Foundation of Paraconsistent Rule-Based Reasoning with Graded Truth Values

DE ANGELIS, Francesco, DI MARZO SERUGENDO, Giovanna, SZALAS, Andrzej

Abstract

This technical report includes the proofs of several lemmas and theorems used in the paper "Paraconsistent Rule-Based Reasoning with Graded Truth Degrees", written by the same authors.

Reference

DE ANGELIS, Francesco, DI MARZO SERUGENDO, Giovanna, SZALAS, Andrzej. Foundation of Paraconsistent Rule-Based Reasoning with Graded Truth Values. 2017, Complementary text to a paper

Available at:
http://archive-ouverte.unige.ch/unige:94264

Disclaimer: layout of this document may differ from the published version.
Foundation of Paraconsistent Rule-Based Reasoning with Graded Truth Values

Francesco Luca De Angelis\textsuperscript{a,}\textsuperscript{*}, Giovanna Di Marzo Serugendo\textsuperscript{a}, Andrzej Szalas\textsuperscript{b,c}

\textsuperscript{a}Institute of Services Science, University of Geneva, CH-1227 Carouge, SWITZERLAND
\textsuperscript{b}Institute of Informatics, University of Warsaw, PL-02-097 Warsaw, POLAND
\textsuperscript{c}Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, SWEDEN

Abstract

This technical report includes the proofs of several lemmas and theorems used in “Paraconsistent Rule-Based Reasoning with Graded Truth Degrees”, written by the same authors [1].

Keywords: graded truth-values, knowledge bases, paraconsistent reasoning, rule-based languages

1. Interpretations

Lemma 1.1: (Section 4 in [1]) Let \( T \) be a set, \( \leq \) a partial order such that \( \langle T, \leq \rangle \) is a complete lattice, \( S \) is a generic set and \( \mathcal{I} \) is the set of all functions \( S \rightarrow T \). We consider the extension \( \leq \) over \( \mathcal{I} \) such that, for every couple \( I_1, I_2 \in \mathcal{I} \): \( I_1 \leq I_2 \) iff \( I_1(s) \leq I_2(s) \) for all \( s \in S \). Then \( \langle \mathcal{I}, \leq \rangle \) is a complete lattice.

Proof We have to prove that every subset \( V \) of \( \mathcal{I} \) admits an infimum \( \inf(V) \) and a supremum \( \sup(V) \). We prove the existence of \( \sup(V) \), the case for \( \inf(V) \) is similar. We have to show that there exists \( \sup(V) \in \mathcal{I} \) such that:

*Corresponding author

Email addresses: francesco.deangelis@unige.ch (Francesco Luca De Angelis),
giovanna.dimarzo@unige.ch (Giovanna Di Marzo Serugendo),
andrzej.szalas@mimuw.edu.pl, liu.se (Andrzej Szalas)

Preprint submitted to Journal of applied logic August 31, 2017
• \( \forall I \in V \ I \leq \sup(V) \);
• if \( (\forall I \in V \ I \leq J) \) then \( \sup(V) \leq J \).

Let \( V \) be a subset of \( I \); for every \( s \in S \) we define:

\[
V(s) \overset{\text{def}}{=} \{ I(s) \mid I \in V \}.
\]

Given that \((\tau, \leq)\) is a complete lattice, \( V(s) \) admits a supremum \( \sup(V(s)) \).

We define \( \sup_V \) as follows: for every \( s \in S \) we define:

\[
\sup_V(V(s)) \overset{\text{def}}{=} \sup(V(s)).
\]

We observe that \( \forall I \in V \ (I(s) \leq \sup(V(s))) \) for \( s \in S \).

Then:

\[
\forall s \in S(\forall I \in V(I(s) \leq \sup(V(s)))) \equiv
\forall I \in V(\forall s \in S(I(s) \leq \sup(V(s)))) \equiv
\forall I \in V(\forall s \in S(I(s) \leq \sup_V(s))) \equiv
\forall I \in V(I \leq \sup_V).
\]

Moreover, according to its definition, given \( s \in S \) and \( J \in I \),

\[
\text{if } \forall I \in V (I(s) \leq J(s)) \text{ then } \sup_V(s) \leq J(s).
\]

Thus:

\[
\text{if } \forall s \in S(\forall I \in I(s) \leq J(s)) \text{ then } \forall s \in S(\sup_V(s) \leq J(s))
\]

i.e., if \( \forall I \in I (I \leq J) \) then \( \sup_V \leq J \).

