Reprezentácia neúplných a nekonzistentných znalostí (rozšírený abstrakt)

Ján Šefránek

Fakulta matematiky, fyziky and informatiky, Univerzita Komenského
sefranek@fmph.uniba.sk

1 Úvod

Prednáška[1] bude venovaná hlavne teoretickému výskumu v oblasti reprezentácie znalostí, založenému na logike. Akcent bude na výpočtoch s neúplnými a nekonzistentnými znalosťami. Odvodzovanie z neúplných (nekonzistentných) znalostí je nepostrádateľná schopnosť netriviálneho znalostného systému, zvlášť ak je navrhovaný pre dynamickú doménu (alebo ak očakávame, že jeho báza znalostí sa bude postupne vyvíjať). Ak sa v báze znalostí vyskytne nekonzistentnosť, usudzujúci stroj by ju mal vedieť lokalizovať, prípadne odstrániť (na rozdiel od klasickej logiky, ktorá z nej odvodí čokoľvek). Všímať si budeme predovšetkým sémantické problémy, ale aj hľadisko vypočítateľnosti.

Súčasná teória reprezentácie znalostí považuje odvodzovanie dôsledkov (usudzovanie, reasoning) za centrálny problém, kľúčovú črtu a úlohu reprezentácie znalostí. Neseparuje odovodzovanie od reprezentácie. Súčasťou definície jazyka na reprezentáciu znalostí je špecifikácia relácie odvodenia. Nástroje na reprezentáciu poznatkov, blízke implementačnej úrovni, údajovým štruktúram nie sú z hľadiska súčasnej teórie zaujímavými a dôležitými objektami výskumu.

V prednáške najprv načrtneme históriu reprezentácie znalostí založenej na logike. Potom sa budeme venovať niektorým formalizáciám, na ktorých ukážeme, ako sa riešia problémy s neúplným a nekonzistentným poznaním. Všimneme si nemonotónne pravidlá, abdukciu, sémantiku stabilných modelov, dynamické logické programovanie. Poukážeme na negatívne zložitostné charakterizácie nemonotónneho usudzovania, načrtneme, aký pozitívny obsah tieto charakterizácie majú. Budeme sa venovať paradigme answer set programming a možnostiam rozsiahlych aplikácií v tejto paradigme.

2 Historická perspektíva

John Mc Carthy [1] podľa všetkého ako prvý prišiel s predstavou reprezentácie znalostí založenej na logike. Vlastnú prácu [1] characterizuje ako “probably the first paper on logical AI[2], i.e. AI in which logic is the method of representing information in computer memory”. Výskumný program, s ktorým prišiel, ilustroval návrhom počítačového programu, zvaného *advice taker* (AT). AT navrhoval tak, aby bol rozumný (a *program with common sense*) v tom zmysle, že “it automatically deduces for itself a sufficiently wide

[1] Práca bola podporená grantom APVV-20-P04805 a grantom VEGA 1/0173/03.

[2] artificial intelligence

class of immediate consequences of anything it is told and what it already knows". Jednou z jeho kľúčových myšlienok bola myšlienka deklaratívneho programovania (tak dnes tú predstavu nazývame). Správanie programu je determinované vetami, ktoré mu oznamujeme. Jeho správanie môžeme opraviť, ak mu oznámime iné vety. Tento spôsob špecifikácie správania programu umožňuje modularitu a nezávislosť v nasledujúcom zmysle slova: ak oznamujeme nejaké vety programu AT, nepotrebujeme vedieť nič (alebo potrebujem vedieť iba veľmi málo) o AT a o jeho predchádzajúcich poznatkoch (oveľa neskôr na označenie tejto schopnosti použil Mc Carthy slogan *elaboration tolerance* [4] – ak rozpracujeme, konzistentne doplníme poznatky systému, nemení to jeho spráávanie, založené na starších poznatkoch).

