A Theoretical Basis for Intensional Logic Programming

Mehmet A. Orgun, William W. Wadge

Department of Computer Science
University of Victoria
P.O. Box 1700, Victoria, B.C.,
CANADA V8W 2Y2

e-mail: morgun@uvunix.bitnet

Abstract

Intensional Logic Programming (ILP) is a new form of logic programming based on intensional logic. The denotations of formulas of an intensional first order language are given according to intensional interpretations and to a set of possible worlds. ILP provides users with several intensional operators which are used to express relationships between different worlds.

After introducing the formal syntax and semantics of the underlying intensional logic, we investigate the general properties of intensional operators which will be used to impose certain constraints on intensional logic programming systems. Then the model-theoretic semantics of ILP are developed in conjunction with the notions of intensional Herbrand interpretations and minimum models of intensional logic programs. We will show that in particular our results apply to a temporal logic programming language called Chronolog in which the set of possible worlds is the collection of moments in time.

1. Introduction

Intensional Logic Programming (ILP) is a new form of logic programming based on intensional logic. Intensional logic [HC68] allows us to describe context-dependent properties of certain problems in a natural and problem-oriented way. In intensional logic, the values of formulas depend on an implicit context parameter. Therefore a formula may be true at some context, but false at another as relationships between elements in each context differ. The set of contexts over which the values of formulas vary is also known as the universe or the set of possible worlds. The values from different contexts can be combined through the use of intensional operators that serve as context changing operations.

This paper in particular discusses an intensional logic programming language. The underlying intensional logic incorporates as the set of possible worlds a set of triples, each of which represent the location of a world in terms of two spatial coordinates and a time coordinate, and six intensional operators,
all of which are applied to formulas of the language. Two of these operators are called "temporal" in the sense that they only affect the time coordinate of a world; the rest of the operators are called "spatial", since they operate on the spatial coordinates. All formulas in intensional logic programs are universally quantified intensional Horn clauses, similar to those in Prolog programs [Llo84].

In the following sections, we first introduce intensional Horn logic programs and explain the meaning of a program informally. The syntax and semantics of the underlying intensional logic will be described next. We will then proceed to investigate the properties of intensional operators such as monotonicity, conjunction and finiteness, all which have been defined in [OW88] for the generalisation of temporal logic programming [Wad86], [AM87]. These properties will be used to impose certain constraints on intensional operators that can be made available in the language. Later in the paper we will develop the model-theoretic semantics of intensional Horn logic programs based on intensional Herbrand models in the style of [VK76] and [Llo84]. However, we will take a different approach than that of [OW88] where the model-theoretic semantics of the temporal logic programming language called Chronolog [Wad86] were established, which rely on the notion of canonical temporal ground atoms. We will show that our results are valid both for the ILP language introduced in this paper and for Chronolog; in fact they are valid for any intensional logic programming language provided the language meets our restrictions.

2. Intensional Horn Logic Programs

Intensional logic deals with formulas whose meanings vary depending on an implicit context. Each context can be regarded as a world in which the relationships between the elements of the language may be different than those at other worlds. In this paper, we are particularly interested in an intensional language, which we call IL, in which the set U of possible worlds (the universe) is given as \{<x,y,z> | x \in \mathbb{Z}, y \in \mathbb{Z}, z \in \omega \}. A triple w = <x,y,z> in U is interpreted as representing the coordinates of some world; the first two elements (x,y) in w correspond to the location of the world on a plane and the last element refers to a moment in time.

In the underlying intensional logic, the intensional operators first and next refer to the initial and the next moment in time respectively at some spatial point. These two operators are analogous to those of Chronolog [Wad86] and the dataflow language Lucid [AW76] in a slightly different way. The semantics of these two operators are informally defined as follows: A formula of the form first A is true at world w in some intensional interpretation I iff A is true at world v where w = <x,y,z> and v = <x,y,0>. In other words, first always takes us to the initial moment in time at the same spatial point. A
formula of the form next A is true at world w in some intensional interpretation I iff A is true at world v where w = <x,y,z> and v = <x,y,z+1>. The operator next takes us to the next moment in time at the same spatial point. Since we have interpreted the third element in any w = <x,y,z> in U as the time reference, we may call first and next as temporal operators to clarify our intention.