We therefore conclude that \( \sup_V = \sup(V) \).

\[\Box\]

**Lemma 1.2:** (Lemma 3.1 in [1]) Let \( I_1 \) and \( I_2 \) be two many-valued Herbrand interpretations for a program \( P \) and let \( H \leftarrow B \) a ground rule in \( P' \) with \( B = B_1, ..., B_n \). If \( I_1 \leq_k I_2 \) and \( t_1 \leq_k I_1(B) \) then \( I_1(B) \leq_k I_2(B) \).

**Proof**

(i) If \( I_1(B) = t_i \) for some \( 1 \leq i \leq N \) then there exists \( 1 \leq m \leq n \) such that \( I_1(B_m) = t_i \) and \( I_1(B_z) \geq_m t_i \) for every \( z \neq m \). Then \( I_2(B_m) \geq_k t_i \) and \( I_2(B_z) \geq_k t_i \), entailing \( I_2(B) \geq t_i = I_1(B) \).

(ii) If \( I_1(B) = i_{i,j} \) for some \( 1 \leq i, j \leq N \) there are two cases. We use the following properties of greatest lower bounds: given a partially ordered set \((S, \leq)\) and \( \{a_1, ..., a_n\} \subseteq S \), \( \text{glb}\{a_1, ..., a_n\} \leq \text{glb}\{a_i, a_j\} \)
for all $1 \leq i, j \leq n$; moreover if $a, b, c, d \in S$, $a \leq c$ and $b \leq d$ then \( \text{glb}\{a, b\} \leq \text{glb}\{c, d\} \).

1) There exist $1 \leq m < z \leq n$ such that \( I_1(B_m) = i_{a,b} \) and \( I_1(B_z) = i_{c,d} \) with \( i_{a,b} \wedge m i_{c,d} = i_{i,j} \). Then \( I_2(B_m) \geq_k i_{a,b} \) and \( I_2(B_z) \geq_k i_{c,d} \), with:

\[
I_2(B_m) \wedge_m I_2(B_z) \leq_m i_{a,b} \wedge_m i_{c,d} = i_{i,j}
\]

Concluding, \( I_2(B) \geq_k i_{i,j} \).

2) Otherwise there exists $1 \leq m \leq n$ such that \( I_1(B_m) = i_{i,j} \) and \( I_1(B_z) \geq_m i_{i,j} \) for every \( z \neq m \). Then \( I_2(B_m) \geq_k i_{i,j} \) and \( I_2(B_z) \geq k_t_1 \), concluding \( I_2(B) \geq_k i_{i,j} \).

\[\triangleright\]

**Theorem 1.1:** (Theorem 4.1 in [1]) Let $P$ be a program, $I_1$ and $I_2$ two many-valued Herbrand interpretations. If \( I_1 \leq_k I_2 \) then \( T_P(I_1) \leq_k T_P(I_2) \).

**Proof** For each rule \( H \leftarrow B \in P' \) we have \( I_1(B) \leq_k I_2(B) \) (because of Lemma 1.2). Given that in the operator \( T_P \) we consider least upper bounds of (negated) interpretations of rule bodies, we obtain \( T_P(I_1) \leq_k T_P(I_2) \).

\[\triangleright\]

2. Fixpoint semantics

**Theorem 2.1:** (Theorem 4.2 in [1]) The operator \( T_P : V \to V \) has a unique least fixpoint\(^1\) w.r.t. the knowledge-ordering.

**Proof** By Theorem 1.1, \( T_P \) is monotonic and by Lemma 1.1, \( \langle V, \leq_k \rangle \) is a complete lattice. By the Knaster-Tarski theorem we then conclude that \( T_P \) has a unique least fixpoint \( \text{lfp}_P \).

\[\triangleright\]

**Theorem 2.2:** (Theorem 4.3 in [1]) Let $P$ be a program. Then:

(a) Every fixpoint of \( T_P : V \to V \) is a model of $P$.