Teraz budeme venovať pozornosť inej kľúčovej myšlienke, ktorá stála za projektom AT (za víziou reprezentácie znalostí, založenej na prvorádovej logike). Ide o predstavu, že je potrebné (ale aj postačujúce) zozbierať všetky potrebné poznatky a z nich "jednokrokovou" dedukciou odvodzovať dôsledky, pričom zabezpečíme, aby sa dôsledky odvodzovali iba z relevantných predpokladov. Táto vízia precenila možnosti deduktívnej logiky, ale aj ľudí, ktorých úlohou by bolo zozbierať všetky informácie potrebné a postačujúce na to, aby z nich počítač získal všetky dôsledky, ktoré získava common sense.

Výskum v oblasti umelej inteligencie (a reprezentácie znalostí), založenej na logike, skoro narazil na obmedzenia dedukcie. Ukázalo sa, že je potrebné pracovať s neúplným poznaním a riešiť nekonzistentnosti. To, samozrejme, nie je zvládnuteľné klasickou deduktívnou logikou. Pokusy s prvorádovou logikou narážali aj na výpočtovú zložitosť a nevypočítateľnosť problémov, spätých s korektným a úplným odvodzovaním.

Spomínané okolnosti a skúsenosti viedli k tomu, že priekopníci representácie znalostí, založenej na logike, hľadali iný, vhodnejší logický aparát. Aparát, ktorého vyjadrovacia sila by bola vyššia a ktorého výpočtové vlastnosti priaznivejšie. Aj sám John Mc Carthy prišiel s novou víziou. Charakterizoval ju sloganom "jumping to conclusions". Obsahom tejto predstavy bolo, že z reprezentovaných poznatkov treba vedieť odvodiť aj také dôsledky, ktoré nezískame dedukciou (dôkladnou, ale zdĺhavou). Takéto dôsledky sú vlastne hypotézami, domnienkami, ľudia ich získavajú vďaka common sensu. Logiku, ktorú navrhol Mc Carthy, nazval cirkumskripciou [2, 3]. Predpokladajme, že program dostal špecifikáciu nejakej úlohy (vyjadrenú pomocou logických formúl). Základnou myšlienkou cirkumskripcie bolo minimalizovať výnimočné stavy, odchýľky od špecifikovaného prípadu a tak získať iba dôsledky, zodpovedajúce danej špecifikácii a nezohľadňujúce všetky možné stavy sveta, ktoré by situáciu komplikovali.

Usudzovanie, formalizované pomocou cirkumskripcie je *nemonotónne*: ak sa doplnia nové poznatky, nemusia byť odvoditeľné niektoré dôsledky, ktoré boli odvoditeľné predtým. Ak množina formúl A je nadmnožinou B, potom dôsledky A nemusia byť nadmnožinou dôsledkov B. Ďalšími najznámejšími formalizáciami nemonotónneho usudzovania z konca sedemdesiatych a začiatku osemdesiatych rokov sú defaultová logika [5] a autoepistemická logika [6]. Nemonotónne usudzovanie sa stalo oblasťou horúceho výskumu v reprezentácii znalostí a je v centre pozornosti dodnes. Dosiahli sa hlboké výsledky týkajúce sa sémantických základov, výpočtovej zložitosti. Navrhli a implementovali sa sofistikované algoritmy.

Ďalší dôležitý impulz dostal výskum vreprezentácii znalsotí koncom osemdesiatych a začiatkom deväťdesiatych rokov. Výskumníci z troch oblastí – logického programovania, deduktívnych databáz a nemonotónneho usudzovania – si uvedomili blízkosť, príbuznosť problémov, ktoré riešili. Každá z týchto oblastí pritom vznikla s inou motiváciou a s odlišnými cieľmi. Logické programovanie a deduktívne databázy narazili na problémy s nemonotónnosťou negácie ako konečného zlyhania. Medzi hlavné výzvy patrila uspokojujúca sémantická charakterizácia jazyka s negáciou. Na druhej strane, logické programovanie a deduktívne databázy sa ukazovali byť sľubnými prostrediami pre implementáciu nemonotónnych odvodzovacích strojov. Celkom prirodzene výskum v reprezentácii znalostí vďaka synergii týchto troch oblastí dosiahol hlbšie teoretické výsledky, objavili sa nové a úspešné sémantické konštrukcie, získali sa komplexitárske charakterizácie a vytvoril sa aj priestor pre solídne implementácie. Logické programovanie sa začalo považovaž za prostriedok (prinajmenšom teoretický model) reprezentácie znalostí [7].

