



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky



**Ing. Alexander Šimko**

Autoreferát dizertačnej práce

# Logické programovanie s preferenciami na pravidlách

na získanie akademického titulu philosophiae doctor

v odbore doktorandského štúdia: 9.2.1 Informatika

Bratislava, 2014

Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia na Katedre aplikovanej informatiky Fakulty matematiky, fyziky a informatiky Univerzity Komenského v Bratislave.

**Predkladateľ:** Ing. Alexander Šimko  
Katedra aplikovanej informatiky  
Fakulta matematiky, fyziky a informatiky  
Univerzita Komenského  
Mlynská dolina  
842 48 Bratislava

**Školiteľ:** doc. PhDr. Ján Šefránek, CSc.  
Katedra aplikovanej informatiky  
Fakulta matematiky, fyziky a informatiky  
Univerzita Komenského  
Mlynská dolina  
842 48 Bratislava

**Oponenti:**

Obhajoba dizertačnej práce sa koná ..... o ..... h  
pred komisiou pre obhajobu dizertačnej práce v odbore doktorandského štúdia vymeno-  
vanou predsedom odborovej komisie .....

9.2.1 Informatika

na .....

Predseda odborovej komisie:

.....

# 1 Úvod

Logické programovanie so sémantikou stabilných modelov [29] sa stalo populárnym deklaratívnym programovacím a reprezentačným jazykom. Bolo aplikované v mnohých oblastiach, ako napríklad: podpora rozhodovania [43, 3], konfigurácia produktov [63], plánovanie [12, 36], diagnostika [16, 27] a mnoho iných [53].

Logický program pozostáva z pravidiel typu ak-potom. Preferencia na pravidlách bola identifikovaná ako dôležitý koncept v reprezentácii znalostí. V prípade konfliktných pravidiel nám preferencia umožňuje vybrať, ktoré z pravidiel sa použije. Napríklad, zvykneme preferovať informácie od blízkeho priateľa než od cudzinca. Ďalším príkladom je známy problém s nelietajúcimi vtákmi: Vtáky lietajú. Tučniaky ale nelietajú. Tweety je tučniak. Nakoľko je druhé pravidlo založené na špecifickejšej informácii, preferujeme ho. V právnej doméne sú konflikty medzi zákonmi riešené podľa princípov. Napríklad, zákon s väčšou silou alebo novší zákon je použitý.

Na rozšírenie logického programovania o preferencie na pravidlách bolo navrhnutých mnoho prístupov. Bežným prístupom je reprezentovať preferencie na pravidlách pomocou relácie na pravidlách. Mnoho sémantik pre logické programy s preferenciami bolo navrhnutých. Spomenieme napríklad Brewkove a Eiterove preferované a slabo preferované stabilné modely [6], poradie zachovávajúce stabilné modely [11], teóriu stabilných bodov [68], Zhangove a Foove stabilné modely pre prioritizované logické programy [70], Šefránkov argumentačný prístup k preferovaným stabilným modelom [57], Gabaldonove preferované stabilné modely [22] a Sakamovu a Inoueho transformáciu na preferencie na literáloch [51]. Dôležitým smerom vo výskume preferencií je štúdium princípov pre usudzovanie s preferenciami. Čiastočne sa tomuto výskumu venujú práce [6, 57, 58, 11]. Princípy sú kľúčové vlastnosti sémantik, ktoré môžeme použiť na vyhodnotenie zdravosti sémantik a na výber vhodnej sémantiky pre riešenie úlohu.

Nasledujúce tri hlavné dôvody robia logické programovanie s preferenciami na pravidlách konceptuálne ťažkým:

- Pojem „preferencia“ je abstraktný a zastupuje iný pojem s intuitívnym významom, napríklad špecifickosť, silu zákona, dôveru a pod. Je prirodzené očakávať, že rôzne štýly usudzovania a sémantiky s rôznymi vlastnosťami sú potrebné pre rôzne interpretácie tohto pojmu.
- Mnoho sémantik pre logické programovanie s preferenciami bolo navrhnutých. Každá je zvyčajne definovaná úplne iným spôsobom. Nie je vždy zrejmé, pre ktorú interpretáciu preferencií je sémantika použiteľná.
- Vlastnosti existujúcich sémantik boli študované. Žiadna systematická analýza však nebola vykonaná. Napríklad je známe, že prístup Delgranda a kolegov [11] spĺňa Brewkov a Eiterov Princíp I, nie je ale známe, či tento prístup spĺňa alebo porušuje sémantika Zhanga a Foa [70].

Nazeranie na preferencie na pravidlách ako na mechanizmus zápisu výnimiek medzi pravidlami má výhody: (i) Poskytuje kompaktnjší a čitateľnejší zápis ako ručne zakódovanie výnimiek. (ii) Zápis je tolerantnejší na zmeny. (iii) Vlastnosti sémantiky môžu byť formálne študované. (iv) Sú situácie, v ktorých ručné zakódovanie nie je možné, napríklad ak dynamické preferencie alebo automaticky generované preferencie sú potrebné. Ak chápeme preferencie na pravidlách ako mechanizmus zápisu výnimiek medzi pravidlami, Brewkova a Eiterov Princíp I by mal byť splnený. Zachytáva základnú intuíciu preferencií. Ak sú dva stabilné modely generované rovnakými pravidlami až na dve pravidla, stabilný model generovaný menej preferovaným pravidlom nie je akceptovaný. Avšak existujúce prístupy, ktoré spĺňajú Princíp I, majú nasledovné nedostatky:

- Vnášajú imperatívne črty do inak deklaratívneho jazyka. Chápu preferencie ako poradie, v ktorom sa majú pravidla programu aplikovať. To ide proti deklaratívnemu charakteru pôvodného logického programovania.
- Majú problémy ignorovať preferencie medzi nekonfliktnými pravidlami. Ak chápeme preferencie ako mechanizmus zápisu výnimiek medzi konfliktnými pravidlami, preferencie medzi nekonfliktnými pravidlami by nemali zohrávať žiadnu rolu. Existujú situácie, v ktorých je schopnosť ignorovať preferencie medzi nekonfliktnými pravidlami kľúčová, napríklad, ak preferencie na pravidlách musia byť automaticky generované.