\(^1\)\( F \in V \) is a fixpoint of \( T_P \) iff \( T_P(F) = F \). \( F \) is the least fixpoint of \( T_P \) w.r.t \( \leq_k \) iff \( F \) is a fixpoint of \( T_P \) and \( F \leq_k F' \) for every \( F' \) such that \( T_P(F') = F' \).
(b) The least fixpoint of $T_P : \mathcal{V} \rightarrow \mathcal{V}$ is the least model of $P$ w.r.t. the ordering $\leq_k$.

**Proof**

First we show that over finite domains, the least fixpoint $\mathrm{lfp}_P$ can be computed in a finite number of steps: we consider the empty interpretation $\emptyset \in \mathcal{V}$ and the following inductively defined relation:

\[ T_P^{\uparrow 0} \overset{\text{def}}{=} \emptyset \]
\[ T_P^{\uparrow (z+1)} \overset{\text{def}}{=} T_P(T_P^{\uparrow z}) \]

When the domain is finite, there exists a natural number $n_P \in \mathbb{N}$ such that $T_P^{\uparrow n_P} = \mathrm{lfp}_P$. In fact we observe that $\emptyset \leq_k T_P^{\uparrow 1}$ and because of the monotonicity of $T_P$,

\[ T_P^{\uparrow z} \leq_k T_P^{\uparrow (z+1)} \]

i.e., $T_P^{\uparrow (z+1)} \leq_k T_P^{\uparrow (z+2)}$. Thus $T_P^{\uparrow z} \leq_k T_P^{\uparrow (z+1)}$ for every $z \geq 0$. Therefore, given that the domain is finite, there exists $n_P \in \mathbb{N}$ such that $T_P(T_P^{\uparrow n_P}) = T_P^{\uparrow n_P}$, i.e., $T_P^{\uparrow n_P}$ is a fixpoint of $T_P$. Indeed $T_P^{\uparrow n_P}$ is the least one. In fact, if we consider $\mathrm{lfp}_P$, we have $T^{\uparrow 0} = I_\emptyset \leq_k \mathrm{lfp}_P$; also, if $T_P^{\uparrow z} \leq_k \mathrm{lfp}_P$ then because of the monotonicity of $T_P$ we obtain $T^{\uparrow (z+1)} \leq_k T_P(\mathrm{lfp}_P) = \mathrm{lfp}_P$. Thus $T_P^{\uparrow n_P} \leq_k \mathrm{lfp}_P$.

(a) In the $T_P$ operator we consider least upper bounds of (negated) interpretations of rule bodies. Thus, for a literal $l \in G_P^+$ such that $l \leftarrow B \in P'$, if $I(B) = (i, j) \geq_k t_1$ then $I(l) \geq_k (i, j)$, so $I(l \leftarrow B) = t_N$.

If the rule is of the form $\neg l \leftarrow B \in P'$ with $I(B) = (i, j) \geq_k t_1$, then $I(l) \geq_k (j, i)$, i.e., $I(\neg l) \geq_k (i, j)$, entailing $I(\neg l \leftarrow B) = t_N$.

(b) Let $M$ be a model of $P$: we prove by induction that $T_P^{\uparrow z} \leq_k M$ for every $z \geq 0$. $I_\emptyset \leq_k M$, because $I_\emptyset$ assigns $u$ to every ground literal in $\mathcal{L}_+ \cup \mathcal{L}_-$, whereas the interpretation of logical constants is the same for $I_\emptyset$ and $M$. Let be $T_P^{\uparrow z} \leq_k M$. For an interpretation $I$ and for a literal $l \in G_P^+$ we define:\footnote{We recall that $P'$ is the ground version of $P$ - see Definition 3.1 in [1].}

\[ I_l \overset{\text{def}}{=} \{ I(B) \mid l \leftarrow B \in P', I(B) \geq_k t_1 \} \cup \{ \neg I(B) \mid \neg l \leftarrow B \in P', I(B) \geq_k t_1 \} \]
Given that \( M \) is a model of \( P \), for a positive ground literal \( l \) we have \( M(l) \geq_k \text{lub}_k M \). From inductive hypothesis \( T_P \uparrow z \leq_k M \); thus, from Lemma 1.2, it follows that \( \text{lub}_k M \geq_k \text{lub}_k (T_P \uparrow z)(l) = T_P \uparrow (z+1)(l) \). We conclude that \( M(l) \geq_k T_P \uparrow (z+1)(l) \). If \( l \) is a negative ground literal, i.e., \( l = -l' \) with \( l' \in G^+_P \), then from the previous case we obtain \( M(-l') \geq_k T_P \uparrow (z+1)(-l') \). Concluding, \( T_P \uparrow z \leq_k M \) for every \( z \geq 0 \) and every model \( M \) of \( P \).