3 Formalizácie usudzovania s neúplnými a nekonzistentnými znalosťami

Jadro prednášky bude venované niektorým prístupom k formalizácii usudzovania s neúplnými a nekonzistentnými znalosťami, ich sémantikám, zložitostným aspektom, implementáciám. Nasleduje iba veľmi stručný prehľad tém.

3.1 Nemonotónne pravidlá

Neúplnosť (ľudských) znalostí kompenzujeme hypotetickým usudzovaním. Hypotézy (aj tie najelementárnejšie – napríklad o tom, kde je kniha, ktorú práve hľadáme) prijímame, ak ich podporuje naša doterajšia evidencia a naše všeobecnejšie poznatky. Prijímame ich dovtedy, kým ich nevyvráti ďalšia získaná evidencia alebo nejaká solídne podložená teória (i keby sa týkala iba hľadanej knihy).

Takýto druh neúplnosti nášho poznania sa často prejavuje tým, že poznáme iba čiastočne podmienky platnosti nejakého dôsledku, ale vieme identifikovať aspoň niektoré prípady, kedy neplatí. Budeme sa venovať formalizácii tohto druhu hypotetického usudzovania. Príkladom sú defaultové pravidlá [5]. Ich zovšeobecnenie – nemonotónne pravidlá – pochádza od [16]. Nemonotónnne pravidlá sú trojice tvaru $\langle \gamma, X, Y \rangle$, kde γ je dôsledok, X množina predpokladov a Y možno nazvať množinou stráží. Ak niektorý z predpokladov neplatí, dôsledok nemôžeme odvodiť podľa tohto pravidla. Podobne, ak niektorý prvok z Y je vyvrátený (platí jeho opak), pravidlo nemožno aplikovať. Inak je pravidlo aplikovateľné. Vidno, že doplnenie poznatkov môže viesť k vyvráteniu množiny stráží Y.

Prednáška bude venovaná aj sémantickej charakterizácii systémov s nemonotónnymi pravidlami. Použije sa fixpointová konštrukcia. Intuitívny zmysel konštrukcie možno vyjadriť tak, že sa špecifikuje nasýtená množina hypotéz, ktorá konzistentne zúplňuje neúplné poznanie (bázu poznatkov).

3.2 Abdukcia

Ďalším spôsobom usudzovania, ktorým sa vyrovnávame s neúplnosťou nášho poznania, je abdukcia. Pri diagnostickom usudzovaní usudzujeme z pozorovateľných prejavov (symptómov) na ich skryté príčiny, na stavy, ktoré nie sú priamo pozorovateľné. Naše poznanie týchto vzťahov býva neúplné, odhadované príčiny alebo stavy iba hypotetické. Pri plánovaní postupne odvodzujeme dosiahnuteľnosť vzdialených cieľov z opisu počiatočnej situácie, z predpokladaného splnenia podmienok vykonania nejakých akcií a z ich očakávaných výsledkov. Opäť, dosiahnutie cieľa naplánovanými akciami nie je zaručené, efekty akcií nie sú dokonale predvídateľné, okolnosti, ktoré ich podmieňujú, nie sú spravidla opísateľné do detailov. I plánovanie, i diagnostikovanie možno modelovať abdukciou.

