A Sound Semantics for Bousi~Prolog

Pascual Julián-Iránzo and Clemente Rubio-Manzano
Department of Information Technologies and Systems, University of Castilla-La Mancha. 
{Pascual.Julian,Clemente.Rubio}@uclm.es

Abstract. Bousi~Prolog is an extension of the standard Prolog language aiming at to make more flexible the query answering process and to deal with vagueness applying declarative techniques. In this paper we precise a model-theoretic semantics for a pure subset of this language, we recall both the WSLD-resolution principle and a similarity-based unification algorithm which is the basis of its operational mechanism and then we prove the soundness of WSLD-resolution. Keywords: Fuzzy Logic Programming, Declarative Semantics, Fuzzy Herbrand Model, Fixpoint Semantics, Weak Unification, Weak SLD-Resolution, Proximity/Similarity Relations.

1 Introduction

Logic Programming is founded on the idea that logic, or at least substantial subsets of it, can be used as a programming language [8, 18]. Although logic programming has been used in a wide range of applications, it has not direct tools to deal with the essential vagueness or uncertainty of some problems. In recent years there has been a renewed interest in amalgamating logic programming with concepts coming from Fuzzy Logic or akin to this field. As tokens of this interest we mention the works on Fuzzy Logic Programming [5, 11, 12, 19], Qualified Logic Programming [14, 2, 15] (which is a derivation of the van Emden’s Quantitative Logic Programming [17]) or Similarity-Based Logic Programming [3, 4, 9, 16].

Bousi~Prolog (BPL for short) is a representative of the last class of fuzzy logic programming languages. It replaces the syntactic unification mechanism of classical SLD-resolution by a fuzzy unification algorithm based on fuzzy binary relations on a syntactic domain. This algorithm provides a weak most general unifier as well as a numerical value, called the approximation degree. Intuitively, the approximation degree represents the truth degree associated with the (query) computed instance. The result is an operational mechanism, called Weak SLD-resolution, which differs in some aspects w.r.t. the one of [16], based exclusively on similarity relations.

This work can be seen as a continuation of the investigation started in [6]. In this paper after introducing some refinements to the model-theoretic and fix-point semantics of Bousi~Prolog defined in [6] for definite programs, we introduce for the first time in our framework the concept of a correct answer providing a declarative description for the output of a program and a goal. It
is noteworthy that, although the refinements introduced in the declarative semantics do not dramatically alter the original definitions, given in [6], they are important in order to establish the soundness of our proposal.

Afterwards, we recall the operational semantics of Bousi-Prolog and we prove, among other results, its soundness. The soundness theorem is established following a proof strategy comparable with the one appeared in [8]. It is important to remark that, the soundness in our framework will be proven under certain conditions. To be precise, we only consider programs without negation and we restrict ourselves to similarity relations on syntactic domains.

Finally, it is worthy to say that, along this paper we also clarify some of the existing differences between our framework and the related proposal introduced by [16].

2 Preliminaries.

2.1 Fuzzy relations, proximity and similarity relations

A binary ordinary relation on \( U \) is a subset of \( U \times U \) and it can be identified by its characteristic function \( U \times U \rightarrow \{0, 1\} \). Therefore, similarly to the case of ordinary sets, the easy extension of this concept to the fuzzy case is to agree that, a binary fuzzy relation on a set \( U \) is a fuzzy subset on \( U \times U \) (that is, a mapping \( U \times U \rightarrow [0, 1] \)). Since fuzzy relations are fuzzy subsets they have \( \lambda \)-cuts. If \( R \) is a fuzzy relation on \( U \), the \( \lambda \)-cut \( R_\lambda = \{ (x,y) \mid R(x,y) \geq \lambda \} \) (which is an ordinary relation on \( U \)). There are some important properties that fuzzy relations may have:

1. (Reflexive) \( R(x,x) = 1 \) for any \( x \in U \);
2. (Symmetric) \( R(x,y) = R(y,x) \) for any \( x, y \in U \);
3. (Transitive) \( R(x,z) \geq R(x,y) \triangle R(y,z) \) for any \( x, y, z \in U \);

where the operator ‘\( \triangle \)’ is an arbitrary t-norm. The notion of transitivity above is \( \Delta \)-transitive, if the operator \( \Delta = \wedge \) (that is, it is the minimum of two elements), we speak of mim-transitive or \( \wedge \)-transitive.

A proximity relation is a binary fuzzy relation which is reflexive and symmetric. A proximity relation is characterized by a set \( \Lambda = \{ \lambda_1, ..., \lambda_n \} \) of approximation levels. We say that a value \( \lambda \in \Lambda \) is a cut value.

A special, and well-known, type of proximity relations are similarity relations, which are nothing but transitive proximity relations. As we have just mentioned, in this paper we pay a special attention on similarity relations on syntactic domains.

In classical logic programming different syntactic symbols represent distinct information. Following [16], this restriction can be relaxed by introducing a proximity or similarity relation \( R \) on the alphabet of a first order language, allowing \( R \) to provide a possible non-zero value for function/predicate symbols with the same arity, whereas it is the identity relation for variables. This makes possible to treat as indistinguishable two syntactic symbols which are related by the proximity or similarity relation \( R \) with a certain degree greater than zero.
The similarity relation $\mathcal{R}$ on the alphabet of a first order language can be extended to terms by structural induction in the usual way\(^1\):

1. $\mathcal{R}(X, X) = 1$;
2. Let $f$ and $g$ be two $n$-ary function symbols and let $t_1, \ldots, t_n, s_1, \ldots, s_n$ be terms. $\mathcal{R}(f(t_1, \ldots, t_n), g(s_1, \ldots, s_n)) = \mathcal{R}(f, g) \land (\bigwedge_{i=1}^n \mathcal{R}(t_i, s_i))$;

Otherwise, the approximation degree of two expressions is zero. The extension for atomic formulas and compound formulas can be done in an analogous form.

### 2.2 Formulas, interpretations and truth in the context of a proximity relation

A first order theory consists of an alphabet, a first order language, a set of axioms and a set of inference rules [8]. The first order language consists of the well-formed formulas of the theory. In this section we discuss the notions of interpretation and truth for a first order theory in the context of a proximity relation.