From point (a) we obtain that \( T_P \uparrow n_P = \text{lfp}_P \) is also a model of \( P \) so we can conclude that it is the least model of \( P \) w.r.t. the ordering \( \leq_k \).

\[
3. \text{Fixpoint semantics for extended programs}
\]

The following lemmas are used to compute the fixpoint semantics of extended programs.

**Lemma 3.1:** Let \( M_i \) be the least Herbrand model of \( P^t_i \). Then \( \bigcup_{j=0}^i M_j \) is the least Herbrand model of \( \bigcup_{j=0}^{i+1} P^t_j \).

**Proof** We prove it by induction on \( i \). For \( i = 0 \) it is clearly satisfied as \( M_0 = \emptyset = P^t_0 \). Let \( \bigcup_{j=0}^i M_j \) be the least Herbrand model of \( \bigcup_{j=0}^i P^t_j \) and \( M \) a model of \( \bigcup_{j=0}^{i+1} P^t_j \). We consider a positive ground literal \( l \) for which it exists either \( l \leftarrow B \in P^t_k \) or \( -l \leftarrow B \in P^t_k \), for \( k \in \{1,...,i+1\} \).

We define:

\[
I^l \overset{\text{def}}{=} \text{lub}_k \left( \begin{array}{l}
\{l(B)| l \leftarrow B \in P^t_k, I(B) \geq_k t_1\} \cup \\
\{-l(B)| -l \leftarrow B \in P^t_k, I(B) \geq_k t_1\} 
\end{array} \right).
\]

Now let us consider \( l \leftarrow B \in P^t_k \) (the case for \( -l \leftarrow B \in P^t_k \) is similar) and let be \( B = B_1,...,B_n \). For \( h \in \{1,...,n\} \), if \( B_h \) is a logical constant, then \( (\bigcup_{j=0}^{i+1} M_j)(B_h) = \bar{M}(B_h) \). Otherwise, according to the construction of \( P^t_k \), \( B_h \) is a literal that either never appears as head of a rule or it appears as head of a rule of \( P^t_k \). Given that \( \bar{M} \) is also a model of \( P^t_k \), we obtain that \( (\bigcup_{j=0}^{i+1} M_j)(B_h) = M_k(B_h) \leq_k \bar{M}(B_h) \). The same considerations hold for

---

3By \( P^t_k \) we denote the ground version of \( P^t_k \) - see Definition 3.1 in [1].
\[ n \leftarrow B \in P'_{k}. \] From the fact that \( M_k \) is the least Herbrand model of \( P^t_k \) we obtain:

\[ (\bigcup_{j=0}^{i+1} M_j)(l) = M_k(l) = M^t_k. \]

Also, from Lemma 1.2, we obtain:

\[ M^t_k \leq \bar{M} \leq \bar{M}(l). \]

We conclude that \( \bigcup_{j=0}^{i+1} M_j \leq \bar{M} \) for every \( k \in \{1, \ldots, i + 1\} \). From the inductive hypothesis and according to its definition, \( \bigcup_{j=0}^{i+1} M_j \) is also a model of \( \bigcup_{j=0}^{i+1} P^t_j \). Then \( \bigcup_{j=0}^{i+1} M_j \) must be the least Herbrand model of \( \bigcup_{j=0}^{i+1} P^t_j \). \(<

**Lemma 3.2**: \( T'_{P_i+1} \uparrow n_{i+1} \) is a model of \( \bigcup_{j=0}^{i+1} P^t_j \) whenever \( T'_{P_i} \uparrow n_h \) is the least Herbrand model of \( \bigcup_{j=0}^{h} P^t_j \), for every \( 1 \leq h \leq i \) and \( i \geq 1 \).

