1. Úvod

Táto práca stručne popisuje Defeasible Logic Programming, kombináciu klasického logického programovania a defeasible argumentácie. Tento prístup umožňuje postaviť program, v ktorom sú zahrnuté aj nekompletné a sporné informácie. Využíva dva druhy pravidiel, silné a slabé, z ktorých vytvára argumenty podporujúce alebo vyvracajúce dotazy, na ktoré je potom schopný odpovedať.

2. Jazyk

Jazyk Defeasible Logic Programming je definovaný tromi množinami, množinou *faktov*, a *pravidlami*, ktoré sa delia na *silné* (strong, strict) a *slabé* (defeasible). *Literály* budú v tomto jazyku tvorené atómami "A" alebo ich negáciami "∼A".

Definícia 2.1 (*Fakt*)**.** Faktom je každý literál, t.j. atóm alebo negácia atómu.

Definícia 2.2 (*Silné pravidlo*)**.** Silné pravidlo je usporiadaná dvojica, označovaná ako "$Head \leftarrow Body$" kde $Head$ je literál a $Body$ konečná neprázdna množina literálov.

Definícia 2.3 (*Slabé pravidlo*)**.** Slabé pravidlo je usporiadaná dvojica, označovaná ako "$Head \prec Body$" kde $Head$ je literál a $Body$ konečná neprázdna množina literálov.

Pri oboch typoch pravidiel, kde hlava je literál "L_0" a telo množina $\{L_1, \ldots, L_n\}$, $(n > 0)$ sa používa zápis $L_0 \leftarrow L_1, \ldots, L_n$, resp. $L_0 \prec L_1, \ldots, L_n$

Rozdiel medzi týmito pravidlami je zo syntaktického hľadiska iba v použitom symbole. V praxi však silné pravidlá budú vyjadrovať nepopierateľnú závislosť a slabé takú závislosť, ktorá sa dá vyvrátiť, prípadne neplatí vždy. Príkladom použitia silného pravidla by bola veta "*všetky kovy sú prvkami*" vyjadrená ako $Prvok(X) \leftarrow Kov(X)$. Veta "*Kov je zvyčajne pevného skupenstva*" by sa zapísala ako $Pevny(X) \prec Kov(X)$.

Definícia 2.4 (*Defeasible Logic Program*)**.** Defeasible logic program (de.l.p.) $\mathcal{P}$, voľne preložené ako zrušiteľný, prípadne napadnuteľný logický program, je množina nie nutne konečnej množiny faktov, silných a slabých pravidiel. V programe $\mathcal{P}$ budeme rozlišovať podmnožinu faktov a silných pravidiel, označovanú Π a podmnožinu slabých pravidiel Δ. Program $\mathcal{P}$ potom bude dvojica (Π, Δ)

Definícia 2.5 (*Napadnuteľné odvodenie*)**.** Nech $\mathcal{P} = (\Pi, \Delta)$ je de.l.p a L literál. Napadnuteľné odvodenie L z $\mathcal{P}$, označované ako $\mathcal{P} \mid\!\sim L$, je konečnou postupnosťou literálov $L_1, \ldots, L_n$, pričom platí:

(1) $L_n = L$
(2) L_i je faktom v Π, alebo existuje pravidlo $R \in \mathcal{P}$ s hlavou L_i a telom $B_1, \ldots B_k$, pričom každý z literálov $B_1 \ldots B_k$ je niektorým z predchádzajúcich členov tejto postupnosti, t.j. $B_m = L_j$, $(j < i)$,

Toto odvodenie nazývame napadnuteľným z toho dôvodu, že v danom programe $\mathcal{P}$ môže existovať informácia, ktorá je v spore s L, čím zabráni akceptovaniu L ako platného faktu.