## 2 Ciele práce

Cieľmi práce je:

- Poskytnúť bázu, ktorá by uľahčovala výber sémantiky pre logické programovanie s preferenciami na pravidlách, a to:
  - Rozpracovať princípy pre logické programovanie s preferenciami na pravidlách. Sumarizovať existujúce princípy, navrhnúť nové princípy získané z existujúcich sémantik alebo založené na našej intuícii a preskúmať vzťahy medzi princípmi.
  - Vyhodnotiť existujúce sémantiky pomocou princíпов. t.j. pre každý princíp a každú sémantiku poskytnúť dôkaz, či daný princíp spĺňa alebo porušuje
  - Pre chápanie preferencií ako mechanizmu na zápis výnimiek medzi pravidlami navrhnúť požiadavky na sémantiky, vrátane stanovenia ktoré princípy majú byť splnené a ktoré porušené.
  - Vyhodnotiť existujúce sémantiky vzhľadom na požiadavky. Ako si neskôr ukážeme, žiadna z existujúcich sémantik plne nespĺňa stanovené požiadavky.

- Navrhnuť sémantiku pre logické programovanie s preferenciami na pravidlách spĺňajúcu všetky stanovené požiadavky, vrátane toho, že: (i) sémantika je kompatibilná so sémantikou stabilných modelov, (ii) sémantika je plne deklaratívna, (iii) sémantika berie do úvahy priame a aj nepriame konflikty, (iv) sémantika ignoruje preferencie medzi nekonfliktnými pravidlami.
  - Formálne preskúmať vlastnosti novo navrhutej sémantiky.
  - Vytvoriť prototypovú implementáciu sémantiky, ktorá by slúžila na experimentovanie so sémantikou.

## 3 Dosiiahnuté výsledky

### 3.1 Princípy a požiadavky na sémantiku

V Kapitole 5 sa venujeme princípom a požiadavkám na sémantiku logického programovania s preferenciami na pravidlách. Zhrňame existujúce princípy, navrhujeme nové princípy a skúmame vzťahy medzi nimi.

#### Existujúce princípy

- Princíp I [6] sa snaží zachytiť hlavnú intuíciu za preferenciami. Ak dva stabilné modely sú generované rovnakými pravidlami až na dve pravidla, stabilný model generovaný menej preferovaným pravidlom nie je akceptovaný.
- Princíp II [6] hovorí, že preferencie špecifikované na pravidlách s nesplneným pozitívnym telom nemôžu zmeniť preferovaný stabilný model na nepreferovaný.
- Princíp III požaduje, že program má preferovaný stabilný model, ak má štandardný stabilný model. Vyjadruje pohľad, že preferencie by nám mali pomôcť vo výbere preferovaných stabilných model, ale nie znemožniť výber.

Nasledujúce dva princípy sú v existujúcej literatúre zvyčajne študované vlastnosti sémantik.

- Preferencie slúžia na výber spomedzi alternatívne použiteľných pravidiel, ktoré spôsobujú, že program má viacero stabilných modelov. Princíp IV požaduje, že pridávanie preferencií nezväčšuje počet možností.
- Ak relácia preferencie je prázdna, potom máme štandardný logické program. Princíp V požaduje, že preferované stabilné modely sú v tomto prípade zhodné so štandardnými stabilnými modelmi.

## Nové princípy

Jednou z ústredných otázok usudzovania s preferenciami je: Ktoré preferencie môžeme ignorovať, a ktoré musíme brať do úvahy?

- Princíp VI. Ak akceptujeme pohľad, že sémantika by mala byť schopná ignorovať preferencie medzi nekonfliktnými pravidlami, je zmysluplné očakávať, že sémantika preferovaných stabilných modelov je zhodná so sémantikou stabilných modelov pre triedu stratifikovaných programov. Stratifikované programy neobsahujú cyklickú defaultovú negáciu, a teda neobsahujú žiaden konflikt.
- Princíp VII je postačujúca podmienka, inšpirovaná alternatívnymi definíciami sémantik [11, 68, 6], stanovujúca kedy je stabilný model preferovaný. Taktiež hovorí o ignorovaní preferencií. Ak všetky preferované pravidlá nie sú aplikovateľné v stabilnom modeli lebo ich prerekvizity nie sú odvoditeľné v stabilnom modeli, môžeme použiť menej preferované pravidlá na odvodenie stabilného modelu.
- Princíp VIII, extrahovaný zo sémantiky [51], hovorí, že preferencie na pravidlách, ktoré nie sú použité na generovanie stabilných modelov by mali byť ignorované.

Zatiaľ žiadna sémantika spĺňajúca Princíp I a Princíp III nebola definovaná, a to aj napriek snahám viacerých výskumníkov. Ukázali sme, že Princíp I a Princíp III sú kompatibilné. To znamená, že výskumníci museli intuitívne akceptovať ďalšie princípy, ktoré im znemožnili zdefinovať takú sémantiku.

- Navrhli sme jeden z takýchto princíпов, označujeme ho ako Princíp IX. Neformálne, princíp hovorí, že: ak jediný spôsob ako zablokovať pravidlo je jediné menej preferované konfliktné pravidlo, potom musí byť preferovanejšie pravidlo použité.

Ukázali sme, že akceptovanie Princípu IX má ďalekosiahle dôsledky: Princíp I je automaticky splnený a Princíp III je porušený.

Čiže Princíp III a Princíp IX reprezentujú nekompatibilné pohľady na preferencie na pravidlách. V časti 5.5 sme ukázali program s preferenciami a dve interpretácie preferencií, jednu kompatibilnú s Princípom III a druhú kompatibilnú s Princípom IX. Navyše, každý z existujúcich prístupov [6, 11, 68, 70, 57, 51] spĺňa buď Princíp III alebo Princíp IX. Dichotómia Princípu III a Princípu IX je dôležitým výsledkom, nakoľko môže byť použitá ako prvý krok pri výbere sémantiky vhodnej pre danú doménu.