**Proof**  From the definition of partitions, for every \( j > 1 \) it follows that \( T'_{P_j} \uparrow n_j \) does not alter the interpretation of any ground literal appearing in a program \( P^t_k \), with \( k < j \). From the construction of \( P^t_k \), every literal \( l \) appearing in a rule of \( P^t_k \) either never appears as head of a rule or it appears as a head of a rule in \( P^t_k \). In both cases, \( T'_{P_i+1} \uparrow n_{i+1}(l) = T'_{P_i} \uparrow n_k(l) \).

Also, from the hypothesis of the lemma and from Lemma 3.1 we obtain: \( T'_{P_i} \uparrow n_k(l) = (\bigcup_{j=0}^{k} M_j)(l) = M_k(l) \). Given that \( M_k \) is the least Herbrand model of \( P^t_k \) it satisfies such a rule, thus \( M_k \) and as a consequence also \( T'_{P_i+1} \uparrow n_{i+1} \) satisfy the rule in which \( l \) appears. Given that \( l \) is arbitrary, \( T'_{P_i+1} \uparrow n_{i+1} \) is a model of \( \bigcup_{j=0}^{i} P^t_j \).

Now we observe that \( T'_{P_i+1} \uparrow n_{i+1} \) is a model of \( P_{i+1} \). Excluding logical constants representing introspection operators and literals appearing in a program \( P_h \) with \( h \leq i \), \( P_{i+1} \) is equal to \( P^t_{i+1} \). Let \( T \) stand for an introspection operator appearing in \( P_{i+1} \) or a literal whose head appears in a program \( P_h \) with \( h \leq i \). Then \( T'_{P_{i+1}} \uparrow n_{i+1}(T) = T'_{P_i} \uparrow n_i(T) = (\bigcup_{j=0}^{i} M_j)(T) \), i.e. \( T \) is replaced in \( P_{i+1}^t \) by a logical constant \( B_i \) such that

\[ T'_{P_{i+1}} \uparrow n_{i+1}(B_i) = (\bigcup_{j=0}^{i} M_j)(T) = T'_{P_{i+1}} \uparrow n_{i+1}(T). \]

In other words, the only differences between \( P_{i+1}^t \) and \( P_{i+1} \) are represented by some logical constants whose interpretations under \( T'_{P_{i+1}} \uparrow n_{i+1} \) give the same
truth-degrees of the components interpreted in $P_{i+1}$ through $T_{P_{i+1}}^{i+1}$. Thus, $T_{P_{i+1}}^{i+1} n_{i+1}$ is also a model of $P_{i+1}$. Given that $T_{P_{i+1}}^{i+1} n_{i+1}$ does not alter the interpretation of literals appearing in $\bigcup_{j=0}^{i} P_j$, we conclude that $T_{P_{i+1}}^{i+1} n_{i+1}$ is a model of $\bigcup_{j=0}^{i+1} P_j$.

Lemma 3.3: Let be $T_{P_{i+1}}^{i+1} z \leq k \bigcup_{j=0}^{i+1} M_j$ for a $z \geq 0$ and $T_{P_{i+1}}^{i+1} n_{i+1} = \bigcup_{j=0}^{i} M_j$. Then $T_{P_{i+1}}^{i+1} (z + 1) \leq k \bigcup_{j=0}^{i+1} M_j$.

Proof. Let $l$ be a ground literal.

- If $l$ does not appear as head of a rule in any program $P_h$ with $h \leq i + 1$ then $T_{P_{i+1}}^{i+1}(z + 1)(l) = u$.

- If $l$ appears as head of a rule in a program $P_h$ with $h \leq i$, then $T_{P_{i+1}}^{i+1}(z + 1)(l) = T_{P_{i+1}}^{i+1}(z)(l) \leq_k (\bigcup_{j=0}^{i+1} M_j)(l)$.