There are four more intensional operators in the language, north, south, east and west. We will give the semantics of north; the semantics of the rest of these intensional operators can be understood without any difficulty. A formula of the form north A is true at world w in some intensional interpretation I iff A is true at world v where w = <x,y,z> and v = <x,y+1,z>. In short, north simply refers to the immediate neighbour of any point in space at the same moment in time, located one step to the north. Similarly, we call north, south, east and west as spatial operators.

Now we will give an example of intensional logic programming to illustrate how an intensional logic program works. We will adopt a Cprolog-like syntax in which the upper-case letters and the lower-case letters will be used for variables and for predicates and functions respectively. Perhaps Conway's game of life is one of the best examples which include relative references to the neighbours of a point in space at different moments in time. The game involves a (possibly infinite) plane divided into grids. Inside each grid (or cell) resides an organism that may become alive or dead depending on the status of its immediate neighbours in the surrounding cells on the plane. The game starts with an initial configuration on the plane in which some of the organisms are alive.

Supposing the initial configuration is defined elsewhere, the following program describes all relationships and state changes in the game.

```
next organism(alive) ← neighbour-list(L), count-alive(L, s(s(0)))
next organism(alive) ← organism(alive), neighbour-list(L), count-alive(L, s(s(0))))
next organism(dead) ← neighbour-list(L), lonely(L)
next organism(dead) ← neighbour-list(L), overcrowded(L)
neighbour-list([X1,X2,X3,X4,X5,X6,X7,X8]) ←
  north west organism(X1), north organism(X2),
  north east organism(X3), east organism(X4),
  south east organism(X5), south organism(X6),
  south west organism(X7), west organism(X8)
lonely(L) ← count-alive(L,X), less-than(X,s(s(0)))
overcrowded(L) ← count-alive(L,X), less-than(s(s(0)),X)
count-alive([],0) ←
count-alive([alive[L]], s(X)) ← count-alive(L,X)
count-alive([dead[L]], X) ← count-alive(L,X)
less-than(0,s(Y)) ←
less-than(s(X), s(Y)) ← less-than(X,Y)
```

We will briefly explain what the first five clauses mean. Notice that, in these clauses, some intensional operators have been used. The first clause says that an organism will become alive at the next moment
if exactly two of its neighbours are alive at the current moment no matter it is alive or dead. This clause also covers the case that a birth of an organism will occur at the next moment if it is dead and exactly two of its neighbours are alive at the current moment. The second clause says that an alive organism will continue to live at the next moment if exactly three of its neighbours are alive at the current moment. The next two clauses state that an organism will become dead at the next moment if it is lonely (less than two neighbours are alive) or the surrounding area is overcrowded (more than three neighbours are alive). The fifth clause simply bundles up the status of the neighbours of a given cell in a list for further use. The rest of the clauses define some auxiliary predicates.

Let us imagine an initial configuration (the status of the whole plane at time 0) according to which only the cells whose coordinates on the plane are given as (0,1), (1,0), (1,2) and (2,1) contain alive organisms. Put it another way, the atom organism(alive) represents a true statement at worlds <0,1,0>, <1,0,0>, <1,2,0>, and <2,1,0> and a false one at any other world with time coordinate 0. The atom organism(dead) represents a true statement at any world with time coordinate 0 where organism(alive) is false. Then at the next moment, i.e., 1, a birth will occur in cell (0,0), since exactly two of its neighbours contain alive organisms. In other words, the atom organism(alive) is true at world <0,0,1>. The rest of the progress of the game can be followed easily. According to the program, note that at any world exactly one of the atoms organism(alive) and organism(dead) is true.

We are interested in those intensional interpretations of the program in which all clauses represent true statements at all worlds in U. Now suppose the query first next organism(alive) has been asked. Since we can not put our finger on any of the cells and satisfy the query at that cell, the answer in fact involves an attempt to satisfy it at each cell. In other words, we want to enumerate all cells with alive organisms. Let us start with the cell (0,0). Given the semantics of first, next organism(alive) must be satisfied at cell (0,0) at time 0, i.e., world <0,0,0>. Now the semantics of next tells us to satisfy organism(alive) at cell (0,0) at time 1, i.e., world <0,0,1>. According to the first clause in the program, the predicate organism succeeds with <alive> at world <0,0,1>. So that the query is a logical consequence of the program at that world. Next we might consider the immediate neighbours of the cell (0,0) and satisfy the query at those worlds whose the first two elements are (x,y) where (x,y) is the spatial coordinate of a neighbour of the cell (0,0) and so forth.