### 3.2 Požiadavky na sémantiky

V práci sa sústreďujeme na chápanie preferencií ako na mechanizmu zápisu výnimiek medzi pravidlami: *Menej preferované konfliktné pravidlo nemôže definovať výnimku preferovanejšiemu konfliktnému pravidlu*. Navrhujeme, že sémantika pre takého chápanie preferencií by mala

- byť plne deklaratívna, čím by mala nasledovať štýl pôvodného logického programovania,
- spĺňať Princíp IX, ktorý formalizuje hlavnú intuíciu za chápaním preferencií ako mechanizmu na zápis výnimiek medzi pravidlami,
- byť kompatibilná so sémantikou stabilných modelov, a teda spĺňať Princíp V, ktorý požaduje, že v prípade absencií preferencií, je sémantika ekvivalentná sémantike stabilných modelov,
- spĺňať Princíp IV – pridávanie preferencií by nikdy nemalo zväčšiť počet preferovaných stabilných modelov, nakoľko menej preferované konfliktné pravidlá nedefinujú výnimky preferovanejším konfliktným pravidlám,
- byť schopná ignorovať preferencie medzi nekonfliktnými pravidlami, konkrétne by mala
  - spĺňať Princíp VI – sémantika je ekvivalentná sémantike stabilných modelov pre stratifikované programy, keďže tie neobsahujú konflikty,
  - spĺňať Princíp VII – preferencie na pravidlách s nesplnenými prerekvizitami nehrajú žiadnu rolu, nakoľko pravidlá nemôžu byť použité,
- preferovane pracovať aj s nepriamymi konfliktami

Na základe týchto požiadaviek automaticky získavame, že sémantika by taktiež mala

- spĺňať Princíp II práve vtedy, keď pracuje iba s priamymi konfliktami,
- spĺňať Princíp I, a
- porušovať Princíp III a Princíp VIII.

### 3.3 Vyhodnotenie sémantik

V práci sme vykonali úplné vyhodnotenie existujúcich sémantik na princípoch, t.j. pre každý princíp a každú sémantiku sme dokázali, či sémantika spĺňa alebo porušuje daný princíp. Význam tohto vyhodnotenia spočíva v tom, že poskytuje základňu, ktorú môžeme použiť pri výbere vhodnej sémantiky pre riešenie úlohu.

Tabuľka 1 zobrazuje výsledky vyhodnotenia sémantik na princípoch. Taktiež vyhodnocuje, či sémantiky spĺňajú požiadavky pre chápanie preferencií ako mechanizmu na zápis výnimiek medzi pravidlami.  $\mathcal{PAS}_{BE}$  označuje Brewkovu a Eiterovu sémantiku preferovaných stabilných modelov [6],  $\mathcal{PAS}_{BEW}$  označuje Brewkovu a Eiterovu sémantiku slabo preferovaných stabilných modelov,  $\mathcal{PAS}_{DST}$  poradie zachovávajúce stabilné modely Delgranda a kolegov [11],  $\mathcal{PAS}_{WZL}$  preferované stabilné modely Wanga a kolegov [68],  $\mathcal{PAS}_{ZF}$  Zhangove a Foove stabilné modely pre prioritizované logické programy [70] a  $\mathcal{PAS}_G$  Sakamove a Inoueho preferované stabilné modely [51].

	d/g	I	II	III	IV	V	VI	VII	VIII	IX
$\mathcal{PAS}_{BE}$	d	s[6]	s[6]	v[6]	s[6]	s[6]	v	s	v	s
$\mathcal{PAS}_{BEW}$	d	v[6]	v[6]	s[6]	v	s	s	s	v	v
$\mathcal{PAS}_{DST}$	g	s[11]	s[11]	v	s	s[11]	v	v	v	s
$\mathcal{PAS}_{WZL}$	g	s	s	v	s	s[68]	v	v	v	s
$\mathcal{PAS}_{ZF}$	g	v	v[6]	s[6]	v	s	s	v	v	v
$\mathcal{PAS}_{SI}$	g	v	v	s[51]	v[51]	s[51]	s	v	s	v

Tabuľka 1: Princípy a požiadavky splnené a porušené existujúcimi sémantikami. Druhý stĺpec označený „d/g“ zobrazuje, či sémantika spracuje s: „d“ – iba priamymi konfliktami, alebo „g“ – všeobecnými konfliktami vrátane priamych a nepriamych. Jednotlivé bunky tabuľky sú označené nasledovne: písmeno „s“ znamená, že princíp je splnený, písmeno „v“ znamená, že princíp je porušený, zelené pozadie znamená, že požiadavka je splnená a červené pozadie znamená, že požiadavka je porušená.

Ako môžeme z tabuľky vidieť, žiaden zo skúmaných prístupov plne nespĺňa stanovené požiadavky.

### 3.4 Sémantiky

V ďalšej časti práce sa venujeme návrhu sémantiky, ktorá by vylepšovala existujúce sémantiky, hlavne aby spĺňala všetky stanovené požiadavky.

Najprv v sekcii 6.9 práce ukazujeme, že sémantika spĺňajúca Princíp IX, ktorý je dôležitý pri chápaní preferencií ako mechanizmu zápisu výnimiek medzi pravidlami, nemôže byť definovaná použitím základných myšlienok prístupov [51, 70]. Po prvé, preferované stabilné modely nemôžeme definovať ako porovnávanie generujúcich množín stabilných modelov, a to bez ohľadu na to, ako definujeme reláciu preferencie na stabilných modeloch. Po druhé, nemôžeme definovať preferované stabilné modely prostredníctvom nedeterministického odstraňovania menej preferovaných pravidiel a zobrať stabilných modelov zvyškov programu za predpokladu, že všetky maximálne pravidla sú ponechané.

#### 3.4.1 Sémantika pre priame konflikty

V Kapitole 7 sa najprv venujeme sémantike, ktorú označujeme ako  $\mathcal{PAS}_D$ , a ktorá je založená na jednoduchšej myšlienke. Pridávanie defaultovo negovaných literálov do tiel pravidiel mení nezávislé pravidlá na pravidlá definujúce výnimky a pravidlá definujúce výnimky na konfliktné pravidlá. Základnou myšlienkou je chápať usudzovanie s preferenciami ako opačnú transformáciu: *konfliktné pravidlá sú zmenené na pravidla definujúce výnimky*. Transformácia je realizovaná odstránením takých defaultovo negovaných literálov z tela pravidla, ktoré sú odvádzané menej preferovanými konfliktnými pravidlami. Musíme si ale dávať pozor na situáciu, keď je ten istý literál odvádzaný viacerými pravid-



lami. Riešenie spočíva v použití pomocných literálov, pomocou ktorých reprezentujeme, že pravidlo bolo použité a nahradíme literály v negatívnych telách pravidiel týmito pomocnými literálmi.