V deduktívnej logike sa abdukcia definuje nasledovne. Majme pozorovania Δ a relevantné poznatky Γ (teóriu v pozadí). Hovoríme, že Φ je abduktívnym vysvetlením pozorovaní Δ, ak

- $\Gamma \not\models \Delta$ (inak by nemalo zmysel hľadať vysvetlenie, naše poznanie domény by bolo z hľadiska daných pozorovaní úplné),
- $\Gamma \cup \Phi \models \Delta$,
- $\Gamma \cup \Phi$ je konzistentné (v opačnom prípade by v klasickej logike bola predchádzajúca podmienka splnená triviálne).

Je prirodzené hľadať zovšeobecnenia, ktoré by prekonali obmedzenia klasickej logiky. Klasické vyplývanie nie je celkom vhodný vzťah, pozorovania i hypotézy bývajú niekedy nekonzistentné s dostupným poznaním. Jeden spôsob, ako prekonať obmedzenia, späté s klasickou logikou ukázal [17].

Prístupy k abdukcii, pri ktorých sa konštrukty klasickej logiky nahrádzajú konštruktami z nemonotónnych formalizmov, budú prezentované v prednáške. Dôraz bude na črtách, ktoré prinášajú navyše v porovnaní s deduktívnou logikou, pozri napr. [14, 15].

3.3 Výpočtová zložitosť

Pôvodné očakávania, že nemonotónne logiky umožnia, aby reprezentácie znalostí a odvodzovanie boli rýchlejšie, sa nenaplnili. Komplexitárske výskumy dokonca ukázali, že výpočtová zložitosť formalizácií nemonotónneho usudzovania je podstatne vyššia než pri dedukcii. Najzávažnejšie výsledky uvedieme v prednáške, prehľad v [8, 9, 11].

Predpokladalo sa, že hlavnou príčinou negatívnych zložitostných charakterizácií je nutnosť preverovať splniteľnosť (konzistentnosť) a vyplývanie (videli sme to na príklade abdukce). Ukázalo sa však, že nejde o jediné príčiny výpočtovej náročnosti hypotetického usudzovania. Ak uvažujeme radikálne limitovaný jazyk (defaultové teórie bez disjunkcie), kde sa problém splňovania a vyplývania redukuje na preverovanie členstva v množine, zložitostné charakteristiky naďalej ukazujú, že ide o ťažké problémy, pozri [10]. Hlavnou príčinou (v prípade problémov, súcisiacich s extenziami defaultových teórií) bývajú interakcie, cyklická závislosť defaultových pravidiel a potenciálny exponenciálny nárast počtu extenzií. Toto pozorovanie možno generalizovať.

3.4 Kompaktnosť reprezentácie

Veľmi zaujímavú otázku si postavili [11]: je výpočtová nezvládnuteľnosť nemonotónneho usudzovania skutočne jeho nedostatkom, nevýhodou? Podarilo sa im ukázať, že tieto výsledky možno pochopiť ako extrémnu kompaktnosť reprezentácie v nemonotónnych formalizmoch. Použili na to techniku, ktorú nazvali kompilovateľnosťou.

Predpokladajme, že máme inštanciu (alebo viac inštancií) výpočtovo ťažkého (napr. NP-úplného) problému, representovaného v jazyku J_1, ale môžeme si dovoliť preložiť ho off-line (keď nezáleží na čase) do inej reprezentácie v jazyku J_2 a potom, neskôr, keď záleží na čase, riešiť problém v novej reprezentácii (kde riešenie bude on-line a rádovo rýchlejšie).

Príklad: nech T je formula v konjunktívnej normálnej forme, l je literál a problémom je zistiť, či $T \models l$. Trojicu $\langle T \models l; T; l\rangle$ považujme za problém s fixnou (T) a variabilnou (l) časťou. Ak off-line skonštruujeme tabuľku, v ktorej pre každý literál vyskytujúci sa v T zaznamenáme, či vyplýva z T alebo nie, môžeme hovoriť, že problém vyplývania sme skompilovali na problém prehľadávania tabuľky. Navyše, rozsah tabuľky je $\mathcal{O}(n)$, kde n je počet literálov v T. Pri konštrukcii tabuľky riešime $\mathcal{O}(n)$ inštancií co-NP-úplného problému, ale vtedy, keď potrebujeme riešenie, môžeme ho získať v lineárnom čase.