In classical logic the meaning of a formula is defined relatively to an interpretation (also called structure) which specifies a meaning for each symbol in the formula. An interpreted formula is either true or false. However, on our framework, since we use similarity relations on the lattice $[0, 1]$, truth is a matter of grade.

A fuzzy Interpretation $\mathcal{I}$ of a first order language $\mathcal{L}$ is a pair $\langle \mathcal{D}, \mathcal{I} \rangle$ where $\mathcal{D}$ is the domain of the interpretation and $\mathcal{I}$ is a mapping which assigns: (i) for each constant symbol $a$ an element $\mathcal{I}(a) = \bar{a} \in \mathcal{D}$; (ii) for each $n$-ary function symbol $f$ a mapping $\mathcal{I}(f) = \bar{f}$, such that $\bar{f} : \mathcal{D}^n \rightarrow \mathcal{D}$; (iii) if $r^n$ is a $n$-ary relation symbol then a $n$-ary fuzzy relation $\mathcal{I}(r^n) = \bar{r}^n \subset \mathcal{D}^n$ (i.e., a mapping $\bar{r}^n : \mathcal{D}^n \rightarrow [0, 1]$).

The meaning of a formula will be defined as a function of the meaning of its components. As in the classical case, in order to evaluate open formulas we need to introduce the notion of variable assignment. A variable assignment, $\vartheta$, with respect to the fuzzy interpretation $\mathcal{I} = \langle \mathcal{D}_I, \mathcal{J} \rangle$, is a mapping from the set of variables $\mathcal{V}$ in $\mathcal{L}$ to the elements in the domain of $\mathcal{I}$ (i.e. $\vartheta : \mathcal{V} \rightarrow \mathcal{D}_I$). The concept of an assignment can be extended in a natural way to the set of the terms $\mathcal{T}$ of $\mathcal{L}$ by structural induction.

Given a fuzzy interpretation $\mathcal{I} = \langle \mathcal{D}, \mathcal{J} \rangle$ and a variable assignment $\vartheta$ in $\mathcal{I}$, the valuation of a formula w.r.t. $\mathcal{I}$ and $\vartheta$ is:

\[
\mathcal{I}(p(t_1, \ldots, t_n))[\vartheta] = \bar{p}(t_1\vartheta, \ldots, t_n\vartheta), \text{ where } \mathcal{J}(p) = \bar{p} \\
\mathcal{I}(A \land B)[\vartheta] = \inf\{\mathcal{I}(A)[\vartheta], \mathcal{I}(B)[\vartheta]\} \\
\mathcal{I}(A \rightarrow B)[\vartheta] = \left\{ \begin{array}{ll} 1 & \text{if } \mathcal{I}(Q)[\vartheta] \leq \mathcal{I}(A)[\vartheta] \\ \mathcal{I}(A)[\vartheta] & \text{if } \mathcal{I}(Q)[\vartheta] > \mathcal{I}(A)[\vartheta] \end{array} \right. \\
\mathcal{I}((\forall x)A)[\vartheta] = \inf\{\mathcal{I}(A)[\vartheta'] \mid \vartheta' \ x-\text{equivalent to } \vartheta\}
\]

\(^1\) The extension of a proximity relation is more cumbersome and it is not treated in this paper.
where \( p \) is a predicate symbol, and \( A \) and \( B \) formulas. An assignment \( \vartheta' \) is \( x \)-equivalent to \( \vartheta \) when \( z\vartheta' = z\vartheta \) for all variable \( z \neq x \) in \( \mathcal{V} \). When the assignment would not be relevant, we shall omit it during the valuation of a formula. At this point, it is noteworthy that the implication operator is being interpreted as an \( R \)-implication (See [13]). The valuation of disjunctive and existential formulas is done as in the case of conjunctive and universal formulas (respectively) but replacing the supremum with the infimum. Negation will deserve further comments soon.

In the context of a first order theory equipped with a proximity relation \( R \), characterized by a set \( \Lambda = \{\lambda_1, \ldots, \lambda_n\} \) of approximation levels, it makes sense that the notion of truth be linked to a certain approximation level \( \lambda \in \Lambda \): for a fixed value \( \lambda \), we consider that a formula \( A \) is true in a fuzzy interpretation \( I \) if, when interpreted, its truth value is equal or greater than \( \lambda \). We say that \( \lambda \) is the considered cut value (Intuitively, a cut value \( \lambda \) is delimiting truth degrees equal or greater than \( \lambda \) as true).

**Definition 1.** Let \( I \) be a fuzzy interpretation of a first-order language \( L \), \( R \) be a proximity relation with a set \( \Lambda \) of approximation levels and a cut value \( \lambda \in \Lambda \). Let \( A \) be a formula of \( L \):

- \( A \) is \( \lambda \)-true in \( I \) iff for every assignment \( \vartheta \) in \( I \), \( I(A)[\vartheta] \geq \lambda \).
- \( A \) is \( \lambda \)-false in \( I \) iff for every assignment \( \vartheta \) in \( I \), \( I(A)[\vartheta] < \lambda \).
- \( A \) is \( \lambda \)-valid iff \( A \) is \( \lambda \)-true for all interpretation \( I \).
- \( A \) is \( \lambda \)-unsatisfiable iff \( A \) is \( \lambda \)-false for all \( I \).
- \( A \) is \( \lambda \)-satisfiable iff there exists an \( I \) and a \( \vartheta \) in \( I \) such that \( I(A)[\vartheta] \geq \lambda \).

The valuation of a closed formula is completely determined by an interpretation, independently of a variable assignment. Therefore we can speak about the truth value of a formula in an interpretation, leading to the notion of a model for a closed formula.

**Definition 2 (\( \lambda \)-Model for a closed formula).** Let \( R \) be a proximity relation which is characterised by a set \( \Lambda \) of approximation levels and a cut value \( \lambda \in \Lambda \). Let \( I \) be an interpretation of a first order language \( L \) and \( A \) be a closed formula. Then, \( I \) is \( \lambda \)-model for \( A \) if and only if \( I(A) \geq \lambda \).