V práci ďalej ukazujeme, ako môžeme sémantiku danú touto transformáciou definovať priamo. Myšlienka transformácie sa dá vyjadriť nasledovne: *pravidlo nemôže byť zamietnuté menej preferovaným konfliktným pravidlom*. Za týmto účelom sme navrhli alternatívnu definíciu sémantiky stabilných modelov. Sleduje štruktúru pôvodnej Gelfondovej a Lifschitzovej definície, ale je založená na množine pravidiel na rozdiel od množín literálov. Tipne sa množina pravidiel, ktorá generuje stabilný model. Redukt programu sa získa z programu odstránením každého pravidla, ktoré je porazené nejakým pravidlom z nášho typu. Táto alternatívna definícia umožňuje elegantným spôsobom definovať sémantiku pre logické programy s preferenciami pridaním jednoduchej podmienky do definície reduktu: *pravidlo je odstránené ak v našom tipe existuje pravidlo, ktoré ho poráža, ale nie je menej preferované a konfliktné*.

Táto sémantika spĺňa všetky stanovené požiadavky, vrátane schopnosti ignorovať preferencie medzi nekonfliktnými pravidlami, s výnimkou toho, že pracuje iba s priamymi konfliktmi.

### 3.4.2 Sémantika pre všeobecné konflikty

V Kapitole 8 sme navrhli sémantiku  $\mathcal{PAS}_G$ , ktorá pracuje so všeobecnými konfliktmi (priame aj nepriame), zovšeobecnením myšlienky stojacej na pozadí sémantiky  $\mathcal{PAS}_D$ . Ukázalo sa, že nie je jednoduché definovať, kedy sú dve pravidla všeobecne konfliktné. Elegantne sme sa vyhli tomuto problému presunutím sa od pravidiel na množiny pravidiel a zadefinovali kedy sú dve množiny pravidiel konfliktné. Opäť sme navrhli alternatívnu definíciu sémantiky stabilných modelov, tentokrát postavenú na fragmentoch – množinách pravidiel, ktoré môžu tvoriť jednu stranu konfliktu. Alternatívna definícia sémantiky stabilných modelov opäť nasleduje štruktúru Gelfondovej a Lifschitzovej definície. Tipne sa množina fragmentov a vypočíta sa redukt programu vzhľadom na tento tip – zo všetkých možných fragmentov odstránime tie fragmenty, ktoré sú porazené inými fragmentami v našom tipe. Sémantiku pre logické programy s preferenciami na pravidlách získavame opäť malou zmenou: *fragment odstránime iba vtedy, ak porážajúci fragment nie je konfliktný a založený na menej preferovaných pravidlách*.

Táto sémantika spĺňa všetky stanovené požiadavky, vrátane toho, že pracuje so všeobecnými konfliktmi a je schopná ignorovať preferencie medzi nekonfliktnými pravidlami.

### 3.4.3 Dve relaxácie pre všeobecné konflikty

Definícia sémantiky  $\mathcal{PAS}_G$  operuje s potenčnou množinou programu. Stanovili sme hornú hranicu výpočtovej zložitosti problému  $\mathcal{PAS}_G(\mathcal{P}) \neq \emptyset$  – tretia úroveň polynomickej hierarchie. Určenie dolnej hranice ostáva zatiaľ medzi otvorenými problémami. V Kapitole 9

sa preto sústredíme na dve relaxácie. Sémantika  $\mathcal{PAS}_{NO}$  sa vzdáva požiadavky na ignorovanie preferencií medzi nekonfliktnými pravidlami. Je založená na myšlienke, že *pravidlo nemôže byť zamietnuté menej preferovaným pravidlom alebo pravidlom závislým na menej preferovanom pravidle*. Sémantika spĺňa všetky požiadavky okrem schopnosti ignorovať preferencie medzi nekonfliktnými pravidlami. Sémantika  $\mathcal{PAS}_{CNF}$  sa zameriava na podtriedu konfliktov. Nepriame konflikty môžu byť tvorené iba pravidlami používajúcimi stratifikovanú negáciu. Vďaka tomuto zjednodušeniu definujeme, kedy sú dve pravidla konfliktné. Sémantika spĺňa všetky požiadavky v výnimkou toho, že je navrhnutá pre podtriedu programov. Vďaka uvedeným zjednodušeniam sú oba rozhodovacie problémy  $\mathcal{PAS}_{NO}(\mathcal{P}) \neq \emptyset$  a  $\mathcal{PAS}_{CNF}(\mathcal{P}) \neq \emptyset$  NP-úplné.

### 3.5 Vyhodnotenie nových sémantik

Tabuľka 2 rekapituluje princípy splnené a porušené existujúcimi sémantikami spolu s nami navrhnutými sémantikami. Taktiež vyhodnocuje sémantiky z pohľadu požiadaviek.

	d/g	I	II	III	IV	V	VI	VII	VIII	IX
$\mathcal{PAS}_{BE}$	d	s[6]	s[6]	v[6]	s[6]	s[6]	v	s	v	s
$\mathcal{PAS}_{BEW}$	d	v[6]	v[6]	s[6]	v	s	s	s	v	v
$\mathcal{PAS}_{DST}$	g	s[11]	s[11]	v	s	s[11]	v	v	v	s
$\mathcal{PAS}_{WZL}$	g	s	s	v	s	s[68]	v	v	v	s
$\mathcal{PAS}_{ZF}$	g	v	v[6]	s[6]	v	s	s	v	v	v
$\mathcal{PAS}_{SI}$	g	v	v	s[51]	v[51]	s[51]	s	v	s	v
$\mathcal{PAS}_D$	d	s	s	v	s	s	s	s	v	v
$\mathcal{PAS}_G$	g	s	v	v	s	s	s	s	v	v
$\mathcal{PAS}_{NO}$	g	s	s	v	s	s	v	s	v	v
$\mathcal{PAS}_{CNF}$	cnf	s	v	v	s	s	s	s	v	v

Tabuľka 2: Princípy a požiadavky splnené a porušené existujúcimi sémantikami. Druhý stĺpec označený „d/g“ zobrazuje, či sémantika spracuje s: „d“ – iba priamymi konfliktami, „g“ – všeobecnými konfliktami vrátane priamych a nepriamych, alebo „cnf“ – podtriedou všeobecných konfliktov, kde nepriame konflikty sú tvorené cez pravidla používajúce iba stratifikovanú negáciu. Jednotlivé bunky tabuľky sú označené nasledovne: písmeno „s“ znamená, že princíp je splnený, písmeno „v“ znamená, že princíp je porušený, zelené pozadie znamená, že požiadavka je splnená, červené pozadie znamená, že požiadavka je porušená, a oranžové pozadie znamená, že požiadavka je čiastočne splnená.

$\mathcal{PAS}_G$  spĺňa všetky požiadavky.  $\mathcal{PAS}_D$  a  $\mathcal{PAS}_{CNF}$  spĺňajú všetky požiadavky s výnimkou toho, že  $\mathcal{PAS}_D$  pracuje iba s priamymi konfliktmi a  $\mathcal{PAS}_{CNF}$  s podtriedou všeobecných konfliktov.