Definícia 2.6 (*Striktné odvodenie*)**.** Nech $\mathcal{P}$ je de.l.p a L literál s napadnuteľným odvodením $L_1, \ldots, L_n$. Potom L sa dá striktne odvodiť z $\mathcal{P}$ práve vtedy, keď je L buď fakt, alebo všetky pravidlá použité na odvodenie postupnosti $L_1, \ldots, L_n$ boli silné.

Definícia 2.7 (*Sporná množina pravidiel*)**.** Množina pravidiel je *sporná* práve vtedy, keď existuje napadnuteľné odvodenie dvojice komplementárnych literálov, t.j. literálov L a $\sim L$ používajúce pravidlá z tejto množiny.

Podľa konvencie množina Π striktných pravidiel a faktov nikdy *nebude obsahovať spor*. Naopak je užitočné, ak celý program $\mathcal{P}$ sporný je, pretože následne sa tento spor dá riešiť ďalej zavedeným formalizmom.

3. Napadnuteľná argumentácia

V napadnuteľných logických programoch sa odpovede na dotazy budú podporovať argumentmi. Aj keď celý logický program byť sporný môže, tieto argumenty budú definované ako nesporné a ich účelom bude buď odpoveď na dotaz podporiť, alebo vyvrátiť iný argument, ktorý ju podporuje.

Definícia 3.1 (*Argumentačná štruktúra*)**.** Nech h je literál a $\mathcal{P} = (\Pi, \Delta)$ de.l.p.. Hovoríme, že $\langle \mathcal{A}, h \rangle$ je argumentačnou štruktúrou(alebo argumentom) pre h, ak $\mathcal{A} \subset \Delta$ a platí:

(1) pre h existuje napadnuteľné odvodenie z množiny pravidiel $\Pi \cup \mathcal{A}$
(2) množina $\Pi \cup \mathcal{A}$ nie je sporná
(3) $\mathcal{A}$ je minimálna, t.j. neexistuje taká vlastná podmnožina $\mathcal{A}$, ktorá by spĺňala body 1. a 2.

Argumentačná štruktúra $\langle \mathcal{A}, h \rangle$, alebo skrátene argument $\mathcal{A}$ pre h je teda minimálna množina napadnuteľných pravidiel neobsahujúca spor, získaná z napadnuteľného odvodenia pre literál h.

Definícia 3.2 (*Podargument*)**.** Argumentačnú štruktúru $\langle \mathcal{B}, h \rangle$ nazveme podargumentom $\langle \mathcal{A}, h \rangle$, ak $\mathcal{B} \subset \mathcal{A}$

Definícia 3.3 (*Nesúhlas*)**.** Nech $\mathcal{P} = (\Pi, \Delta)$ je de.l.p.. Hovoríme, že literály h a h_1 nesúhlasia práve vtedy, keď je množina $\Pi \cup \{h, h_1\}$ sporná.

Definícia 3.4 (*Protiargument*)**.** Hovoríme, že argument $\langle \mathcal{A}_1, h_1 \rangle$ je protiargumentom $\langle \mathcal{A}_2, h_2 \rangle$ vzhľadom na literál h práve vtedy, keď existuje podargument $\langle \mathcal{A}, h \rangle$ argumentu $\langle \mathcal{A}_2, h_2 \rangle$ taký, že h a h_1 nesúhlasia.

Na porovnanie jednotlivých argumentov budeme potrebovať zaviesť reláciu preferencie medzi napadnuteľnými pravidlami, ktoré jediné môžu spôsobiť v programe spor. Táto preferencia medzi jednotlivými pravidlami môže byť ľubovoľného typu a nebudeme ju zavádzať nijako špecificky. Iba predpokladáme, že existuje a budeme ju označovať $>$.

