

Rprezentácia znalostí, klasická logika, nemonotónnosť

Klasická (deduktívna) logika nie je vhodným nástrojom na reprezebtáciu znalostí. Pre netriviálnu reálnu doménu je pochybný cieľ sformulovať takú bázu znalostí (také axiómy), z ktorej by sa dali vydedukovať (z ktorej by logicky vyplývali) všetky zaujímavé, potrebné závery. Na získanie takýchto záverov potrebujeme pracovať s **výnimkami**, usudzovať z **neúplného** poznania. Charakter takého usudzovania je nemonotónny.

Pri voľbe prostriedkov reprezentácie znalostí sa vyvažuje **vyjadriteľnosť** a výpočtová **efektívnosť**. Pod reprezentáciou znalostí rozumieme štvoricu (J, E, Cn, Ind). V tomto kontexte je zaujímavý jazyk J a operátor odvodenia Cn. Ak nemáme ambíciu, aby reprezentácia mala veľkú vyjadrovaciu silu, môžeme zvoliť J a Cn tak, aby sme dokázali počítať efektívnejšie. V opačnom prípade sme nútení zľaviť z nárokov na výpočtovú efektívnosť. Demonštrovali sme si to na príklade relačných databáz, deduktívnych databáz (datalogu) a nelimitovanej dedukcie v prvorádovom jazyku.

Zavedenie negácie do jazyka definitných logických programov nutne vedie k **nemonotónnosti**, ak chceme aby operátor odvodenia mal rozumnú vlastnosť slabej korektnosti. Podobné zistenie platí pre ľubovoľný jazyk, v ktorom chceme odvodiť viac ako dedukciou - teda, ak operátor odvodenia nie je korektný. Ak tento operátor odvodenia má zachovávať konzistentnosť, musí byť nemonotónny.

Z histórie

Nemonotónne usudzovanie sa začalo študovať koncom 70. rokov minulého storočia. Hlavné formalizmy, ktoré boli navrhnuté: cirkumskripcia, defaultová logika a autoepistemická logika (a mnoho ďalších). Vychádzali z rôznych motivácií a boli postavené na odlišných konštrukciách.

Logické programovanie sledovalo iné ciele. Jeho ambíciou bolo postaviť programovanie na ydravých logických zásadách, úlohou programátora mala byť logická charakterizácia problému a jeho riešenia, úlohou výpočtovej mašinerie jej realizácia. Kowalski to vzjadril sloganom algorithm = logic + control. Ďalšou motiváciou pre rozvoj logického programovania bola ambícia rozšíriť relačné databázy (deduktívne databázy).

Časom sa ukázalo, že logické programovanie je aj vhodným formalizmom na štúdium nemonotónneho usudzovania. V 80. rokoch sa dosiahli významné výsledky, ktoré ukázali ekvivalenciu logického programovania a základných nemonotónnych formalizmov (vzhľadom na isté sémantiky). Dá sa povedať, že logické programovanie sa stalo dominantným formalizmom na štúdium nemonotónneho usudzovania a reprezentácie znalostí. Možno to pripísať jednak jeho formálnym vlastnostiam, istež jednoduchosti a aj výhodnosti z ako implementačného nástroja.

Defaultové teórie

Pravidlá defaultovej logiky (na rozdiel od pravidiel dedukcie) nie sú štrukturálne, neaplikujeme ich lokálne, majú doménovo-špecifický význam. V podstate reprezentujú nejaké hypotézy. Kľúčovým pojmom je extenzia defaultovej teórie. Intuitívne to je nasýtená a koherentná množina viet, hypotéz, ktoré dokopy dávajú zmysel. Na jej presnú definíciu sa používa často používaná a účinná **fixpointová** konštrukcia.

Výpočet extenzie defaultovej teórie je pekným príkladom **nedeterministického** uvažovania, typického pre nemonotónne logiky i pre intuitívne hypotetické uvažovanie (dáva táto množina hypotéz zmysel? skúsím inú? môže ísť o sériu pokusov a omylov).

Nekumulatívnosť defaultového usudzovania - dôsledok alternatívnosti. Dobré vlastnosti normálnych defaultových teórií - menšia vyjadriteľnosť, lepsie vlastnosti (zarúčená existencia extenzie, semimonotónnosť, existuje dôkazová procedúra, zisťujúca prislusnosť nejakej formuly do nejakej extenzie).

Reprezentácia znalostí pomocou normálnych defaultových pravidiel môže viesť k istým problémom. Nikedz sa objavujú neintuitívne extenzie. Do jednej extenzie sa môžu dostať dôsledky defaultových pravidiel, ktorých stráže nie sú konzistentné. Riešením sú v prvom prípade semi=normálne defaultové teórie, v druhom prípade obmedzované extenzie.

Pre cieľ reprezentácie znalostí je vhodné zaviesť na defaultových pravidlách preferencie. Ďalším možným rozšírením je zavedenie viacerých kontextov, ich reprezentácia defaultovými teóriami a toku informácií z kontextu do kontextu defaultovými pravidlami (bridge rules) .

Stabilné modely - sémantika logických programov

Zavedenie negácie ako konečného zlyhania do logických programov s cieľom zvýšiť vyjadrovaciu silu viedlo k sémantickým problémom. Jedným z ich riešení je sémantika stabilných modelov, ktorú možno opvažovať za v súčasnosti najrozšírenejší prostriedok na reprezentáciu znalostí logickými programami. Opäť ide o nedeterministickú fixpointovú konštrukciu.

Parakonzistentná logika

Niekedy je potrebné pracovať s takým logickým formalizmom, ktorý nemožno odvodiť z nekonzistentnosti čokoľvek (tým aj nie je nutné okamžite po identifikovaní nekonzistentnosti robiť revízie). Zoznámili sme sa s takým formalizmom. Neuspokojuje nás, že v jeho rámci nemožno používať modus ponens. Pokúsili sme sa modus ponens zachrániť tým, že sme za preferované modely zvolili tie, čo majú najmenší stupeň konfliktnosti. Dôsledkom zavedenia preferenčného vyplývania je nemonotónnosť a závislosť na kontexte.

Argumentácia (ako metóda dôkazu i operacionálna sémantika)

Abstraktný argumentačný framework je založený na kľúčovej myšlienke, že rozumná argumentácia je založená na prijatí množiny argumentov, ktoré sa dokopy prejavujú nejakými dobrými vlastnosťami.

Pohľad na (akékoľvek) nemonotónne usudzovanie ako na argumentáciu a kontraargumentáciu.