Ending this subsection we like to introduce a remark about the interpretation of negate formulas. The interpretation of formulas with other connectives different of negation is independent of the cut value \( \lambda \) used in presence of a similarity relation. However, we note that in order to preserve some valuable classical properties the interpretation of negate formulas must be dependent on the cut value \( \lambda \):

\[
I(\neg A)[\vartheta] = \begin{cases} 
1 & \text{if } I(A)[\vartheta] < \lambda \\
0 & \text{if } I(A)[\vartheta] \geq \lambda 
\end{cases}
\]

This form of negation, clearly inspired on Gödel’s negation, preserves some important properties for our framework. For instance, it is possible to prove that \( A \) is \( \lambda \)-true in \( I \) if and only if \( \neg A \) is \( \lambda \)-false in \( I \).

Take notice that, from now onwards, we mainly deal with closed formulas, unless we say the contrary.
2.3 Closed conditional formulas and models

In this section we elucidate the notion of model for a set of closed conditional formulas in the context of a similarity relation. By conditional formula we mean a formula of the form \( C \equiv A \leftarrow Q \), where \( A \) (called the head) is an atom, \( Q \) a formula (called the body) and all variables are assumed universally quantified. When \( Q \equiv B_1 \wedge \ldots \wedge B_n \) is a conjunction of atoms, the formula \( C \) is called a Horn clause or definite clause. As it is well known, this kind of formulas play a special role in logic programming where a set of definite clauses is called a program and a goal is any conjunctive body.

A direct naive translation to our context of the classical concept of model for a set of formulas does not work. We need a new definition supported by the notion of what we called an annotated set of formulas of level \( \lambda \). In particular, if we are working with the set \( \Gamma \equiv \{ p(a) \} \) and the similarity defined by the entry \( \mathcal{R}(a, b) = 0.8 \) then the intended meaning of \( \Gamma \) and \( \mathcal{R} \) is that we believe in \( p(a) \) with truth degree 1 but also, because \( b \) is similar to \( a \), we believe in \( p(b) \) with truth degree 0.8. That is \( \mathcal{R} \) induces meaning into \( \Gamma \) and we can reflect this fact by means of an annotated set of formulas \( \{ (p(a), 1), (p(b), 0.8) \} \) (see [6] to obtain more intuitive insights on this idea).

We want to formalize this concept of an annotated set of formulas, but before doing that we need some technical definitions introduced to cope with some problems that appear when conditional formulas have non-linear atoms on their heads\(^2\). Given a non-linear atom \( A \), the linearization of \( A \) (as defined in [2]) is a process by which it is computed the structure \( \langle A_l, C_l \rangle \), where \( A_l \) is a linear atom built from \( A \) by replacing each one of the \( n \) multiple occurrences of the same variable \( X_i \) by new fresh variables \( Y_{ki} (1 \leq k \leq n_i) \); and \( S_i \) is a set of proximity constrains \( X_i \sim Y_k (1 \leq k \leq n_i) \). The operator “\( s \sim t \)” is asserting the proximity of two terms \( s \) and \( t \) and when interpreted, \( I(s \sim t) = \mathcal{R}(s, t) \), whatever the interpretation \( I \) of \( \mathcal{L} \). Now, let \( C \equiv A \leftarrow Q \) be a conditional formula and \( S_l = \{ X_1 \sim Y_{11}, \ldots, X_n \sim Y_{n1} \} \), \( lin(C) = A_l \leftarrow X_1 \sim Y_1 \wedge \ldots \wedge X_n \sim Y_n \wedge Q \). For a set \( \Gamma \) of conditional formulas, \( lin(\Gamma) = \{ lin(C) \mid C \in \Gamma \} \).

The following algorithm, which is a reformulation of the one that appears in [6] to cope with the linearization process, gives a precise procedure for the construction of the set of annotated formulas of level \( \lambda \).

Algorithm 1

Input: A set of conditional formulas \( \Gamma \) and a proximity relation \( \mathcal{R} \) with a set of levels \( \Lambda \) and a cut value \( \lambda \in \Lambda \).

Output: A set \( \Gamma^\lambda \) of annotated formulas of level \( \lambda \).

Initialization: \( \Gamma_1 := lin(\Gamma) \) and \( \Gamma^\lambda := \{ \langle C, 1 \rangle \mid C \in \Gamma_1 \} \).

For each conditional formula \( C \equiv A \leftarrow Q \in \Gamma_1 \) do:

- \( K_\lambda(C) = \{ \langle C', A' \leftarrow Q, \alpha \rangle \mid \mathcal{R}(A, A') = \alpha \geq \lambda \} \)
- For each element \( \langle C', A' \rangle \) in \( K_\lambda(C) \) do:

\(^2\) The apparition of this problem in our framework was pointed out by R. Caballero, M. Rodríguez and C. Romero in a private communication. So we want to express them our gratitude.
If \((C', L) \in \Gamma^\lambda\) then 
\[
\Gamma^\lambda = (\Gamma^\lambda \setminus \{(C', L)\}) \cup (C', L \land \alpha)
\]
else 
\[
\Gamma^\lambda = (\Gamma^\lambda \cup \{(C', \alpha)\})
\]

Return \(\Gamma^\lambda\)

The general idea behind this algorithm is to start annotating each formula in the set \(I_l\) with a truth degree equal to 1. On the other hand, the rest of the formulas generated by proximity, starting from formulas of the original set \(I_l\), are annotated with its corresponding approximation degree (regarding the original formula). Afterward, if several formulas of the set generate the same approximate formula, with different approximations degrees, we take the least degree as annotation.

Now we are ready to define the concept of a model for a set of closed conditional formulas and a proximity relation.

**Definition 3 (\(\lambda\)-Model for a set of closed conditional formulas).** Let \(\Gamma\) be a set of closed conditional formulas of a first order language \(L\), \(R\) be a proximity relation which is characterised by a set \(\Lambda\) of approximation levels with cut value \(\lambda \in \Lambda\) and \(I\) be an fuzzy interpretation of \(L\). We say \(I\) is \(\lambda\)-model for \(\{\Gamma, R\}\) iff for all annotated formula \(\langle A, \lambda'\rangle \in \Gamma^\lambda\), \(I(A) \geq \lambda'\).

