SAT feladat megoldásáról.

Thesis of Doctoral (PhD) Dissertation

**THE INVESTIGATION OF
PARTIAL-APPROXIMATION-SPACE-BASED LOGICAL
SYSTEMS AND THEIR APPLICATION IN ARTIFICIAL
INTELLIGENCE**

Tamás Kádek

Supervisor:
Dr. Tamás Mihálydeák

Consultant:
Dr. Magda Várterész



UNIVERSITY OF DEBRECEN
Doctoral School of Informatics

Debrecen, 2017

4 Introduction and Motivation

Rough set theory hasn't lost its actuality till now, despite it was founded by Pawlak at the beginning of the 1980s [6, 7]. 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 [1, 8]. 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 [5, 4]. 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 is 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 Results

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) [3]. 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

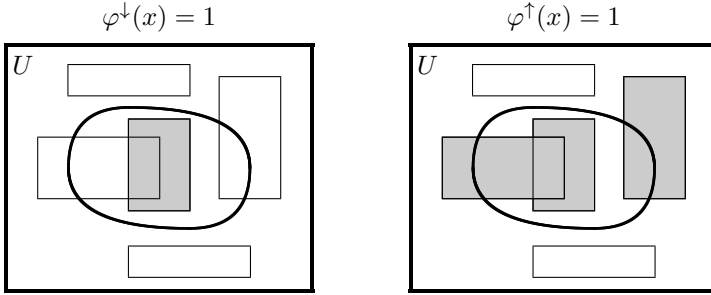
$$\varphi^\downarrow(u) = \begin{cases} 1 & \text{if } u \in l([\varphi]^+), \\ 0 & \text{if } u \in l(U \setminus u([\varphi]^+)), \\ 2 & \text{otherwise;} \end{cases}$$

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 [4]), 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.

Thesis 1. *Two of the best-known inference rules with two premises — modus ponens and modus tollens — hold in the partial logical system until we avoid using lower and upper approximations (approximative sentence functors) together in the compound premises and the conclusion. Using both lower and upper approximations together and choosing appropriate compound premises and conclusion, refutations can be constructed.*

We managed to prove the above statement by investigating the satisfiability of formula sets as the following two theorems show.

Theorem 5.4. (2.15.) *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. (2.17.) *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. (2.16.) *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:

$$\varphi_j^\downarrow(u) = \begin{cases} 1 & \text{if there exists an } i \text{ such that } u \in [\varphi_i]^+ \text{ and } b_{i,j} = 0, \\ 0 & \text{if there exists an } i \text{ such that } u \in [\varphi_i]^+ \text{ and } a_{i,j} = 0, \\ & \text{and there is no } \ell \text{ such that } a_{\ell,i} > 0 \text{ and } a_{\ell,j} > 0, \\ 2 & \text{otherwise;} \end{cases}$$

$$\varphi_j^\uparrow(u) = \begin{cases} 1 & \text{if there exists an } i \text{ such that } u \in [\varphi_i]^+ \text{ and } a_{i,j} > 0, \\ 0 & \text{if there exists an } i \text{ such that } u \in [\varphi_i]^+ \text{ and } a_{i,j} = 0, \\ & \text{and there is no } \ell \text{ such that } a_{\ell,i} > 0 \text{ and } a_{\ell,j} > 0, \\ 2 & \text{otherwise;} \end{cases}$$

where $i, \ell \in \{1, \dots, k\}$ and $j \in \{1, \dots, n\}$. Using the application, we can compare the truth values of simple fomulae in the classical and partial logical system.

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 $P \in Con$, 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 $P \in Con$, 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 $P \in Con$, 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 $P \in Con$ 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

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

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.

Thesis 2. *In the tool-based partial logical system, Aristotle’s categorical statements are invalid, independently from the properties of the chosen approximation space and the generating interpretation.*

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.