Problém $\langle P, F, V\rangle$, kde F je jeho fixná a V jeho variabilná časť, je *kompilovateľný*, ak existujú dva polynómy p_1 a p_2 a algoritmus $\mathcal{A}$ také, že pre každú inštanciu f fixnej časti F existuje údajová štruktúra D_f taká, že jej rozsah $|D_f|$ nepresahuje hodnotu $p_1(|f|)$ a pre každú inštanciu v variabilnej časti V dáva $\mathcal{A}$ pre vstupy D_f a v presne tú odpoveď ako riešenie pôvodného problému P a čas, ktorý $\mathcal{A}$ na riešenie potrebuje, neprevyšuje $p_2(|v| + |D_f|)$.

Celá táto konštrukcia má zmysel iba vtedy, ak nová reprezentácia je nanajvýš polynomicky rozsiahlejšia než pôvodná. Ukázalo sa, že pre niektoré výpočtovo nezvládnuteľné problémy, späté s nemonotónnym usudzovaním, nie je možná kompilácia do priestoru iba polynomicky rozsiahlejšieho. Interpretácia tohto výsledku je, že nemonotónny formalizmus umožňuje reprezentovať poznatky vo veľmi kompaktnom tvare [11] – iná reprezentácia by vyžadovala exponenciálne väčší priestor.

Poznamenajme na záver, že práca, ocenená ako najlepšia na ostatnej vrcholnej konferencii o reprezentácii znalostí KR 2004 používala techniku kompilácie na reprezentáciu preferencie [13]. Preferencie sú ďalšou dôležitou témou z hľadiska reprezentácie znalostí.

3.5 Answer set programming

Answer set programming je relatívne nová (koniec 20. storočia) paradigma logického programovania (radšej sa vyhnem pokusu o slovenský termín programovanie s odpoveďovými množinami). Sémantickým základom prístupu sú stabilné modely [18], neskôr zovšeobecnené na odpoveďové množiny (answer sets) pre jazyky s explicitnou negáciou [19]. Nasledovali ďalšie zovšeobecnenia pre ďalšie rozšírenia jazyka. Stabilný model si môžeme definovať ako množinu literálov S, ktorá spĺňa nasledujúcu podmienku: nech S^- je množina všetkých negatívnych litetálov z S, P je logický program a *least* je operátor, ktorý vypočíta najmenší model logického programu bez negácie (na negatívne literály nazeráme ako na nové atómy). Potom S je stabilným modelom programu P práve vtedy, keď $S = least(P \cup S^-)$.

Základná myšlienka tejto novej paradigmy je nasledovná – vypočíta sa stabilný model programu (všetky jeho stabilné modely) a na otázky sa odpovedá (problémy sa riešia) na základe stabilného modelu (stabilných modelov). Fakt, že môže existovať viac stabilných modelov, zväčšuje vyjadrovaciu silu formalizmu, umoňuje uvažovať alternatívne riešenia, možno rozlíšiť skeptické a dôverčivé riešenia. Samozrejme, z výpočtového hľadiska ide o ťažké problémy, nájdenie stabilného modelu je NP-úplný problém. Napriek tomu existujú sofistikované a optimalizované systémy dlv [25, 26] a smodels [23, 24], ktoré dovoľujú rozsiahle úlohy (rádovo desaťtisíce pravdiel) riešiť v solídnom čase (rádovo desiatky sekúnd).