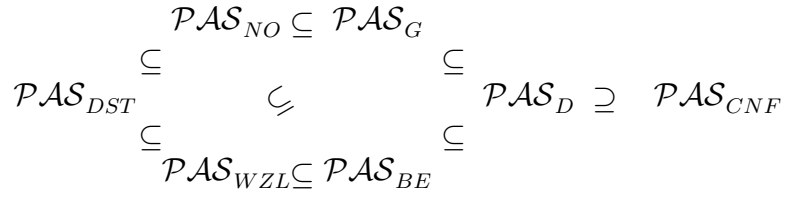
V kontraste s  $\mathcal{PAS}_{CNF}$ , sémantika  $\mathcal{PAS}_{NO}$  porušuje Princíp VI. Reflektuje to relaxácie, ktoré boli v sémantikách vykonané.  $\mathcal{PAS}_{CNF}$  dovoľuje aby boli nepriame konflikty

vytvorené iba cez podtriedu pravidiel a je schopná ignorovať preferencie medzi nekonfliktnými pravidlami. Na druhej strane  $\mathcal{PAS}_{NO}$  umožňuje nepriame konflikty vytvorené cez ľubovoľné pravidlá, ale nie je vždy schopná ignorovať preferencie medzi nekonfliktnými pravidlami.

### 3.6 Porovnanie

V Kapitole 11 skúmame v akom vzájomnom vzťahu sú nami navrhnuté sémantiky a aký je vzťah medzi našimi sémantikami a existujúcimi sémantikami. Ukázali sme, že  $\mathcal{PAS}_D$  pokračuje v hierarchii sémantik  $\mathcal{PAS}_{DST} \subseteq \mathcal{PAS}_{WZL} \subseteq \mathcal{PAS}_{BE}$  objavenej v [56]. Taktiež sémantiky  $\mathcal{PAS}_{DST}$ ,  $\mathcal{PAS}_{NO}$ ,  $\mathcal{PAS}_G$  a  $\mathcal{PAS}_D$  tvoria hierarchiu  $\mathcal{PAS}_{DST} \subseteq \mathcal{PAS}_{NO} \subseteq \mathcal{PAS}_G \subseteq \mathcal{PAS}_D$ , a teda sémantiky tvoria jednu hierarchiu s dvomi vetvami. Celkový pohľad zobrazuje Obr. 1.

Obr. 1: Hierarchia sémantik.



$\mathcal{PAS}_{DST}$  je silne preskriptívna sémantika. Požaduje, že

- pravidlá programu sú aplikované v poradí danom preferenciami,
- pravidlo je použité až keď jeho pozitívne telo je odvodené, a
- pravidlo môže byť porazené iba pravidlom, ktoré nezávisí na menej preferovanom pravidle.

Ako Schaub a Wang [56] komentujú, po hierarchii  $\mathcal{PAS}_{DST} \subseteq \mathcal{PAS}_{WZL} \subseteq \mathcal{PAS}_{BE}$  sa hýbeme zmenou, do akej miery nasledujúce koncepty interagujú: (i) preferencie, a (ii) požiadavka, že pravidlo je použité až keď jeho pozitívne telo je odvodené.

Na druhej strane, zo sémantiky  $\mathcal{PAS}_{DST}$  prechádzame do vetvy  $\mathcal{PAS}_{NO} \subseteq \mathcal{PAS}_G \subseteq \mathcal{PAS}_D$  zahodením požiadavky, že pravidlá sú aplikované v poradí danom preferenciami, čo robí sémantiky plne deklaratívne. Po hierarchii  $\mathcal{PAS}_{NO} \subseteq \mathcal{PAS}_G \subseteq \mathcal{PAS}_D$  sa posúvame zmenou toho, ktorým pravidlám zabránime v porážaní iných pravidiel: (i) od pravidiel závislých na menej preferovaných pravidlách, (ii) cez všeobecne konfliktné menej preferované pravidla, (iii) až po priamo konfliktné menej preferované pravidlá.

Po hierarchii  $\mathcal{PAS}_{CNF} \subseteq \mathcal{PAS}_D$  sa posúvame zmenou toho, ktorým pravidlám zabránime v porážaní iných pravidiel: (i) od konfliktných menej preferovaných pravidiel,

kde nepriame konflikty môžu byť vytvorené iba pravidlami so stratifikovanou negáciou, (ii) po priamo konfliktných menej preferovaných pravidiel.

$\mathcal{PAS}_{CNF}$  stojí mimo hierarchie  $\mathcal{PAS}_{NO} \subseteq \mathcal{PAS}_G \subseteq \mathcal{PAS}_D$  nakoľko je definovaná pre podtriedu programu, v ktorých nepriame konflikty môžu byť tvorené iba pravidlami používajúcimi stratifikovanú negáciu.

### 3.7 Implementácia

V Kapitole 12 implementujeme naše sémantiky pomocou meta-interpretáčnej techniky [18]. Cieľom bolo poskytnúť implementáciu, ktorú môžu výskumníci požiť na experimentovanie so sémantikami. Nebolo cieľom vytvoriť implementáciu vhodnú do produkcie. Za týmto cieľom sme taktiež implementovali nástroj PASS. Na svojom vstupe vezme logický program s preferenciami, zbaví sa premenných, transformuje ho na meta-program a vyhodnotí ho pomocou zvoleného meta-interpretéra.

### 3.8 Význam výsledkov

V tejto práci sme rozpracovali princípy pre logické programovanie s preferenciami na pravidlách. Zosumarizovali sme existujúce princípy, navrhli nové a preskúmali vzťahy medzi nimi. Vyhodnotili sme existujúce a aj nami navrhnuté sémantiky na týchto princípoch. Ktoré princípy by mali byť splnené pri ktorej interpretácii preferencií je veľmi podstatná, no zároveň ťažká otázka. Ukázali sme, že Princíp III a Princíp IX reprezentujú dva nekompatibilné pohľady na preferencie. Všetky existujúce prístupy spĺňajú buď jeden alebo druhý princíp. Pre chápanie preferencií ako mechanizmu zápisu výnimiek sme navrhli požiadavky na sémantiky a prepojili ich so splnením/porušením princíпов. Vytvorili sme tak chýbajúci podklad, na základe ktorého môžeme vybrať sémantiku vhodnú pre riešenie úlohu.