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 $\langle \mathcal{A}, a_0, \mathcal{C}, \mathcal{O} \rangle$ [2], 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.

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 $a \in A$

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.

Thesis 3. *The classic backtracking and tree search algorithms can be extended in such a way that the state-space graph is discovered — until it is possible — based on the knowledge believed to be certain by the approximation. The approximation may control the search such as heuristics do.*

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:

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

Theorem 5.13. (3.9.) 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

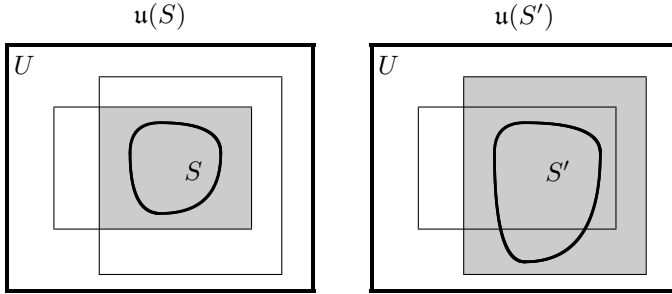
$$\mathcal{M}_{\langle P, Q \rangle \sqcup \langle P', Q' \rangle} \supseteq \mathcal{M}_{\langle P, Q \rangle} \cup \mathcal{M}_{\langle P', Q' \rangle}.$$

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.

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.

Figure 5.2: Illustration of set-intersection-based upper approximation



- (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.

Thesis 4. *In case of SAT problems generated from finite discrete state spaces, the existence of the strategies based on the approximate solutions of subproblems is justifiable.*

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.

6 Summary

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.

Irodalomjegyzék /

References

- [1] DÜNTSCH, I., AND GEDIGA, G. *Approximation Operators in Qualitative Data Analysis*. Springer, Berlin, Heidelberg, 2003, pp. 214–230.
- [2] FUTÓ, I., Ed. *Mesterséges Intelligencia*. Aula Kiadó, Budapest, 1999.
- [3] MIHÁLYDEÁK, T. *Partial First-order Logic with Approximative Functors Based on Properties*. Springer, Berlin, Heidelberg, 2012, pp. 514–523.
- [4] MIHÁLYDEÁK, T. Partial first-order logical semantics based on approximations of sets. *Non-classical Modal and Predicate Logics* (2011), 85–90.
- [5] 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.
- [6] PAWLAK, Z. Rough sets. *International Journal of Computer & Information Sciences* 11, 5 (1982), 341–356.
- [7] PAWLAK, Z. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Theory and Decision Library D:. Springer Netherlands, 1991.
- [8] YAO, Y. Y. *On Generalizing Rough Set Theory*. Springer, Berlin, Heidelberg, 2003, pp. 44–51.

Publikációs lista / List of Publications

Referált közlemények / Refereed Publications

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., Bátfai, N., Jeszenszky, P., Laki, S., Vágner, A., Kollár, L., Sidló, C., Besenczi, R., Smajda, M., Kövér, G., Szincsák, 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

Egyéb közlemények / Other Publications

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 Zawadzki, 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, Zakynthos Island, Greece, 44-50, 2015.
12. Fábrián, 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.



Nyilvántartási szám: DEENK/272/2017.PL
Tárgy: PhD Publikációs Lista

Jelölt: Kádek Tamás
Neptun kód: GHCC97
Doktori Iskola: Informatikai Tudományok Doktori Iskola
MTMT azonosító: 10038705

A PhD értekezés alapjául szolgáló közlemények

Idegen nyelvű, külföldi könyvrészletek (2)

1. **Kádek, T.,** Mihálydeák, T. S.: 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, (Annals of Computer Science and Information Systems, ISSN 2300-5963 ; Vol. 7.) ISBN: 9788360810781
2. **Kádek, T.,** Mihálydeák, T. S.: 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, (Lecture Notes in Artificial Intelligence, ISSN 0302-9743 ; vol. 8536.) ISBN: 9783319086439