- If $l$ appears as a (possibly negated) head of a rule in $P_{i+1}$, then consider:

$$I_l \overset{\text{def}}{=} \text{lub}_k(\{l(B)\mid l \leftarrow B \in P_{i+1}^{t}, I(B) \geq_k t\} \cup \{-l(B)\mid -l \leftarrow B \in P_{i+1}^{t}, I(B) \geq_k t\} \cup \{I(l)\})$$

We have:

$$(T_{P_{i+1}}^{i+1} z)^l = T_{P_{i+1}}^{i+1}(z + 1)(l).$$

We consider a rule $l \leftarrow B \in P_{i+1}^{t}$ with $B = B_1, ..., B_n$ (the case for $-l \leftarrow B \in P_{i+1}^{t}$ is similar). Then there exists a rule $l \leftarrow \hat{B} \in P_{i+1}^{t}$ with $\hat{B} = \hat{B}_1, ..., \hat{B}_n$ such that, for every $j \in \{1, ..., n\}$: (i) either $B_j = \hat{B}_j$ or (ii) $B_j$ is an introspection operator or it is a literal that appears as a head of a program $P_h$ with $h \leq i$ and $\hat{B}_j$ is a logical constant such that $M_{i+1}(\hat{B}_i) = (\bigcup_{j=0}^{i+1} M_j)(B_j)$.

In both cases it follows that:

$$T_{P_{i+1}}^{i+1}(z)(B_i) \leq_k (\bigcup_{j=0}^{i+1} M_j)(\hat{B}_i), \quad (1)$$
in fact, if $B_j = \hat{B}_j$ then that is a direct consequence of the hypothesis of the lemma, otherwise:

$$T'_{P_{i+1}} \uparrow z(B_j) = T'_{P_i} \uparrow n_i(B_j) = (\bigcup_{j=0}^{i} M_j)(B_i) = M_{i+1}(\hat{B}_i) = (\bigcup_{j=0}^{i+1} M_j)(\hat{B}_i).$$

We define:

$$I_l \overset{\text{def}}{=} \text{lub}_k \{l(\hat{B})| l \leftarrow \hat{B} \in P'_{i+1}, I(\hat{B}) \geq_k t_1\} \cup \{\neg l(\hat{B})| \neg l \leftarrow \hat{B} \in P'_{i+1}, I(\hat{B}) \geq_k t_1\} \cup \{I(l)\}.$$

From Lemma 3.1, $\bigcup_{j=0}^{i+1} M_j$ is the least Herbrand model of $\bigcup_{j=0}^{i+1} P_j'$, thus:

$$\left(\bigcup_{j=0}^{i+1} M_j\right)_l = \left(\bigcup_{j=0}^{i+1} M_j\right)(l).$$

From Lemma 1.2, we conclude that:

$$(T'_{P_{i+1}} \uparrow z)^l = T'_{P_{i+1}} \uparrow (z + 1)(l) \leq_k \left(\bigcup_{j=0}^{i+1} M_j\right)_l = \left(\bigcup_{j=0}^{i+1} M_j\right)(l),$$

thus, $T'_{P_{i+1}} \uparrow (z + 1) \leq_k \bigcup_{j=0}^{i+1} M_j$. ◁

We can now prove the following theorem.

**Theorem 3.1:** (Theorem 5.1 in [1]) Let be $P = P_1 \cup ... \cup P_m$ an extended program with a partition $P_1, ..., P_m$. Then $T'_{P_m} \uparrow n_m$ is the least Herbrand model of $P$.

**Proof** We prove the statement by induction on the index $i$ of $P_i$.

- $T'_{P_1} \uparrow 0 = I_0 \leq_k M_1$ and $T'_{P_0} \uparrow 0 = I_0 = M_0$. By applying Lemma 3.3, we obtain $T'_{P_1} \uparrow n_1 \leq_k M_1$. From Lemma 3.2, $T'_{P_1} \uparrow n_1$ is a model of $P'_1$. From Lemma 3.1, $M_1$ is the least Herbrand model of $P'_1 \cup P'_0 = P'_1$. Thus we conclude that $T'_{P_1} \uparrow n_1 = M_1$. ◁
Let the inductive hypothesis (HI) be $T'_P \uparrow n_i = \bigcup_{j=0}^i M_j$ and $T'_P \uparrow n_h$ be the least Herbrand model of $\bigcup_{j=0}^h P^t_j$ for every $1 \leq h < i$. From Lemma 3.1, we directly obtain that $T'_P \uparrow n_i = \bigcup_{j=0}^i M_j$ is the least Herbrand model of $\bigcup_{j=0}^i P^t_j$. We prove by induction on $z$ that $T'_P \uparrow z \leq_k \bigcup_{j=0}^{i+1} M_j$ for every $z \geq 0$. From hypothesis HI,

$$T'_P \uparrow 0 = T'_P \uparrow n_i = \bigcup_{j=0}^i M_j \leq_k \bigcup_{j=0}^{i+1} M_j.$$ 