Pre chápanie preferencií ako mechanizmu pre zápis výnimiek sme navrhli sémantiky, ktoré vylepšujú vlastnosti existujúcich sémantik. Pre prípad priamych konfliktov sme navrhli sémantiku  $\mathcal{PAS}_D$  a pre prípad všeobecných konfliktov sémantiku  $\mathcal{PAS}_G$ . Hlavnými vylepšeniami je, že sémantiky sú plne deklaratívne a ignorujú preferencie medzi nekonfliktnými pravidlami, čo umožňuje použitie sémantik v scenároch, v ktorých sú existujúce sémantiky nepoužiteľné, ako napríklad keď sú preferencie medzi pravidlami automaticky generované. Formálne preskúmanie vlastností a vzťahov medzi sémantikami nám poskytlo lepšiu a celistvejšiu pohľad na sémantiky. Sémantiku  $\mathcal{PAS}_D$  môžeme chápať ako plne deklaratívnu alternatívu k sémantike  $\mathcal{PAS}_{BE}$ . Sémantika  $\mathcal{PAS}_{NO}$  je zase plne deklaratívnym proľajškom sémantiky  $\mathcal{PAS}_{DST}$ .

Implementovali sme jednoduchý nástroj, ktorý tieto sémantiky umožňuje vypočítať prostredníctvom meta-interpretácie. Existencia takéhoto nástroja pozitívne prispieva k propagovaniu výsledkov tejto práce a umožňuje jednoduchšie experimentovanie so sémantikami a prvotné zoznámenie sa so sémantikami bez nutnosti študovať ich definície.

## 4 Zoznam relevantných publikácií autora

- Alexander Šimko. Accepting the natural order of rules in a logic program with preferences. In *Technical Communications of the 27th International Conference on Logic Programming*, pages 284–288. Schloss Dagstuhl : Leibniz-Center for Informatics, 2011.
- Alexander Šimko. Detection of irrelevant preferences in logic programs. In *Proceedings of 6th Workshop on Intelligent and Knowledge Oriented Technologies*, pages 51–56. Košice: EQUILIBRIA, 2011.
- Alexander Šimko. Preferred answer sets – banned generating set approach. In *Študentská vedecká konferencia FMFI UK, Bratislava 2011 : Zborník príspevkov*, pages 326–333. Bratislava : Fakulta matematiky, fyziky a informatiky UK, 2011.
- Ján Šefránek, Alexander Šimko. Warranted derivations of preferred answer sets. In *Proceedings of 19th International Conference on Applications of Declarative Programming and Knowledge Management and 25th Workshop on Logic Programming*, pages 195–207. Vienna : University of Technology, 2011.
- Alexander Šimko. Preferred answer sets: comparison of generating sets. In *Proceedings of 14th International Workshop on Non-Monotonic Reasoning*, pages 1–9. Vienna : University of Technology, 2012.
- Alexander Šimko. Extension of Gelfond-Lifschitz reduction for preferred answer sets: Preliminary report. In *Kiel Declarative Programming Days 2013*, pages 2–16. Kiel : Institut für Informatik, 2013.
- Ján Šefránek, Alexander Šimko. A Descriptive Approach to Preferred Answer Sets. In *Applications of Declarative Programming and Knowledge Management - 19th International Conference, INAP 2011, and 25th Workshop on Logic Programming, WLP 2011, Vienna, Austria, September 28-30, 2011, Revised Selected Papers*, pages 195–214. Vol. 7773 of Lecture Notes in Computer Science. Springer, 2013.
- Alexander Šimko. A Descriptive Handling of Directly Conflicting Rules in Preferred Answer Sets. To appear in *Declarative Programming and Knowledge Management*. Lecture Notes in Computer Science. Springer. Accepted on 09.02.2014.
- Alexander Šimko. A Family of Descriptive Approaches To Preferred Answer Sets. Accepted for *15th International Workshop on Non-Monotonic Reasoning*. 2014.

## 5 Ocenenia

- “Best Newcomer Award”, ocenenie za najlepší študentský článok získaný za článok “Extension of Gelfond-Lifschitz Reduction for Preferred Answer Sets : Preliminary

Report” získaný na podujatí “22th International Workshop on Functional and (Constraint) Logic programming, 27th Workshop on Logic Programming”.

## 6 Ohlasy

- Alfredo Gabaldon. A Selective Semantics for Logic Programs with Preferences. In *Logics in Artificial Intelligence - 13th European Conference, JELIA 2012, Toulouse, France, September 26-28, 2012. Proceedings*, pages 215–227, 2012.

## 7 Zoznam použitej literatúry

- [1] Krzysztof R. Apt, Howard A. Blair, and Adrian Walker. Towards a Theory of Declarative Knowledge. In *Foundations of deductive databases and logic programming*, pages 89–148. Morgan Kaufmann Publishers Inc., 1988.
- [2] Vernon Asuncion and Yan Zhang. Translating Preferred Answer Set Programs to Propositional Logic. In *Logic Programming and Nonmonotonic Reasoning, 10th International Conference, LPNMR 2009*, pages 396–401, 2009.
- [3] Marcello Balduccini and Michael Gelfond. Model-Based Reasoning for Complex Flight Systems. In *Infotech@Aerospace*, 2005.
- [4] Chitta Baral. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, New York, NY, USA, 2003.
- [5] Gerhard Brewka. Well-Founded Semantics for Extended Logic Programs with Dynamic Preferences. *Journal of Artificial Intelligence Research*, 4:19–36, 1996.
- [6] Gerhard Brewka and Thomas Eiter. Preferred Answer Sets for Extended Logic Programs. *Artificial Intelligence*, 109(1-2):297–356, 1999.
- [7] Francesco Calimeri, Wolfgang Faber, Martin Gebser, Giovambattista Ianni, Roland Kaminski, Thomas Krennwallner, Nicola Leone, Francesco Ricca, and Torsten Schaub. ASP-Core-2, March 2013. URL <https://www.mat.unical.it/aspcomp2013/files/ASP-CORE-2.01c.pdf>.
- [8] Weidong Chen and David Scott Warren. Computation of Stable Models and Its Integration with Logical Query Processing. *IEEE Transactions on Knowledge and Data Engineering*, 8(5):742–757, 1996.
- [9] cmodels. URL <http://www.cs.utexas.edu/users/tag/cmodels.html>.
- [10] csmodels. URL <http://www.ceng.metu.edu.tr/~orkunt/csmodels/>.