Magyar nyelvű konferencia közlemények (1)

3. **Kádek, T.,** Mihálydeák, T. S.: 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, (ISSN 1842-4546)





Idegen nyelvű konferencia közlemények (1)

4. **Kádek, T.**, Mihálydeák, T. S.: 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. [elektronikus dokumentum]. Eds.: Emőd Kovács, Gábor Kusper, Roland Kunkli, Tibor Tómacs, Eszterházy Károly Főiskola, Eger, 51-58, 2014. ISBN: 9786155297182

További közlemények

Magyar nyelvű könyvek (1)

5. **Kádek, T.**, Robu, J., Várterész, M.: Matematikai logika példatár. Presa Universitara Clujeana, Cluj-Napoca, 174 p., 2010. ISBN: 9789735951221

Idegen nyelvű, külföldi könyvrészletek (4)

6. 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. ISBN: 9781467381291
7. 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, Zakynthos Island, Greece, 44-50, 2015, (Recent Advances in Computer Engineering Series, ISSN 1790-5109 ; 32.) ISBN: 9781618043207
8. 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. ISBN: 9781479972814





9. Szabó, R., Farkas, K., Ispány, M., Benczúr, A. A., Bátfai, N., Jeszenszky, P., Laki, S., Vágner, A., Kollár, L., Sidló, C., Besenczi, R., Smajda, M., Kövér, G., Szincsek, T., **Kádek, T.**, Kósa, M., Adamkó, A., Lendák, I., Wiandt, B., Tomás, T., Nagy, A. Z., 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. ISBN: 9781479915439

Idegen nyelvű közlemények hazai folyóiratban (1)

10. **Kádek, T.**: Press-ready deduction trees in classical logic using point-plus-expressions.
Annales Univ. Sci. Budapest., Sect. Comp. 37, 229-238, 2012. ISSN: 0138-9491.

Idegen nyelvű közlemények külföldi folyóiratban (2)

11. **Kádek, T.**, Pánovics, J.: Some improvements of the extended breadth-first search algorithm.
Studia Univ. Babeş-Bolyai. Ser. Inf. 59 (Special), 165-173, 2014. ISSN: 1224-869X.
12. **Kádek, T.**, Pánovics, J.: Extended breadth-first search algorithm.
Int. J. Comput. Sci. Issues. 10 (6), 78-82, 2013. ISSN: 1694-0814.

Magyar nyelvű konferencia közlemények (11)

13. 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, (ISSN 1842-4546)
14. **Kádek, T.**, Kósa, M., Pánovics, J.: Klasszikus állapotér-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, (ISSN 1842-4546)
15. Fábán, 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, (ISSN 1842-4546)



16. 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, (ISSN 1842-4546)
17. **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, (ISSN 1842-4546)
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, (ISSN 1842-4546)
19. **Kádek, T.**, Pánovics, J.: Általános állapotfőző 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, (ISSN 1842-4546)
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. ISBN: 9788081220630
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, (ISSN 1842-4546)
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, (ISSN 1842-4546)



23. **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, (ISSN 1842-4546)

Időgen nyelvű konferencia közlemények (1)

24. **Kádek, T.,** Pánovics, J.: Extended breadth-first search algorithm in practice.
In: Proceedings of the 9th International Conference on Applied Informatics January 29 -
Február 1, 2014. Eger, Hungary Volume I. [elektronikus dokumentum]. Eds.: Emőd Kovács,
Gábor Kusper, Roland Kunkli, Tibor Tómacs, Eszterházy Károly Főiskola, Eger, 59-66, 2014.
ISBN: 9786155297182

A DEENK a Jelölt által az iDEa Tudóstérbe feltöltött adatok bibliográfiai és tudománymetriai
ellenőrzését a tudományos adatbázisok és a Journal Citation Reports Impact Factor lista alapján
elvégezte.

Debrecen, 2017.09.01.