Now we can give the definition of the core concept of logical consequence of level \(\lambda\).

**Definition 4 (\(\lambda\)-logical consequence).** Given a set of closed formulas, \(\Gamma\), a proximity relation, \(R\), and a closed formula \(A\) of a first order language \(L\). \(A\) is a \(\lambda\)-logical consequence of \(\{\Gamma, R\}\) (denoted \(\{\Gamma, R\} \models^\lambda A\)) if and only if for each fuzzy interpretation \(I\) of \(L\), \(I\) is a \(\lambda\)-model for \(\{\Gamma, R\}\) implies that \(I\) is a \(\lambda\)-model for \(A\).

3 Declarative Semantics.

In this section we recall the declarative semantics of Bousi-Prolog. Roughly speaking, BPL programs are sequences of (normal) clauses plus a proximity relation. However, in this and the following sections we restrict ourselves to definite clauses.

3.1 Fuzzy Herbrand interpretations and models

When we use a logic programming language whose instructions are clauses and employs a refutation procedure, it is well-known that it suffices to pay attention only on (fuzzy) Herbrand interpretations in order to determine the unsatisfiability of a set of clauses. Herbrand interpretations are defined on a syntactic domain, called the Herbrand universe. For a first order language \(L\), the Herbrand universe \(U_L\) for \(L\), is the set of all ground terms in \(L\). Roughly speaking, in a Herbrand interpretation, constant and function symbols are interpreted as themselves in
a fixed way while n-ary relation symbols are freely interpreted as n-ary (fuzzy) relations on \( U \). i.e. (fuzzy) subsets on \( U^2 \) (or equivalently, mappings from \( U^2 \) into the \([0,1]\) interval). On the order hand, the Herbrand base \( B \) for \( L \) is the set of all ground atoms which can be formed by using the predicate symbols of \( L \) jointly with the ground terms from the Herbrand universe taken as arguments. As in the classical case, it is possible to identify a Herbrand interpretation with a fuzzy subset of the Herbrand base. That is, a fuzzy Herbrand interpretation for \( L \) can be considered as a mapping \( I: B \rightarrow [0,1] \). The ordering \( \leq \) in the lattice \([0,1]\) can be easily extended to the set of Herbrand interpretations \( H \), as follows: \( I_1 \sqsubseteq I_2 \) iff \( I_1(A) \leq I_2(A) \) for all ground atom \( A \in B \). It is important to note that the pair \( \langle H, \sqsubseteq \rangle \) is a complete lattice.

In the following, we focus our attention on Herbrand \( \lambda \)-models. For this special kind of \( \lambda \)-models we proved in [6] an analogous property to the model intersection property and we defined the least Herbrand model of level \( \lambda \), for a program \( H \) and a proximity relation \( R \), as the mapping \( M^\lambda_H : B \rightarrow [0,1] \) such that,

\[
M^\lambda_H(A) = \inf\{I(A) | I \text{ is a } \lambda \text{-model for } H \text{ and } R\},
\]

for each \( A \in B \). The interpretation \( M^\lambda_H \) is the natural interpretation for a program \( H \) and a proximity relation \( R \), since, as it was proved in [6], for each \( A \in B \) such that \( M^\lambda_H(A) \neq 0 \), \( A \) is a logical consequence of level \( \lambda \) for \( H \) and \( R \).

### 3.2 Fixpoint semantics

In this section we recall an alternative characterization of the least Herbrand model of level \( \lambda \) for a definite program \( H \) and a proximity relation \( R \), using fixpoint concepts, given in [6]. The idea is to provide a constructive vision of the meaning of a program by defining an immediate consequences operator which allows to construct the least Herbrand model of level \( \lambda \), by means of successive applications.

**Definition 5 (Immediate consequences operator of level \( \lambda \)).** Let \( H \) be a definite program and \( R \) be a proximity relation. We define the immediate consequences operator of level \( \lambda \), \( T^\lambda_H \), as a mapping \( T^\lambda_H : H \rightarrow H \) such that, for all \( A \in B \),

\[
T^\lambda_H(I)(A) = \inf\{\mathcal{P}T^\lambda_H(I)(A)\}
\]

where \( \mathcal{P}T^\lambda_H \) is a non deterministic operator such that \( \mathcal{P}T^\lambda_H(I) : B \rightarrow \wp([0,1]) \) and it is defined as follows: Let \( \Pi = \text{lin}(H) \),

1. For each fact \( H \in \Pi \), let \( K_\lambda(H) = \{(H', \lambda') | R(H, H') = \lambda' \geq \lambda\} \) be the set of approximate atoms of level \( \lambda \) for \( H \). Then

\[
\mathcal{P}T^\lambda_H(I)(H' \vartheta) \ni \lambda',
\]

for all \( H' \) and assignment \( \vartheta \).
2. For each clause $C \equiv (A \leftarrow Q) \in \Pi_I$. Let $K_\lambda(C) = \{ \langle C' \equiv A' \leftarrow Q, \lambda' \rangle \mid R(A, A') = \lambda' \geq \lambda \}$ be the set of approximate clauses of level $\lambda$ for $C$. Then

$$PT_\lambda^{\Pi_I}(I'(A\theta)) \ni \lambda' \land I(Q\theta),$$

for all $C'$ and assignment $\theta$.

In [6], we proved that the immediate consequences operator (of level $\lambda$) is monotonous and continuous and the least fuzzy Herbrand model (of level $\lambda$) coincides with its least fixpoint.

### 3.3 Correct Answer

In this section we define the concept of a correct answer, which provide a declarative description of the desired output from a program, a proximity relation, and a goal. This is a central concept for the later theoretical developments.

**Definition 6 (Answer of level $\lambda$.).** Let $\Pi$ be a definite program, $R$ be a proximity relation, which is characterised by a set $\Lambda$ of approximation levels with cut value $\lambda \in \Lambda$, and $G$ be a goal. An answer of level $\lambda$ for $\{\Pi, R\}$ and $G$ is a pair $\langle \theta, \beta \rangle$ where $\theta$ is any substitution for variables of $G$ and $\beta$ an approximation degree such that $\lambda \leq \beta \leq 1$.