- [11] James P Delgrande, Torsten Schaub, and Hans Tompits. A Framework for Compiling Preferences in Logic Programs. *Theoretical Computer Science*, 3(2):129–187, 2003.
- [12] Yannis Dimopoulos, Bernhard Nebel, and Jana Koehler. Encoding PLanning Problems in Nonmonotonic Logic Programs. In *Proceedings of the Fourth European Conference on Planning*, pages 169–181, 1997.
- [13] DLV. URL <http://www.dlvsystem.com/dlvsystem/>.
- [14] Phan Minh Dung. On the relations between stable and well-founded semantics of logic programs. *Theoretical Computer Science*, 105(1):7–25, 1992.
- [15] Phan Minh Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.
- [16] Thomas Eiter, Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. The Diagnosis Frontend of the dlv System. *AI Communications*, 12(1-2):99–111, 1999.
- [17] Thomas Eiter, Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. Computing Preferred Answer Sets by Meta-Interpretation in Answer Set Programming. Technical report, Institut für Informationssysteme, Technische Universität Wien, 2002.
- [18] Thomas Eiter, Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. Computing Preferred Answer Sets by Meta-Interpretation in Answer Set Programming. *Theoretical Computer Science*, 3(4-5):463–498, 2003.
- [19] Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. dlhex: A System for Integrating Multiple Semantics in an Answer-Set Programming Framework. In *Proceedings of the 20th Workshop on Logic Programming and Constraint Systems (WLP 06)*, pages 206–210, 2006.
- [20] Wolfgang Faber and Kathrin Konczak. Strong Equivalence for Logic Programs with Preferences. In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 430–435, 2005.
- [21] Wolfgang Faber, Nicola Leone, and Simona Perri. The Intelligent Grounder of DLV. In *Correct Reasoning - Essays on Logic-Based AI in Honour of Vladimir Lifschitz*, pages 247–264, 2012.
- [22] Alfredo Gabaldon. A Selective Semantics for Logic Programs with Preferences. In *Logics in Artificial Intelligence - 13th European Conference, JELIA 2012, Toulouse, France, September 26-28, 2012. Proceedings*, pages 215–227, 2012.

- [23] Martin Gebser, Benjamin Kaufmann, André Neumann, and Torsten Schaub. Conflict-Driven Answer Set Solving. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 386–392, 2007.
- [24] Martin Gebser, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Sven Thiele. On the Input Language of ASP Grounder Gringo. In *Proceedings of the 10th International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 502–508, 2009.
- [25] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub, and Sven Thiele. *A User’s Guide to gringo, clasp, clingo, and iclingo*, October 2010. URL [http://sourceforge.net/projects/potassco/files/potassco\\_guide/2010-10-04/guide.pdf/download](http://sourceforge.net/projects/potassco/files/potassco_guide/2010-10-04/guide.pdf/download).
- [26] Martin Gebser, Roland Kaminski, and Torsten Schaub. Complex Optimization in Answer Set Programming. *Theoretical Computer Science*, 11(4-5):821–839, 2011.
- [27] Michael Gelfond and Marcello Balduccini. Diagnostic reasoning with A-Prolog. *Theoretical Computer Science*, 3(4-5):425–461, 2003.
- [28] Michael Gelfond and Yulia Kahl. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents*. 2014. URL <http://redwood.cs.ttu.edu/~mgelfond/FALL-2012/book.pdf>.
- [29] Michael Gelfond and Vladimir Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 9(3-4):365–386, 1991.
- [30] Michael Gelfond and Tran Cao Son. Reasoning with Prioritized Defaults. In *Third International Workshop on Logic Programming and Knowledge Representation*, pages 164–223, 1997.
- [31] Susanne Grell, Kathrin Konczak, and Torsten Schaub. NoMoRe<, 2014. URL <http://www.cs.uni-potsdam.de/wv/nomorepref/>.
- [32] Giovambattista Ianni. Third (Open) Answer Set Programming Competition - 2011, 2011. URL <https://www.mat.unical.it/aspcomp2011/FrontPage>.
- [33] Katsumi Inoue and Chiaki Sakama. Negation as failure in the head. *The Journal of Logic Programming*, 35(1):39–78, 1998.
- [34] Kathrin Konczak, Thomas Linke, and Torsten Schaub. Graphs and colorings for answer set programming. *Theoretical Computer Science*, 6(1-2):61–106, 2006.
- [35] Kenneth Kunen. Signed Data Dependencies In Logic Programs. *The Journal of Logic Programming*, 7(3):231–245, 1989.



- [36] Vladimir Lifschitz. Answer Set Programming and Plan Generation. *Artificial Intelligence*, 138(1-2):39–54, 2002.
- [37] Vladimir Lifschitz. What Is Answer Set Programming? In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1594–1597, 2008.
- [38] Vladimir Lifschitz and Hudson Turner. Splitting a logic program. In *Proceedings of the Eleventh International Conference on Logic Programming*, pages 23–37, 1994.
- [39] Vladimir Lifschitz, Lappoon R Tang, and Hudson Turner. Nested Expressions in Logic Programs. *Annals Of Mathematics and Artificial Intelligence*, 25(3-4):369–389, 1999.
- [40] Vladimir Lifschitz, David Pearce, and Agustin Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2(4):526–541, 2001.
- [41] Fangzhen Lin and Yuting Zhao. ASSAT: Computing Answer Sets of A Logic Program By SAT Solvers. *Artificial Intelligence*, 1-2:115–137, 2002.
- [42] Ilkka Niemela and Patrik Simons. Smodels - an implementation of the stable model and well-founded semantics for normal logic programs. In *Logic Programming and Nonmonotonic Reasoning, 4th International Conference*, pages 421–430. Springer-Verlang, 1997.
- [43] Monica Nogueira, Marcello Balduccini, Michael Gelfond, Richard Watson, and Matthew Barry. An A-Prolog decision support system for the Space Shuttle. In *Proceedings of the Third International Symposium on Practical Aspects of Declarative Languages*, pages 169–183. Springer, 2001. ISBN 3540417680. URL [http://link.springer.com/chapter/10.1007/3-540-45241-9\\_12](http://link.springer.com/chapter/10.1007/3-540-45241-9_12).
- [44] nomore++. URL <http://www.cs.uni-potsdam.de/wv/nomore/index.html>.
- [45] Gerald Pfeifer and Wolfgang Faber. Computing Preferred and Weakly Preferred Answer Sets by Meta-Interpretation in Answer Set Programming, 2005. URL <http://www.dbai.tuwien.ac.at/proj/dlv/preferred/>.
- [46] plp. URL <http://www.cs.uni-potsdam.de/~torsten/plp/>.
- [47] Potassco - the Potsdam Answer Set Solving Collection. URL <http://potassco.sourceforge.net/>.
- [48] Teodor C. Przymusiński. Perfect Model Semantics. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium*, pages 1081–1096, 1988.