Definícia 3.5 (*Preferencia argumentov*)**.** Nech $\mathcal{P}$ je de.l.p. a $>$ relácia definovaná medzi slabými pravidlami. Argument $\langle \mathcal{A}_1, h_1 \rangle$ bude preferovaný nad argumentom $\langle \mathcal{A}_2, h_2 \rangle$, označované $\langle \mathcal{A}_1, h_1 \rangle \succ \langle \mathcal{A}_2, h_2 \rangle$ ak:

(1) existuje aspoň jedno pravidlo $r_1 \in \mathcal{A}_1$ a aspoň jedno pravidlo $r_2 \in \mathcal{A}_2$ také, že $r_1 > r_2$
(2) a neexistuje žiadne pravidlo $r_2' \in \mathcal{A}_2$ a $r_1' \in \mathcal{A}_1$ také, že $r_2' > r_1'$

4. Argumentačné línie

Keďže argumenty sa môžu vzájomne napádat, budú vytvárať postupnosti, v ktorých každý člen bude napádať predchádzajúci argument. Takéto postupnosti sa budú nazývať argumentačné línie.

Definícia 4.1 (*Proper defeater*)**.** Nech $\langle \mathcal{A}_1, h_1 \rangle$ a $\langle \mathcal{A}_2, h_2 \rangle$ sú dva argumenty. Argument $\langle \mathcal{A}_1, h_1 \rangle$ je správny (proper) defeater pre $\langle \mathcal{A}_2, h_2 \rangle$ vzhľadom na literál h práve vtedy, keď existuje podargument $\langle \mathcal{A}, h \rangle$ argumentu $\langle \mathcal{A}_2, h_2 \rangle$ taký, že $\langle \mathcal{A}_1, h_1 \rangle$ je protiargumentom $\langle \mathcal{A}_2, h_2 \rangle$ vzhľadom na h a platí $\langle \mathcal{A}_1, h_1 \rangle \succ \langle \mathcal{A}, h \rangle$.

Definícia 4.2 (*Blokujúci defeater*)**.** Nech $\langle \mathcal{A}_1, h_1 \rangle$ a $\langle \mathcal{A}_2, h_2 \rangle$ sú dva argumenty. Argument $\langle \mathcal{A}_1, h_1 \rangle$ je blokujúci defeater pre $\langle \mathcal{A}_2, h_2 \rangle$ vzhľadom na literál h práve vtedy, keď existuje podargument $\langle \mathcal{A}, h \rangle$ argumentu $\langle \mathcal{A}_2, h_2 \rangle$ taký, že $\langle \mathcal{A}_1, h_1 \rangle$ je protiargumentom $\langle \mathcal{A}_2, h_2 \rangle$ vzhľadom na h a $\langle \mathcal{A}_1, h_1 \rangle$ nie je v žiadnom preferenčnom vzťahu s $\langle \mathcal{A}, h \rangle$, t.j. platí $\langle \mathcal{A}_1, h_1 \rangle \not\succ \langle \mathcal{A}, h \rangle$ a $\langle \mathcal{A}, h \rangle \not\succ \langle \mathcal{A}_1, h_1 \rangle$.

Definícia 4.3 (*Defeater*)**.** Nech $\langle \mathcal{A}_1, h_1 \rangle$ a $\langle \mathcal{A}_2, h_2 \rangle$ sú dva argumenty. Argument $\langle \mathcal{A}_1, h_1 \rangle$ je defeater pre $\langle \mathcal{A}_2, h_2 \rangle$ práve vtedy, keď je $\langle \mathcal{A}_1, h_1 \rangle$ správnym, alebo blokujúcim defeaterom argumentu $\langle \mathcal{A}_2, h_2 \rangle$.

Definícia 4.4 (*Argumentačná línia*)**.** Nech $\mathcal{P}$ je de.l.p. a $\langle \mathcal{A}_0, h_0 \rangle$ je argumentom v $\mathcal{P}$. Argumentačná línia je postupnosť argumentov v $\mathcal{P}$, označovaná ako $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \ldots]$, kde každý člen $\langle \mathcal{A}_i, h_i \rangle, i > 0$ tejto postupnosti je defeaterom jeho predchodcu $\langle \mathcal{A}_{i-1}, h_{i-1} \rangle$.