Let be $T'_P \uparrow z \leq_k \bigcup_{j=0}^{i+1} M_j$. From Lemma 3.3, we obtain that $T'_P \uparrow (z + 1) \leq_k \bigcup_{j=0}^{i+1} M_j$. Thus, $T'_P \uparrow n_{i+1} \leq_k \bigcup_{j=0}^{i+1} M_j$. From Lemma 3.1, $\bigcup_{j=0}^{i+1} M_j$ is the least Herbrand model of $\bigcup_{j=0}^{i+1} P^t_j$. From Lemma 3.2, $T'_P \uparrow n_{i+1}$ is also a model of $\bigcup_{j=0}^{i+1} P^t_j$. Thus we conclude that $T'_P \uparrow n_{i+1} = \bigcup_{j=0}^{i+1} M_j$.

We then conclude that $T'_P \uparrow n_m = \bigcup_{j=0}^m M_j$ is the least Herbrand model of the extended program $P$. \hfill \triangleright
4. Introspection operators and universal quantification

Negation as failure is a non-monotonic inference rule used to entail a negative sentence of the form \textbf{not} \( p \) when the predicate \( p \) cannot be derived from inference rules of the program (depending on the formal system used, the sentence \textbf{not} \( p \) and \( \neg p \) can have different interpretations. In two-valued logic negation as failure can be used to easily model universal quantification; without such an inference rule, expressing universal quantification is more difficult because the whole set of facts must be known in advance. This restriction imposes some limitations when we consider programs as deductive databases, where logic rules represent reasoning components (called intensional database - IDB) using collections of facts (called extensional database - EDB) that may change over time; in this context, to define universal quantification, EDB should be known before defining IDB.

In our logic, the meaning of a sentence of type \textbf{not} \( p \) can be expressed using a sentence of type \( p \in \{ u \} \), i.e. we are able to reproduce negation as failure. Given that De Morgan’s laws do not directly hold in our many-valued logic, we use a different formulation for universal quantification, presented below.

\begin{lemma}
(Section 6 in [1]) Given a domain \( U \overset{\text{def}}{=} \{ a_1, \ldots, a_n \} \subseteq \text{Cons} \), for any many-valued Herbrand interpretation \( I \) we have:

\[ \forall X I(P(X)) \in \{ \tau \} \iff \neg \exists X (\neg I(P(X)) \in \tau_N \setminus \{ \neg \tau \}) . \tag{2} \]
\end{lemma}

\begin{proof}
We recall the definition of the introspection operator “\( \in \)”:

\[ I(O_{\in}(\{ l \}, \{ \tau_1, \ldots, \tau_N \})) \overset{\text{def}}{=} \begin{cases} t_{\tau} & \text{when } I(l) \in \{ \tau_1, \ldots, \tau_N \}; \\ f_{\tau} & \text{otherwise}. \end{cases} \tag{3} \]

According to the definition of negation, \( I(l) = \tau \) if and only if \( I(\neg l) = \neg \tau \). Thus \( \forall X I(P(X)) \in \{ \tau \} \) if and only if \( \forall X I(\neg P(X)) \in \{ \neg \tau \} \). Thus, if \( \forall X I(\neg P(X)) \in \{ \neg \tau \} \) then \( \exists X I(\neg P(X)) \in \tau_N \setminus \{ \neg \tau \} \).