- [49] Teodor C. Przymusiński. On the declarative semantics of deductive databases and logic programs. In *Foundations of deductive databases and logic programming*, pages 193–216. Morgan Kaufmann Publishers Inc., 1988. ISBN 0-934613-40-0.
- [50] Teodor C. Przymusiński. On the declarative and procedural semantics of logic programs. *Journal of Automated Reasoning*, 5(2):167–205, 1989.
- [51] Chiaki Sakama and Katsumi Inoue. Prioritized logic programming and its application to commonsense reasoning. *Artificial Intelligence*, 123(1-2):185–222, 2000.
- [52] Taisuke Sato. On Consistency of First-Order Logic Programs. Technical Report TR 87-12, Electrotechnical Lab, 1987.
- [53] Torsten Schaub. Collection on Answer Set Programming (ASP) and more, 2011. URL <http://www.cs.uni-potsdam.de/~torsten/asp/>.
- [54] Torsten Schaub and Kewen Wang. A comparative study of logic programs with preference. In *Proceedings of the 17th international joint conference on Artificial intelligence*, pages 507–602, 2001.
- [55] Torsten Schaub and Kewen Wang. Preferred well-founded semantics for logic programming by alternating fixpoints: Preliminary Report. In *9th International Workshop on Non-Monotonic Reasoning*, pages 238–246, 2002.
- [56] Torsten Schaub and Kewen Wang. A semantic framework for preference handling in answer set programming. *Theoretical Computer Science*, 3(4-5):569–607, 2003.
- [57] Ján Šeřfránek. Preferred answer sets supported by arguments. In *Proceedings of Twelfth International Workshop on Non-Monotonic Reasoning*, pages 232–240, 2008.
- [58] Ján Šeřfránek and Alexander Šimko. A Descriptive Approach to Preferred Answer Sets. In *Applications of Declarative Programming and Knowledge Management*, pages 195–214. Springer Berlin Heidelberg, 2013.
- [59] Alexander Šimko. PASS, 2014. URL <http://dai.fmph.uniba.sk/~simko/lpp/>.
- [60] Patrik Simons. *Extending and Implementing the Stable Model Semantics*. PhD thesis, Helsinki University of Technology, 2000.
- [61] Patrik Simons, Ilkka Niemela, and Timo Sooinen. Extending and Implementing the Stable Model Semantics. *Artificial Intelligence*, 138(1-2):181–234, 2002.
- [62] smodels. URL <http://www.tcs.hut.fi/Software/smodels/>.
- [63] Timo Sooinen and Ilkka Niemela. Developing a Declarative Rule Language for Applications in Product Configuration. In *PADL '99 Proceedings of the First International Workshop on Practical Aspects of Declarative Languages*, 1999.

- [64] V S Subrahmanian, Dana Nau, and Carlo Vago. WFS + Branch and Bound = Stable Models. *IEEE Transactions on Knowledge and Data Engineering*, 7(3):362–377, 1995.
- [65] Syrjanen Tommi. *Lparse 1.0 User’s Manual*.
- [66] Allen Van Gelder, Kenneth A Ross, and John S Schlipf. The Well-founded Semantics for General Logic Programs. *Journal of the ACM*, 38(3):620–650, 1991.
- [67] Toshiko Wakaki. Preference-Based Argumentation Handling Dynamic Preferences Built on Prioritized Logic Programming. In *Agents in Principle, Agents in Practice*, pages 336–348. Springer Berlin Heidelberg, 2011.
- [68] Kewen Wang, Lizhu Zhou, and Fangzhen Lin. Alternating Fixpoint Theory for Logic Programs with Priority. In *Proceedings of the First International Conference on Computational Logic*, pages 164–178, 2000.
- [69] Yan Zhang. Two results for prioritized logic programming. *Theoretical Computer Science*, 3(2):223–242, 2003.
- [70] Yan Zhang and Norman Y. Foo. Answer Sets for Prioritized Logic Programs. In *Proceedings of the 1998 International Logic Programming Symposium*, pages 69–83, 1997.
- [71] Yan Zhang and Norman Y. Foo. Towards Generalized Rule-based Updates. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97*, pages 82–88, 1997.

## 8 Summary

Logic programming under the answer set semantics is a declarative programming and knowledge representation language. A logic program is a collection of rules. In case of conflicting rules, preferences on rules enable us to select which of the rules to use. It has been studied how logic programming can be extended with preferences on rules, and quite many approaches have been proposed. In addition, principles, key properties that can be used to evaluate healthiness of the semantics, have been studied. However, the term “preferences” is abstract. Many approaches for preferences on rules exist. Comprehensive review of their properties is lacking. All this makes selecting an appropriate semantics for a task at hand difficult. In this work, we elaborate principles for logic programming with preferences on rules. We summarize existing principles, propose new principles extracted from existing approaches or based on our intuitions, and investigate connections between the principles. We conduct a comprehensive evaluation of existing semantics on principles. We thus provide basis for selecting an appropriate semantics for a task at hand. For

viewing preferences as a mechanism for expressing exceptions between rules, we propose requirements on semantics. We connect the requirements with satisfaction and violation of principles, which allows us to formally evaluate whether a semantics satisfies the requirements. It turns out that no semantics for logic programming with preferences on rules fully meets the specified requirements. For viewing preferences as a mechanism for expressing exceptions between rules we develop a family of fully declarative semantics for logic programming with preferences on rules. The first semantics is restricted to the case of direct conflicts. It can be efficiently computed, and is able to ignore preferences between non-conflicting rules. The second semantics meets all the specified requirements, works for general conflicts, and is also able to ignore preferences between non-conflicting rules. So far, it is not known whether an efficient implementation of the semantics is possible. We have investigated relaxations of the second semantics, which can be effectually computed. The third semantics works for general conflicts, but drops the condition for ignoring non-conflicting preferences. It is a programmer's responsibility to specify preferences only between conflicting rules. The fourth one is designed for a subclass of conflicts, and ignores preferences between non-conflicting rules. We have investigated relations among our semantics, and their relations to existing semantics. It turns out that the semantics form two branches of a hierarchy. We also provide a simple tool called PASS that implements solvers for all our semantics via meta-interpretation technique.