If we have in the language an additional intensional operator, say center, to refer to the central point (0,0) on the plane, we could indeed reach any world by applying center together with first followed by some combinations of other operators. For instance, the sequence of operators first next center north north would lead us to the world <0,2,1>. Then the answer to the query first next center north north organism(X) is the ground instance first next center north north organism(alive) or an answer
substitution replacing X by the term alive. Obviously, center is the spatial counterpart of first. In fact, many other interesting intensional operators can be added to the language if they have certain properties. We will return to this issue later in the following sections.

3. The Syntax and Semantics of Intensional Horn Logic Programs

We begin with a standard FOL and extend it with six intensional operators first, next, north, south, east and west. We add six new formation rules: if A is a formula, so are first A, next A, north A, south A, east A and west A. Note that these intensional operators are applied to formulas, not to terms of the language. We call the language in which intensional Horn logic programs are written as IL. The universe U is the set \{ <x,y,z> | x \in \mathbb{Z}, y \in \mathbb{Z}, z \in \omega \}. We now introduce an important class of clauses called intensional Horn clauses that provide the syntactic basis for intensional logic programming.

Definition. An intensional literal is an atomic formula with a number (possibly zero) of intensional operators applied to it.

The following are intensional literals of IL.
- p(0,s(X))
- first north north p(X,a)
- west p(X,s(Y),Z)

Definition. (Intensional Horn clauses.)

(a) A definite intensional clause is the universal closure of a clause of the form
   \[ A \leftarrow B_1, \ldots, B_n \ (n \geq 0) \] where each \( B_i \) and \( A \) are intensional literals.

(b) An intensional goal clause is the universal closure of a clause of the form
   \[ \leftarrow B_1, \ldots, B_n \ (n > 0) \] where each \( B_i \) is an intensional literal.

(c) An intensional Horn clause is either an intensional goal clause or a definite intensional clause.

Definition. An intensional Horn logic program is a finite set of definite intensional clauses.

We understand an intensional logic program \( P \) in terms of intensional interpretations by the following: \( P \) is true in an intensional interpretation \( I \) iff all clauses in \( P \) are true in \( I \). A clause in \( P \) is true in \( I \) iff all of its ground instances are true in \( I \) at all worlds \( w \in U \) (here ground instance of a clause means it does not contain any uninstantiated variable.) An intensional interpretation \( I \) basically attaches meanings to all elements of the language.

37
Definition. An intensional interpretation \( I \) for IL comprises a non-empty set \( D \), called the domain of the interpretation, over which the variables range, together with for each variable, an element of \( D \); for each n-ary function symbol, an element of \( D^n \rightarrow D \); and for each n-ary predicate symbol, an element of \( U \rightarrow \mathcal{P}(D^n) \). (Here \( \mathcal{P}(X) \) is the set of all subsets of \( X \).)

The fact that a formula \( A \) is true at world \( w \) with respect to an intensional interpretation \( I \) will be denoted as \( \models_{I,w} A \). Note that all formulas of IL are intensional, that is, their meanings may vary depending on the elements of \( U \). But here the denotations of variables and function symbols are extensional, that is, independent of the elements of \( U \). Let \( \|E\|_I \) denote the value \( I \) gives an IL term \( E \).

Definition. The semantics of formulas of IL are given inductively by the following, where \( I \) is an intensional interpretation for IL, \( w \in U \), and \( A \) and \( B \) are formulas of IL.

(a) If \( f(e_1,\ldots,e_n) \) is a term, then \( \|f(e_1,\ldots,e_n)\|_I = I(f)(\|e_1\|_I,\ldots,\|e_n\|_I) \in D \). If \( v \) is a variable, then \( \|v\|_I = I(v) \in D \).