**Definition 7 (Correct Answer of level $\lambda$.).** Let $\Pi$ be a definite program and $R$ be a proximity relation, which is characterised by a set $\Lambda$ of approximation levels with cut value $\lambda \in \Lambda$. Let $G \equiv A_1, \ldots, A_k$ be a goal and $\langle \theta, \beta \rangle$ an answer of level $\lambda$ for $\{\Pi, R\}$ and $G$. We say that $\langle \theta, \beta \rangle$ is a correct answer of level $\lambda$ for $\{\Pi, R\}$ and $G$ if and only if:

1. $\forall(A_1, \ldots, A_k)\theta$ is a $\lambda$-logical consequence of $\{\Pi, R\}$, and
2. $M^{\lambda}_{\Pi}(\forall(A_1, \ldots, A_k)\theta) \leq \beta$.

### 4 Operational Semantics.

The operational semantics of Bousi~Prolog is an adaptation of the SLD resolution principle, where classical unification has been replaced by a fuzzy unification algorithm. In this section we recall the features of both the fuzzy unification algorithm and the resolution procedure. Observe that, since the fuzzy unification algorithm we are going to work with is only well-defined for similarity relations, in this section we only deal with this kind of relations.

#### 4.1 Weak Unification based on similarity relations.

Bousi~Prolog uses a weak unification algorithm that, when we work with similarity relations, coincides with the one defined by M. Sessa [16]. However, there
exists some remarkable differences between our proposal and Sessa’s proposal that we shall treat to put in evidence along this section.

In presence of similarity relations on syntactic domains, it is possible to define an extended notion of a unifier and a more general unifier of two expressions³.

**Definition 8.** Let \( \mathcal{R} \) be a proximity relation, \( \lambda \) be a cut value and \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) be two expressions. The substitution \( \theta \) is a weak unifier of level \( \lambda \) for \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) w.r.t \( \mathcal{R} \) (or \( \lambda \)-unifier) if its unification degree, \( \text{Deg}_\mathcal{R}(\mathcal{E}_1\theta,\mathcal{E}_2\theta) = \mathcal{R}(\mathcal{E}_1\theta,\mathcal{E}_2\theta) \), is greater than \( \lambda \).

Note that in Sessa’s proposal the idea of “cut value” is missed. Also in order that a substitution \( \theta \) be a weak unifier for \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) she put a strong constrain: the unification degree of \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) w.r.t. \( \theta \) must be the maximum of the unification degrees of \( \text{Deg}_\mathcal{R}(\mathcal{E}_1\varphi,\mathcal{E}_2\varphi) \) for whatever substitution \( \varphi \). Therefore, some substitution that we consider as a weak unifier, are disregarded by her proposal.

**Definition 9.** Let \( \mathcal{R} \) be a proximity relation and \( \lambda \) be a cut value. The substitution \( \theta \) is more general than the substitution \( \sigma \) with level \( \lambda \), denoted by \( \theta \preceq_{\mathcal{R},\lambda} \sigma \), if there exist a substitution \( \delta \) such that, for any variable \( x \) in the domain of \( \theta \) or \( \sigma \), \( \mathcal{R}(x\sigma,x\sigma\delta) \geq \lambda \).

**Definition 10.** Let \( \mathcal{R} \) be a proximity relation and \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) be two expressions. The substitution \( \theta \) is a weak most general unifier (w.m.g.u.) of \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) w.r.t \( \mathcal{R} \), denoted by \( \text{wmgu}(\mathcal{E}_1,\mathcal{E}_2) \), if:

1. \( \theta \) is a \( \lambda \)-unifier of \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \); and
2. \( \theta \preceq_{\mathcal{R},\lambda} \sigma \), for any \( \lambda \)-unifier \( \sigma \) of \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \).

The weak unification algorithm we are using is a reformulation of the one appeared in [16], which, in turn, is an extension of Martelli and Montanari’s unification algorithm for syntactic unification [1, 7, 10]. The main difference is regarding the so called decomposition rule ⁴: Given the unification problem \( \langle \{f(t_1,\ldots,t_n) \approx g(s_1,\ldots,s_n)\} \cup \mathcal{E},\sigma,\alpha \rangle \), if \( \mathcal{R}(f,g) = \beta > \lambda \), it is not a failure but it is equivalent to solve the new configuration \( \langle \{t_1 \approx s_1,\ldots,t_n \approx s_n\} \cup \mathcal{E},\sigma,\alpha \land \beta \rangle \), where the approximation degree \( \alpha \) has been compounded with the degree \( \beta \). It is important to note that, differently to [16], the resulting approximation degree is casted by a cut value \( \lambda \).

The weak unification algorithm allows us to check if a set of expressions \( S = \{\mathcal{E}_1 \approx \mathcal{E}_1',\ldots,\mathcal{E}_n \approx \mathcal{E}_n'\} \) is weakly unifiable. The w.m.g.u. of the set \( S \) is denoted by \( \text{wmgu}(S) \). In general, a w.m.g.u. of two expressions \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) is not unique [16]. Therefore, the weak unification algorithm computes a representative of a w.m.g.u. class.

³ We mean by “expression” a first order term or an atomic formula.
⁴ Here, the symbol “\( \mathcal{E}_1 \approx \mathcal{E}_2' \)” represents the potential possibility that two expressions \( \mathcal{E}_1 \) and \( \mathcal{E}_2 \) be close.
4.2 Weak SLD-Resolution.