Similarly, if \( \neg \forall X I(P(X)) \in \{ \tau \} \) then \( \exists X I(P(X)) \in \tau_N \setminus \{ \tau \} \), i.e. \( \exists X I(\neg P(X)) \in \tau_N \setminus \{ \neg \tau \} \).
\end{proof}

Note that Lemma 4.1 gives an equivalent formulation of universal quantification that can be used into logic programs what the following example illustrates.
**Example 4.1:** We want to compute the maximum number in the set \( \{1, 3, 7\} \). Assuming \( \tau_N \) with \( N = 1 \), intuitively we search for an integer \( Y \) satisfying \( \max(Y) \):

\[
\max(Y) \text{ iff } \forall X ((X \leq Y) \in \{t_1\}),
\]

where \( \leq \) denotes the standard total order on integers. The following program \( P \) computes the required \( \max \) using equivalences (2) and (4), where \( \maxa \) is an auxiliary relation needed to meet the stratification requirement.

\[
\begin{align*}
\max(X) & \leftarrow \neg \maxa(X) \in \{u\} \\
\neg \maxa(X) & \leftarrow N(X) \in \{t_1\}, N(Y) \in \{t_1\}, \text{less}(X,Y) \in \tau_1 \setminus \{\neg t_1, u\} \\
\text{less}(1,3) & \leftarrow \\
\text{less}(1,7) & \leftarrow \\
\text{less}(3,7) & \leftarrow \\
N(1) & \leftarrow \\
N(3) & \leftarrow \\
N(7) & \leftarrow
\end{align*}
\]

\( P \) is an extended program and the stratification we use is \( P = P_1 \cup P_2 \cup P_3 \) with:

\[
\begin{align*}
P_3 &= \{\max(X) \leftarrow \neg \maxa(X) \in \{u\}\}; \\
P_2 &= \{\neg \maxa(X) \leftarrow N(X) \in \{t_1\}, N(Y) \in \{t_1\}, \text{less}(X,Y) \in \tau_1 \setminus \{\neg t_1, u\}\}; \\
P_1 &= P \setminus (P_2 \cup P_3).
\end{align*}
\]

The second rule of \( P \) defines the condition determining when a number must not be considered the maximum; we have expressed the universal quantification by resorting to the right-hand side of (2). According to Lemma 4.1, the introspection operator should be \( \text{less}(X,Y) \in \tau_1 \setminus \{\neg t_1\} \); in this case we also avoid the cases where \( \text{less}(X,Y) \) is evaluated to \( u \) because, in the current formulation, all ground literals of type \( \text{less}(X,X) \) and \( \text{less}(Y,X) \) (with \( \text{less}(X,Y) \) fact of \( P_1 \)) are evaluated to \( u \), i.e. they would satisfy the second rule producing misleading results. If a number \( X \) never satisfies the body of the rule (w.r.t. any other existing number \( Y \)), there will be no literals of type \( \neg \maxa(X) \) in the semantics of the program, i.e. \( \neg \maxa(X) \) will be evaluated to \( u \); in this case, the first rule will then infer that \( X \) is the
maximum. Computing the fixpoint semantics of $P$ we obtain:

$$T'_P \uparrow 1 = T'_P (I_0) = \{(N(1), t_1), (N(3), t_1), (N(7), t_1), (less(1, 3), t_1), (less(1, 7), t_1), (less(3, 7), t_1)\};$$

$$T'_P \uparrow 2 = T'_P (T'_P \uparrow 1) = T'_P \uparrow 1 = T'_P \uparrow n_1;$$

$$T'_P \uparrow 1 = T'_P (T'_P \uparrow n_1) = T'_P \uparrow n_1 \cup \{(maxa(1), f_1), (maxa(3), f_1)\};$$

$$T'_P \uparrow 2 = T'_P (T'_P \uparrow 1) = T'_P \uparrow 1 = T'_P \uparrow n_2;$$

$$T'_P \uparrow 1 = T'_P (T'_P \uparrow n_2) = T'_P \uparrow n_2 \cup \{(max(7), t_1)\};$$

$$T'_P \uparrow 2 = T'_P (T'_P \uparrow 1) = T'_P \uparrow n_3.$$

In $T'_P \uparrow n_2$, 7 is the only number not appearing as ground literal of $\neg maxa(X)$, so $\neg max(7) \in \{u\}$ is evaluated to $t_1$, obtaining $T'_P \uparrow n_3(max(7)) = t_1$. \hfill △

Acknowledgments

The third author has been supported by the Polish National Science Centre grant 2015/19/B/ST6/02589.