(b) For any n-ary predicate \( P \) and terms \( e_1,\ldots,e_n \), \( \models_{I,w} P(e_1,\ldots,e_n) \) iff \( \langle \|e_1\|_I,\ldots,\|e_n\|_I \rangle \in I(P)(w) \).

(c) \( \models_{I,w} \neg A \iff \not \models_{I,w} A \).

(d) \( \models_{I,w} A \land B \iff \models_{I,w} A \land \models_{I,w} B \).

(e) \( \models_{I,w} \forall x A \iff \models_{I,d,x,w} A \) for all \( d \in D \).

(f) \( \models_{I,<x,y>} \) first \( A \) iff \( \models_{I,x,y>0} A \)

\( \models_{I,<x,y>} \) next \( A \) iff \( \models_{I,x,y+1>0} A \)

(g) \( \models_{I,<x,y>} \) north \( A \) iff \( \models_{I,x,y+1>0} A \)

\( \models_{I,<x,y>} \) south \( A \) iff \( \models_{I,x,y-1>0} A \)

\( \models_{I,<x,y>} \) east \( A \) iff \( \models_{I,x-1,y>0} A \)

\( \models_{I,<x,y>} \) west \( A \) iff \( \models_{I,x-1,y>0} A \)

Again, note that the denotation of a term does not vary with time. Only the truth values of formulas change with time. The above definitions (except (f) and (g)) apply to arbitrary intensional languages since they do not depend on the structure of the elements of \( U \). But, to define the semantics of intensional operators, we are forced to consider the structure of elements of \( U \). Note that the semantics of intensional languages would differ depending on what kind of intensional operators they include.
4. Intensional Operators

In this section, we will investigate how to extend the language with additional intensional operators along the lines of [Sco70] and [OW88]. To simplify the presentation, only unary (intensional) operators will be considered. We can enrich IL with an extra unary intensional operator \( \forall \) as follows. Take \( \| \forall \| \) to be any given element \( \Theta \) of \( \mathcal{P}(U) \rightarrow \mathcal{P}(U) \), and add the following clause to the definition of \( \models \) where \( I \) is an intensional interpretation of IL and \( w \in U \):

\[
\models_{I, w} \forall A \text{ iff } w \in \Theta(\|A\|)
\]

Here \( \|A\| = \{ w \mid I, w \models A \} \). Note that \( \|A\| \subseteq U \) can also be viewed as an element of \( U \rightarrow 2 \), in which case it is called an "intension" which, given any \( w \in U \), returns the truth value (extension) of \( A \) at \( w \). Here \( 2 = \{ 0, 1 \} \); 0 and 1 represent false and true respectively.

The characterisation of unary intensional operators given above is in fact a smooth reformulation of the semantics given by the definition of \( \models \). But it provides a more general framework for our presentation. For example, if \( \Theta \) is the function defined below,

\[
\Theta = \lambda X. \{ <x,y,z> \in U \mid <x,y,0> \in X \}
\]

then \( \forall \) is just the temporal operator first. But if we choose \( \Theta \) so that

\[
\Theta = \lambda X. \{ <x,y,z> \in U \mid <x,y,t> \in X \text{ for some } t > z \}
\]

then \( \forall \) is the "sometime in the future" operator. When \( \Theta \) is the function defined below, it is not hard to see that \( \forall \) is next followed by south, which is not really a new operator.

\[
\Theta = \lambda X. \{ <x,y,z> \in U \mid <x,y,-1,z+1> \in X \}
\]

In fact, the denotations of such combined operators as next south can be defined as the composition of the denotations of the operators involved. For instance, the denotation of next south, i.e., \( \Theta \), is the function \( \Phi \circ \Psi \), where \( \|\text{next}|| = \Psi \) and \( \|\text{south}|| = \Phi \).

\[
\Phi \circ \Psi = \lambda X. \{ <x,y,z> \in U \mid <x,y,z+1> \in X \}
\]

Or simply : \( \Phi \circ \Psi = \lambda X. \Psi(\Phi(X)) \). It can easily be shown that here the function \( \Psi \circ \Phi \) is the same function as \( \Phi \circ \Psi \). However, this is not true in general. Consider first and next; the compositions of the denotations of these two operators yield totally different functions, because they both operate on the same (time) coordinate of a given world.