Let \( \Pi \) be a set of Horn clauses and \( \mathcal{R} \) a similarity relation on the alphabet of a first order language \( \mathcal{L} \). Let \( \Lambda \) be the set of approximation levels of \( \mathcal{R} \). We define Weak SLD (WSLD) resolution as a labeled transition system \( \langle \text{Goals, Labels}, \rightarrow_{\text{WSLD}} \rangle \), where Goals is the set of goals of \( \mathcal{L} \), Labels is a set of triples \( \langle \mathcal{C}, \theta, \alpha \rangle \) (Clause, substitution, approximation degree) and whose transition relation \( \rightarrow_{\text{WSLD}} \subseteq (\text{Goals} \times \text{Labels} \times \text{Goals}) \) is the smallest relation that satisfies:

\[
\begin{align*}
\mathcal{C} = (A \leftarrow Q) \ll \Pi, \sigma = \text{wmg}(A, A') \neq \text{fail}, \quad \beta = \mathcal{R}(A\sigma, A'\sigma)) \geq \lambda \\
\quad \quad \rightarrow A'Q' \xrightarrow{[C,\sigma,\beta]}_{\text{WSLD}} (Q',\sigma)\sigma
\end{align*}
\]

where \( Q, Q' \) are conjunctions of atoms, the notation “\( C \ll \Pi \)” is representing that \( C \) is a standardized apart clause in \( \Pi \), and the value \( \lambda \) is a cut value in \( \Lambda \), which imposes a limit to the expansion of the search space in a computation. We say that the performed step is a step of level \( \lambda \) because the computed approximation degree is greater or equal than \( \lambda \).

A WSLD derivation of level \( \lambda \) for \( \Pi \cup \{G_0\} \) and \( \mathcal{R} \) is a sequence of steps of level \( \lambda \): \( G_0 \xrightarrow{[C_1,\theta_1,\beta_1]}_{\text{WSLD}} \ldots \xrightarrow{[C_n,\theta_n,\beta_n]}_{\text{WSLD}} G_n \). That is, each \( \beta_i \geq \lambda \). And a WSLD refutation of level \( \lambda \) for \( \Pi \cup \{G_0\} \) and \( \mathcal{R} \) is a WSLD derivation of level \( \lambda \) for \( \Pi \cup \{G_0\} \) and \( \mathcal{R} \): \( G_0 \xrightarrow{\theta_\lambda}_{\text{WSLD}} \square \), where the symbol “\( \square \)” stands for the empty clause, \( \theta = \theta_1\theta_2\ldots\theta_n \) is the computed substitution and \( \beta = \prod_{i=1}^{n} \beta_i \) is its approximation degree. The output of a WSLD refutation is the pair \( \langle \theta_\lambda(\forall \mathcal{A}(G)), \beta \rangle \), which is said to be the computed answer. Certainly, a WSLD refutation computes a family of answers, in the sense that, if \( \theta = \{x_1/t_1, \ldots, x_n/t_n\} \) then, by definition, whatever substitution \( \theta' = \{x_1/s_1, \ldots, x_n/s_n\} \), holding that \( R(s_i, t_i) \geq \lambda \), for any \( 1 \leq i \leq n \), is also a computed substitution with approximation degree \( \beta \land (\prod_{i=1}^{n} R(s_i, t_i)) \).

Observe that our definition of proximity based SLD resolution is parameterized by a cut value \( \lambda \in \Lambda \). This introduces an important conceptual distinction between our approach an the similarity based SLD resolution presented in [16]. Moreover, we differ in the way we obtain a family of computed answers (see [6] for details). This may have a determinant impact in the correctness of the overall proposal.

5 Soundness of WSLD-Resolution

In this section we establish the soundness of WSLD-Resolution, but before proving the main result of the paper we need to introduce some important intermediate lemmas.

Lemma 1. Let \( \Pi \) be a definite program, \( \mathcal{R} \) be a proximity relation and \( \lambda \) be a cut value. Given \( (A \leftarrow Q) \in \Pi \) and \( A' \) an atom such that \( R(A, A') = \alpha \geq \lambda \). If \( (\forall Q) \) is a \( \lambda \)-logical consequence of \( \{\Pi, R\} \) then \( (\forall A') \) is a \( \lambda \)-logical consequence of \( \{\Pi, R\} \).
Proof. If $\forall Q$ is an $\lambda$-logical consequence of $\{\Pi, R\}$, then for all $I$, $\lambda$-model for $\{\Pi, R\}$, $I$ is a $\lambda$-model for $\forall Q$. That is, $I(\forall Q) \geq \lambda$ and hence $I(Q\theta) \geq \lambda$ for every assignment $\theta$.

On the other hand, if $(A \leftarrow Q) \in \Pi$ and $R(A, A') = \alpha \geq \lambda$, by definition of annotated program, $\Pi^\lambda$, there exists an annotated clause $(A' \leftarrow Q, \alpha) \in \Pi^\lambda$. Moreover, by definition of $\lambda$-model for $\{\Pi, R\}$, $I(\forall (A' \leftarrow Q)) \geq \alpha \geq \lambda$ and hence $I(A'\theta \leftarrow Q\theta) \geq \alpha \geq \lambda$ for every assignment $\theta$. Now, there are two cases by definition of valuation:

1. $I(A'\theta \leftarrow Q\theta) = 1$ because $I(A'\theta) \geq I(Q\theta) \geq \lambda$.
2. $I(A'\theta \leftarrow Q\theta) = I(A'\theta)$ because $I(Q\theta) > I(A'\theta)$. But, as we just mentioned, $I(A'\theta \leftarrow Q\theta) \geq \alpha \geq \lambda$ and therefore $I(A'\theta) \geq \lambda$.

So, in both cases $I(A'\theta) \geq \lambda$ for every assignment $\theta$. That is, $I(\forall A') = \inf \{I(A'\theta) \mid \theta$ assignment $\} \geq \lambda$. Therefore $(\forall A')$ is a $\lambda$-logical consequence of $\{\Pi, R\}$.

Lemma 2. Let $A$ and $B$ be two atoms such that $A \leq B$. Then, $I(\forall A) \leq I(\forall B)$.

Proof. Immediate because as $A \leq B$ then it implies that there exists a substitution $\xi$, such that $B = A\xi$ and moreover: $I(\forall A) = \inf \{I(A\theta) \mid \theta$ assignment $\} \leq \inf \{I(A\xi\theta') \mid \theta'$ assignment $\} = I(\forall A\xi) = I(\forall B)$.

Corollary 1. Let $\Pi$ be a definite program, $R$ be a proximity relation and $\lambda$ be a cut value. Given $(A \leftarrow Q) \in \Pi$ and $A'$ an atom such that $R(A, A') = \alpha \geq \lambda$. If $(\forall Q\theta)$ is a $\lambda$-logical consequence of $\{\Pi, R\}$ then $(\forall A'\theta)$ is a $\lambda$-logical consequence of $\{\Pi, R\}$, whatever the substitution $\theta$ is.