Answer set programming sa považuje za sľubný nástroj na reprezentáciu poznatkov. Rozvíja sa rôznymi smermi – od fundamentálneho výskumu, vedúcemu k optimalizácii výpočtov až po aplikácie v takých doménach ako usudzovanie o akciách a zmene, usudzovanie o dynamických doménach, plánovanie, diagnostikovanie, integrácia dát z heterogénnych zdrojov, systém na podporu rozhodovania pre NASA, sémantický web, pozri [20–23, 25, 27–31] a mnoho ďalších. Treba zvlášť spomenúť monografiu [22].

3.6 Dynamické logické programovanie

Multidimenzionálny dynamický logický program je množina logických programov, na ktorej je definovaná relácia preferencie. Medzi programami môžu vznikať konflikty, ktoré sa riešia v súlade s reláciou preferencie [32–34]. Poukážeme na niektoré neintuitívnosti sémantiky, založenej iba na riešení konfliktov medzi hlavami pravidiel a načrtneme vlastú sémantiku, založenú na závislostiach, ktorá tieto vady odstraňuje a približuje dynamické logické programovanie ku klasickému výskumu revízií a k preferenčnému logickému programovaniu.

References

1. John Mc Carthy, Programs with common sense. Proceedings of Teddington Conference on the Mechanization of Thought Processes, 1958; http://www-formal.stanford.edu/jmc/mcc59.html
2. John McCarthy, Epistemological problems of artificial intelligence. Proceedings of IJ-CAI'77; http://www-formal.stanford.edu/jmc/epistemological/epistemological.html
3. John Mc Carthy, Circumscription – a form of nonmonotonic reasoning. Artificial Intelligence Journal, 1980; http://www-formal.stanford.edu/jmc/circumscription.html
4. John Mc Carthy, Elaboration tolerance. Proceedings of Common Sense 98; http://www-formal.stanford.edu/jmc/elaboration.html
5. Ray Reiter, A logic for default reasoning. Artificial Intelligence, 13:81-132, 1980.
6. R.C. Moore, Semantical considerations on non-monotonic logic, Arti cial Intelligence 25, pp. 75-94, 1985.
7. Chitta Baral, Michael Gelfond, Logic programming and knowledge representation. J. Log. Program., 19/20, 1994, 73-148, (1980), http://citeseer.ist.psu.edu/baral94logic.html
8. Georg Gottlob, Complexity results for nonmonotonic logic. Journal of Logic and Computation 2:(3), pp.397-425, 1992
9. Marco Cadoli, Marco Schaerf, A survey of complexity results for non-monotonic logics. Journal of Logic Programming 17, pp. 127-160, 1993; http://citeseer.ist.psu.edu/cadoli93survey.html
10. Henry A. Kautz, Bart Selman, Hard problems for simple default logics. Artif. Intell., 49, 1-3, 1991, 243-279; http://citeseer.ist.psu.edu/kautz89hard.html