Táto definícia umožňuje vytvorenie nekonečných argumentačných línií, ktoré však z praktického hľadiska nie sú vhodné. Preto sa ďalej zavedie niekoľko obmedzení, ktoré takýmto nekonečným postupnostiam zabránia vzniknúť.

Definícia 4.5 (*Podporujúce a odporujúce argumenty*)**.** Nech
$\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \ldots]$ je argumentačná línia. Definujeme množinu podporujúcich (supporting) argumentov ako
$\Lambda_S = \{\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \langle \mathcal{A}_4, h_4 \rangle, \ldots\}$
a množinu odporujúcich (interfering) argumentov
$\Lambda_I = \{\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \langle \mathcal{A}_4, h_4 \rangle, \ldots\}$.

Definícia 4.6 (*Súhlasné argumenty(Concordance)*)**.** Nech $\mathcal{P} = (\Pi, \Delta)$ je de.l.p.. Dva argumenty $\langle \mathcal{A}_1, h_1 \rangle$ a $\langle \mathcal{A}_2, h_2 \rangle$ sú súhlasné (concordant) práve vtedy, keď množina $\Pi \cup \mathcal{A}_1 \cup \mathcal{A}_2$ neobsahuje spor. Všeobecnejšie, množina argumentov $\{\langle \mathcal{A}_i, h_i \rangle\}_{i=1}^{n}$ je súhlasná práve vtedy, keď $\Pi \cup \bigcup_{i=1}^{n} \mathcal{A}_i$ neobsahuje spor.

Definícia 4.7 (*Akceptovateľná argumentačná línia*)**.** Nech
$\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \ldots, \langle \mathcal{A}_i, h_i \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle]$ je argumentačná línia. Akceptovateľnou ju nazveme práve vtedy, keď pre ňu platí:

(1) Λ je konečná postupnosť.
(2) Množina podporujúcich argumentov Λ_S je súhlasná a aj množina odporujúcich argumentov Λ_I je súhlasná.
(3) Žiaden argument $\langle \mathcal{A}_k, h_k \rangle \in \Lambda$ nie je podargumentom argumentu $\langle \mathcal{A}_i, h_i \rangle$ vyskytujúceho sa v Λ skôr ($i < k$).
(4) Pre všetky i platí, že ak je $\langle \mathcal{A}_i, h_i \rangle$ blokujúcim defeaterom pre argument $\langle \mathcal{A}_{i-1}, h_{i-1} \rangle$ a $\langle \mathcal{A}_{i+1}, h_{i+1} \rangle$ existuje, potom $\langle \mathcal{A}_{i+1}, h_{i+1} \rangle$ je správnym defeaterom pre $\langle \mathcal{A}_i, h_i \rangle$, t.j. v argumentačnej línii neexistujú za sebou dva blokujúce defeatery.

Dôvody pre tieto obmedzenia na akceptovateľnosť argumentačných línií sú viaceré a existuje niekoľko možností, ako tieto obmedzenia upraviť.

5. Dialektická analýza

Definícia 5.1 (*Dialektický strom*)**.** Nech $\langle \mathcal{A}_0, h_0 \rangle$ je argumentom v programe $\mathcal{P}$. Dialektický strom pre $\langle \mathcal{A}_0, h_0 \rangle$, označený ako $\mathcal{T}_{\langle \mathcal{A}_0, h_0 \rangle}$ je definovaný nasledovne:

(1) Koreň tohto stromu je označený $\langle \mathcal{A}_0, h_0 \rangle$.
(2) Nech N je vrchol stromu označený ako $\langle \mathcal{A}_n, h_n \rangle$ a $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle]$ postupnosť označení vrcholov od koreňa k listu N a nech $\langle \mathcal{B}_1, q_1 \rangle, \langle \mathcal{B}_2, q_2 \rangle, \ldots, \langle \mathcal{B}_k, q_k \rangle$ sú všetky defeatery pre $\langle \mathcal{A}_n, h_n \rangle$.