Proof. Immediate by Lemma 1 and Lemma 2.

Corollary 2. Let $\Pi$ be a program, $R$ be a proximity relation and $\lambda$ be a level of cut. Given $A \leftarrow \xi$ in $\Pi$ and $A'$ an atom such that $R(A, A') = \alpha \geq \lambda$. Then $(\forall A'\theta)$ is an $\lambda$-logical consequence of $\Pi$ for every substitution $\theta$.

Proof. Trivial, since this is a specific case of Corollary 1, when the clause is a fact. Then, if $A$ is a program fact, the atoms $A'$ close to $A$ and their instances are $\lambda$-logical consequence of $\{\Pi, R\}$.

Lemma 3. Let $A$ and $B$ be two atoms, $R$ be a proximity relation, with cut level $\lambda$, and $\theta$ be a $\lambda$-unifier for $A$ and $B$ with degree $\alpha$. Then, there exists an atom $A'$ such that $R(A, A') = \alpha$ and $A'\theta = B\theta$ (That is there exists $A'$ which is close to $A$, with degree $\alpha$ which unifies syntactically with $B$, through the unifier $\theta$).

Proof (Sketch). If $\theta$ is a $\lambda$-unifier for $A$ and $B$ with degree $\alpha$ then $R(A\theta, B\theta) = \alpha \geq \lambda$. Moreover, $A\theta$ and $B\theta$ share the same positions (i.e., $Pos(A\theta) = Pos(B\theta)$) and for all position $u_i \in Pos(A\theta)$, $R(A\theta[u_i], B\theta[u_i]) = \alpha_i$ being $\alpha = \Lambda_{i=1}^{n} \alpha_i$. Note that, only positions $w_i \in Pos(A) \cap Pos(B)$ contribute to the computation of the degree $\alpha$ (for the rest of positions $w'_i$, $R(A\theta[w'_i], B\theta[w'_i]) = 1$).
Now, we take \( A \) and build an atom \( A' \) of the following form: for each \( u_i \in P \succ(A) \cap P \succ(B) \), we replace the symbol of \( A \) at position \( w_i \) (i.e., \( A[w_i] \)) by the corresponding symbol of \( B \) (i.e., \( B[w_i] \)) in \( A \). That is, we build an atom \( A' \) which is exactly equal than \( A \) except that the symbols at non-variable positions (shared with \( B \)) have been replaced with symbols of \( B \). Therefore, \( R(A, A') = \alpha \) and \( A' \theta = B \theta \) by construction of \( A' \).

**Theorem 1 (Soundness of the WSLD–Resolution).** Let \( II \) be a definite program, \( R \) a similarity relation, \( \lambda \) a cut value and \( G \) a definite goal. Then every computed answer \( (\theta, \beta) \) of level \( \lambda \) for \( \{II, R\} \) and \( G \) is a correct answer of level \( \lambda \) for \( \{II, R\} \) and \( G \).

**Proof.** Assume \( G \equiv \leftarrow A_1, \ldots, A_k \) and \( \theta_1, \ldots, \theta_n \) be the sequence of w.m.g.u.’s in a WSLD-refutation of level \( \lambda \) for \( \{II, R\} \) and \( G \). \( G \vdash_{\text{WSLD}} \vdash_{\text{WSLD}} R \rightarrow \vdash_{\text{WSLD}} \square \), leading to the computed answer \( (\theta, \alpha) \) where \( \alpha = \bigwedge_{i=1}^{n} \alpha_i \) is the approximation degree computed in the refutation. We have to prove that:

1. \( I(\forall(A_1, \ldots, A_k)\theta) \) is \( \lambda \)-logical consequence of \( \{II, R\} \).
2. \( M_{II}^{\lambda}(\forall(A_1, \ldots, A_k)\theta) \leq \alpha \).

The result is proven by induction on the number of steps \( n \) of the WSLD-refutation.

**Base case** (\( n = 1 \)): First, we consider refutations of length one. This means that \( G \) must be a goal of the form \( G \equiv \leftarrow A_1 \) and the program \( II \) contains a unit clause (a fact) \( C_1 \equiv A_1 \leftarrow \) which weakly unifies with \( A_1 \). That is, there exists a w.m.g.u. \( \theta_1 \) of \( A_1 \) and \( A \) such that \( R(A \theta_1, A_1 \theta_1) = \alpha \geq \lambda \) (i.e., its approximation degree is \( \alpha_1 = \alpha \geq \lambda \)). On the other hand, it is easy to prove that \( A_1 \theta_1 \) is an instance of a clause of the annotated program \( II^\lambda \). More precisely, there exist an annotated clause \( (A', \alpha_1) \in II^\lambda \), with \( R(A, A_1) = \alpha_1 = \alpha \), such that \( A' \leq A_1 \theta_1 \). Therefore, by Corollary 2, \( \forall(A_1 \theta_1) \) is \( \lambda \)-logical consequence for \( \{II, R\} \) and item 1 is proved.

For proving item 2, remember that if \( R(A, A') = \alpha \), by definition of the immediate consequences operator of level \( \lambda \) (Definition 5), \( \alpha \in PT_{II}^{\lambda}(M_{II}^{\lambda}(A' \theta')) \) for every assignment \( \theta' \). So, for the least Herbrand model \( M_{II}^{\lambda}(A' \theta') \leq \alpha \) whatever be the assignment \( \theta' \). On the other hand, if \( A' \leq A_1 \theta_1 \), then there exists \( \gamma \) such that \( A_1 \theta_1 = A' \gamma \). Hence, for every \( \vartheta_1 \), \( M_{II}^{\lambda}(A_1 \theta_1 \vartheta_1) = M_{II}^{\lambda}(A' \gamma \vartheta_1) \leq \alpha \). Therefore, \( M_{II}^{\lambda}(\forall(A_1 \theta_1)) = \inf\{M_{II}^{\lambda}(A_1 \theta_1 \vartheta_1) | \vartheta_1 \text{ assignment} \} \leq \alpha \).