As our last example, consider the following function for which we do not have a corresponding operator that can be expressed directly by using some combination of intensional operators of IL.

\[
\Theta = \lambda X. \{ <x,y,z> \in U \mid <x,n,z> \in X \text{ for all } n \geq 1 \}
\]

Nevertheless, we can give a "recursive" definition as follows.

\[
\forall A = A \land \forall(\text{north } A)
\]
We can use this definition as a rewrite rule: any formula of the form \( \forall A \) is transformed to a formula of the form \( A \land \forall \text{north} A \) which also contains an application of \( \forall \). By successive applications of the definition, we eventually obtain \( \forall A = \bigwedge_{n \in \mathbb{N}} \text{north}^n A \). The question is whether these kinds of extensions to \( \mathbb{L} \) can be accommodated within the scope of intensional Horn logic or not. The answer surely awaits further research.

We can express the distinction between intensional and extensional operators in terms of their denotations: An operator \( \forall \) is called "extensional" iff \( \Theta \) is of the form \( \lambda X.\{ \ w \in U \mid f(b) = 1 \} \) where \( f \) is an element of \( 2^{\rightarrow 2} \) and \( b = 1 \) if \( w \in X; 0 \) otherwise. For instance, \( f = \lambda v.\overline{v} \) for negation \( (\overline{v} \) is the complement of \( v \).) In other words, for an extensional operator \( \forall \), the value of \( \forall A \) at world \( w \) only depends on the value of \( A \) at the same world. Similarly, an operator \( \forall \) is called "intensional" iff, for any \( w \in U \), the information that \( w \in X \) or not is not enough to decide if \( w \in \Theta(X) \). For an intensional operator \( \forall \), the value of \( \forall A \) at world \( w \) depends on the values of \( A \) at possibly \( w \) and at worlds other than \( w \) (a subset of the universe.)

Some operators enjoy the property that their denotations are independent of any structure of the elements of \( U \). These kind of operators can be more easily included in any intensional logic programming system regardless of the choice of \( U \) when they satisfy the following constraints.

**Definition.** We say that \( \Theta \) is *monotonic* if \( \Theta(X) \subseteq \Theta(Y) \) whenever \( X \subseteq Y \).

Our first requirement of a new intensional operator is that its denotation \( ||\forall|| \) be monotonic - hardly surprising, given the fact that Horn logic programming is based on monotonic logic. In the next section, we will discuss the consequences of monotonicity.

**Definition.** We say that \( \Theta \) is *conjunctive* iff \( \Theta(X \land Y) = \Theta(X) \land \Theta(Y) \) for all \( X \) and \( Y \).

Conjunctivity ensures the model intersection property, i.e., if \( \Theta \) is conjunctive, the intersection of a family of intensional Herbrand models of \( \forall A \) is also an intensional Herbrand model of \( \forall A \). We will elaborate this property later in the next section. The following proposition relates conjunctivity with two other properties. Let \( \Theta_w = \{ X \subseteq U \mid w \in \Theta(X) \} \).

**Proposition 1.** \( \Theta \) is conjunctive iff it is monotonic and for all \( w \in U \), \( \bigcap_{w \in \Theta} \Theta_w \).

**Proof.** It is straightforward from the definition of conjunctivity.

**Definition.** We say that \( \Theta \) is *finitary* iff for any \( X \subseteq U \) and any \( w \in U \), if \( w \in \Theta(X) \) then \( w \in \Theta(S) \) for some finite subset \( S \) of \( X \).
When $\Theta$ is conjunctive, finiteness can also be expressed as follows: $\Theta$ is finitary iff $\bigcap \Theta w$ is finite for all $w \in U$. Note that $\bigcap \Theta w$ is the minimum element in $\Theta w$ for any $w \in U$ for which $w \in \Theta (\bigcap \Theta w)$. It can be shown that here monotonic and finitary operators are continuous as well (see [Yag84] for an analogous proof given for operators of the dataflow language Lucid.)