Ak platí, že argumentačná línia $\Lambda' = [\langle \mathcal{A}_0, h_0\rangle, \langle \mathcal{A}_1, h_1\rangle, \ldots, \langle \mathcal{A}_n, h_n\rangle, \langle \mathcal{B}_i, q_i\rangle]$ pre ľubovoľné $i = 1 \ldots k$ je akceptovateľná, potom N bude mať syna N_i označeného ako $\langle \mathcal{B}_i, q_i\rangle$.

Ak pre $\langle \mathcal{A}_n, h_n\rangle$ žiaden defeater $\langle \mathcal{B}_i, q_i\rangle$ taký, že Λ' je akceptovateľná neexistuje, potom N je list.

Dialtektický strom pre argument $\langle \mathcal{A}, h\rangle$ je teda skupina argumentačných línií, ktoré tento argument podporujú, resp. vyvracajú. Listy tohoto stromu sú argumenty, pre ktoré už neexistuje akceptovateľný defeater a teda sa budú označovať ako platné. Následne sa dá zistiť platnosť argumentov nad nimi a takto postupne označiť celý strom až po koreň.

Definícia 5.2 (*Vyhodnocovanie dialektického stromu*)**.** Nech $\mathcal{T}_{\langle \mathcal{A},h\rangle}$ je dialektický strom pre $\langle \mathcal{A}, h\rangle$. K nemu prislúchajúci označený dialektický strom $\mathcal{T}^\star_{\langle \mathcal{A},h\rangle}$ sa získa označením každého vrcholu v $\mathcal{T}_{\langle \mathcal{A},h\rangle}$ nasledovne:

(1) Všetky listy v strome $\mathcal{T}_{\langle \mathcal{A},h\rangle}$ sa v strome $\mathcal{T}^\star_{\langle \mathcal{A},h\rangle}$ označia ako "U".
(2) Nech $\langle \mathcal{B}, q\rangle$ je vnútorný vrchol stromu $\mathcal{T}_{\langle \mathcal{A},h\rangle}$. Potom $\langle \mathcal{B}, q\rangle$ bude označený ako "U" v $\mathcal{T}^\star_{\langle \mathcal{A},h\rangle}$ práve vtedy, keď každý syn vrcholu $\langle \mathcal{B}, q\rangle$ je označený ako "D". Vrchol $\langle \mathcal{B}, q\rangle$ bude označený ako "D" v $\mathcal{T}^\star_{\langle \mathcal{A},h\rangle}$, ak aspoň jeden jeho syn je označený ako "U".

Takto označený strom určuje platnosť argumentu v jeho koreni, ktorý je argumentom pre literál h a teda je možné rozhodnúť, či je tento literál v programe platný, alebo nie.

Definícia 5.3 (*Zaručené literály*)**.** Nech $\langle \mathcal{A}, h\rangle$ je argumentom v programe $\mathcal{P}$ a $\mathcal{T}^\star_{\langle \mathcal{A},h\rangle}$ je k nemu prislúchajúci označený dialektický strom. Potom literál h je zaručený práve vtedy, keď koreň tohto stromu je označený ako "U". Hovoríme, že $\mathcal{A}$ je zárukou pre h.

Definícia 5.4 (*Odpovede na dotazy*)**.** Pre dotaz h exisujú štyri možné odpovede programu $\mathcal{P}$.

(1) Yes, ak h je v $\mathcal{P}$ zaručený literál.
(2) No, ak $\sim h$ je zaručený literál.
(3) Undecided, ak ani h, ani $\sim h$ nie sú zaručené literály.
(4) Unknown, ak h nie je v jazyku $\mathcal{P}$.

Tento záver nám umožňuje pri ľubovoľnom programe $\mathcal{P}$ rozhodnúť, či v ňom literál zaručený je, alebo nie je. Samozrejme môže nastať prípad, kedy sa o tomto literáli rozhodnúť nedá, alebo sa k nemu program nevyjadruje vôbec (nepozná ho).