**Inductive case** (\( n > 1 \)): Next, assume that the theorem holds for WSLD-refutations with \( n > 1 \) steps. Also suppose that \( C \equiv A \leftarrow Q \in II \) is the input clause and \( A_i \) is the selected atom of \( G \) in the first step of the WSLD-refutation:

\[
\leftarrow A_1, \ldots, A_k \overset{\theta_1}{\rightarrow}_{\text{WSLD}} (A_1, \ldots, Q, \ldots, A_k) \theta_1 \Rightarrow_{\text{WSLD}} \square
\]

Now, by the induction hypothesis, we suppose that the result fulfills for computed answers of refutations with length less than \( n \). Then:

\( (a) \) \( \forall(A_1, \ldots, Q, \ldots, A_k)\theta_1, \theta_2, \ldots, \theta_n \) is \( \lambda \)-logical consequence of \( \{II, R\} \)
(b) $\mathcal{M}_H^\lambda(\forall(A_1, \ldots, Q_1, \ldots, A_k)\theta_1 \theta_2 \ldots \theta_n) \leq \beta = \bigwedge_{i=2}^{n} \alpha_i$

For proving item 1, first note that, from (a), we can immediately infer that $\forall(Q\theta_1 \theta_2, \ldots, \theta_n)$ is $\lambda$-logical consequence of $\{\Pi, R\}$. On the other hand, if the first resolution step, with the clause $C$ and the selected atom $A_1$, is possible, there must exist a wmgu $\theta_1$ with $R(A_1 \theta_1 \theta_2) = \alpha_1 \geq \lambda$. In a similar way to the base case, we can claim that there exists an annotated clause $(\Lambda' \leftarrow Q, \alpha_1) \in \Pi^\lambda$ with $R(A, \Lambda') = \alpha_1$ and such that $\Lambda' \leq A_1 \theta_1$. Therefore, by the Corollary 1, $\forall(A_1 \theta_1)$ is a $\lambda$-logical consequence of $\{\Pi, R\}$. Moreover, by Lemma 2, all of the instances of $A_1 \theta$ are $\lambda$-logical consequences of $\{\Pi, R\}$. Notably, $\forall(A_1 \theta_1, \theta_2, \ldots, \theta_n)$ is a $\lambda$-logical consequence of $\{\Pi, R\}$. Finally, it is immediate to prove that $\forall((A_1, \ldots, A_k) \theta_1, \theta_2, \ldots, \theta_n)$ is a $\lambda$-logical consequence of $\{\Pi, R\}$.

For proving the item 2, note that, since the selected atom $A_1$ on the head $A$ of the input clause $C$ weakly unify with approximation degree $\alpha_1$, by Lemma 3, there exists an atom $A'$ such that $R(A, A') = \alpha_1$ and $A' \theta_1 \leq A_1 \theta_1$. Now, we can build the ground instance:

$$A_1 \theta_1 \theta_2 \ldots \theta_n \theta_1 = A' \theta_1 \theta_2 \ldots \theta_n \theta_1$$

being the head of an instance of an annotated clause $(C' \equiv A' \leftarrow Q, \alpha_1)$ belonging to the proximity class of $C$, $K\lambda(C)$ (see Definition 5). Now by definition of the immediate consequences operator of level $\lambda$:

$$PT^\lambda_H(\mathcal{M}_H^\lambda) \supseteq \alpha_1 \wedge \mathcal{M}_H^\lambda((Q \theta_1 \ldots \theta_n) \theta_1)$$

Consequently, for every assignment $\vartheta_1$:

$$\mathcal{M}_H^\lambda(A_1 \theta_1 \ldots \theta_n \vartheta_1) \leq \alpha_1 \wedge \mathcal{M}_H^\lambda(Q \theta_1 \ldots \theta_n \vartheta_1)$$

So,

$$\mathcal{M}_H^\lambda(\forall(A_1 \theta_1 \ldots \theta_n)) = \inf\{\mathcal{M}_H^\lambda(A_1 \theta_1 \ldots \theta_n, \vartheta) \mid \vartheta \text{ assignment} \} \leq \alpha_1 \wedge \inf\{\mathcal{M}_H^\lambda(Q \theta_1 \ldots \theta_n, \vartheta) \mid \vartheta \text{ assignment} \} = \alpha_1 \wedge \mathcal{M}_H^\lambda(\forall(Q \theta_1 \ldots \theta_n)).$$

Now, by the distributivity of the universal quantifier with respect to the conjunction:

$$\mathcal{M}_H^\lambda(\forall(A_1, \ldots, A_i, \ldots, A_k) \theta_1 \ldots \theta_n) = \bigwedge_{i=1}^{l-1} \mathcal{M}_H^\lambda(\forall(A_1 \theta_1 \ldots \theta_n, \alpha_1) \wedge \mathcal{M}_H^\lambda(\forall(A_i \theta_1 \ldots \theta_n, \alpha_1) \wedge \mathcal{M}_H^\lambda(\forall(A_k \theta_1 \ldots \theta_n, \alpha_1) = \alpha_1 \wedge \bigwedge_{i=1}^{n} \alpha_i = \alpha$$

6 Conclusions and Future Work

In this paper we revisited the declarative semantics of Bousi~Prolog which were defined for a pure subset and presented in [6]. We have given more accurate definitions for the semantic concepts and thereby solved some problems that may arise when we work with non-linear programs. Moreover, we introduce for
the first a notion of correct answer inside our framework. Then, after recalling both the WSLD-resolution principle and a similarity-based unification algorithm, which is the basis of the Bousi-~Prolog operational mechanism for definite programs, we prove the soundness of WSLD-resolution as well as other auxiliary results. Finally, it is worthy to say that, along this paper we have clarified some of the existing differences between our framework and the related proposal introduced by [16].

As a matter of future work we want to continue proving the completeness theorem for this restricted subset of Bousi-~Prolog. On the other hand, at the present time we know that a naive extension of Sessa’s unification algorithm to proximity relations does not work, because correctness problems may arise. Therefore, it is necessary to define a complete new algorithm able to deal with proximity relations and to lift some of the current results to the new framework.

References