11. Marco Cadoli, Francesco M. Donini, Marco Schaerf: Is intractability of non-monotonic reasoning a real drawback? Artif. Intell. 88(1-2): 215-251 (1996); http://citeseer.csail.mit.edu/20108.html
12. Marco Cadoli, Francesco M. Donini, A survey on knowledge compilation. AI Commun., 10, 3-4, 1997, 137-150; http://citeseer.ist.psu.edu/cadoli98survey.html
13. Sylvie Coste-Marquis, Jerôme Lang, Paolo Liberatore, Pierre Marquis, Expressive power and succintness of propositional languages for preference representation. Principles of Knowledge representation and reasoning: Proceedings of the Ninth International Conference (KR 2004); eds.: D.Dubois, C.Welty, M-A. Williams; AAAI Press 2004.
14. Eiter, Gottlob,Leone, Abduction from logic program: semantics and complexity. Theoretical computer science, Vol. 189 , 1-2 (December 1997); http://citeseer.ist.psu.edu/eiter98abduction.html
15. Thomas Eiter, Georg Gottlob, Nicola Leone: Semantics and complexity of abduction from default theories. Artif. Intell. 90(1-2): 177-223 (1997), pp. 129 - 177, 1997; http://www.kr.tuwien.ac.at/staff/eiter/et-archive/
16. V. Wiktor Marek, Anil Nerode, Jeffrey B. Remmel: A theory of nonmonotonic rule systems I. Ann. Math. Artif. Intell. 1: (1990)
17. Martin Lang, Abdukcia bez podmienok konzistentnosti, Zmalosti 2005, 250-257
18. Michael Gelfond, Vladimir Lifschitz, The stable model semantics for logic programming, ICLP/SLP, 1988, 1070-1080
19. Michael Gelfond, Vladimir Lifschitz, Logic programs with classical negation, ICLP, 1990, 579–597, MIT Press
20. Michael Gelfond, Representing knowledge in A-Prolog, Computational Logic: Logic Programming and Beyond 2002: 413-451; www.cs.ttu.edu/ mgelfond/papers/surv2.p
21. Michael Gelfond, Nicola Leone, Logic programming and knowledge representation – the A-Prolog perspective, Artif. Intell., 138, 1-2, 2002, 3-38
22. Chitta Baral, Knowledge representation, reasoning and declarative problem solving, Cambridge University Press, 2003, ISBN: 0521818028, 544 pp.
23. Ilka Niemela, Patrick Simons, Smodels – an implementation of the stable model and well-founded semantics for normal logic programs. In Proc. LPNMR 1997
24. Smodels, http://www.tcs.hut.fi/Software/smodels/
25. Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Francesco Calimeri, Tina Dell'Armi, Thomas Eiter, Georg Gottlob, Giovambattista Ianni, Giuseppe Ielpa, Christoph Koch, Simona Perri, Axel Polleres: The DLV system. JELIA 2002: 537-540
26. The dlv project, http://www.dbai.tuwien.ac.at/proj/dlv/
27. M. Nogueira, M. Balduccini, M. Gelfond, R. Watson, M. Barry, A-Prolog decision support system for the Space Shuttle. PADL '01: Proceedings of the Third International Symposium on Practical Aspects of Declarative Languages, 2001, 169-183, Springer-Verlag
28. Thomas Eiter, Data Integration and Answer Set Programming. LPNMR 2005: Springer, 13-25
29. Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, Hans Tompits, Combining answer set programming with description logics for the semantic web. Proc. of KR 2004: 141-151
30. Thomas Eiter, Wolfgang Faber, Nicola Leone, Gerald Pfeifer, Axel Polleres: Answer set planning under action costs. J. Artif. Intell. Res. (JAIR) 19: 25-71 (2003)
31. Thomas Eiter, Wolfgang Faber, Nicola Leone, Gerald Pfeifer: The diagnosis frontend of the dlv system. AI Commun. 12(1-2): 99-111 (1999)
32. José Júlio Alferes, João Alexandre Leite, Luís Moniz Pereira, Halina Przymusinska, Teodor C. Przymusinski: Dynamic logic programming. Proc. of KR 1998: 98-111
33. João Alexandre Leite, José Júlio Alferes, Luís Moniz Pereira: Multi-dimensional dynamic knowledge representation. Proc. of LPNMR 2001, Springer, 365-378

34. João Alexandre Leite, Evolving knowledge bases. IOS Press, 2003

Annotation:

Representation of incomplete and inconsistent knowledge

Design and implementation of a non-trivial knowledge-based system tackles a task of mastering (notoriously) incomplete and – maybe – inconsistent knowledge, especially in the case of dynamically evolving domain/knowledge. This talk is devoted to the logic-based research in knowledge representation theory, which is relevant to the topic of reasoning with incomplete (and/or inconsistent) knowledge. Semantic problems are emphasized, but the viewpoint of computability cannot be ignored.

Our attention is devoted to nonmonotonic rule systems, abduction, stable model semantics of logic programs and answer set programming, dynamic logic programming, computational complexity of non-monotonic reasoning and knowledge compilation.