The denotations of all the intensional operators we have introduced in the previous sections are monotonic, conjunctive and finitary. It can be verified that compositions of such intensional operators still have the desired properties (e.g., next south.) As for the classical intensional operators necessary $\square$ and possible $\lozenge$, whose denotations are independent of the structure of $U$, $\|\square\|$ is conjunctive and monotonic, but not finitary; and $\|\lozenge\|$ is finitary and monotonic, but not conjunctive. (Read $\square A$ as 'A is true at all worlds' and $\lozenge A$ as 'A is true at some world.') For example, consider the denotation of $\square : \|\square\| = \lambda X. \{ w \in U \mid X = w \}$. Clearly no $w \in U$ is in $\|\square\|(X)$ for a finite $X$. We will see later that the use of such operators in intensional Horn logic programs must be restricted.

5. Models of Intensional Logic Programs

In this section, we will develop model-theoretic semantics of intensional Horn logic programs. We will neither focus on any particular language such as the one defined earlier, nor explicitly make use of the structure of elements of $U$. Instead, constraints will be imposed on intensional operators so that all the results are valid regardless of what the language offers. We assume all operators that can be used in intensional logic programs have the properties formulated in the previous section, that is, their denotations are monotonic, conjunctive and finitary. Chronolog, which is a temporal logic programming language would enjoy all the following results, even though Orgun and Wadge [OW88] have developed an alternative semantics based upon the notion of canonical temporal ground atoms. This is because the temporal operators of Chronolog have the desired properties.

We will follow the theoretical foundations developed for the theory of logic programming [VK76, Llo84]. We first define the notion of a model of a formula of IL and extend it to a set of intensional Horn clauses.

Definition. Let $I$ be an intensional interpretation for IL and let $A$ be a formula of IL. Then $I$ is a model for $A$ iff $A$ is true for all $w \in U$. In other words, $I$ is a model for $A$ iff $\models_{I,w} A$ for all $w \in U$. We shall denote this fact by $I \models A$.

Definition. Let $P$ be an intensional Horn logic program and let $I$ be an intensional interpretation for IL. Then $I$ is a model for $P$ iff $I$ is a model for each clause in $P$, that is, $I \models P$ iff for all $C_i \in P$, $I \models C_i$. 

41
We next study intensional Herbrand interpretations. They are sufficient for the theory of intensional logic programming, since we only deal with intensional Horn logic clauses which are closed formulas of IL.

**Definition.** An intensional ground atom is a positive intensional literal not containing any variables and without any intensional operators applied to it.

It can be argued that this definition of a ground atom coincides with that of classical Horn logic. On the other hand, the value of an intensional ground atom varies in different worlds in an intensional interpretation, whereas the value of a ground atom is simply true or false in an interpretation.

As far as Chronolog is concerned, we have the notion of canonical temporal ground atoms, for which each atom refers to a particular moment in time, so that its value does not vary. Notice that the intensional programming language given in the preceding sections would have the notion of canonical intensional ground atoms providing the language included the spatial operator center as well as first. But here we are developing the model-theoretic semantics which are suitable not only for a particular language, but also for those that may lack the notion of canonical (intensional) ground atoms.

**Definition.** Let $P$ be an intensional Horn logic program. The intensional Herbrand universe $U_{IP}$ of $P$ is the set of all ground terms which can be constructed out of constants and functions that appear in $P$.

**Definition.** Let $P$ be an intensional Horn logic program. The intensional Herbrand base $B_{IP}$ of $P$ is the set of all intensional ground atoms which can be constructed out of predicates that appear in $P$ with ground terms in $U_{IP}$ as arguments.

All intensional ground atoms in $B_{IP}$ can be considered as propositional objects whose meanings vary depending on the elements of $U$. As the meanings of functions are fixed in Herbrand interpretations and the domain of the interpretation is the Herbrand universe $U_{IP}$, we can identify an intensional Herbrand interpretation $I$ of $P$ with a function $H$ which assigns to each intensional ground atom $P(e_1, \ldots, e_n) \in B_{IP}$ an element of $U \rightarrow 2$ by the following.

\[
\langle e_1, \ldots, e_n \rangle \in I(P(w)) \text{ iff } H(P(e_1, \ldots, e_n))(w) = 1
\]

We say that $H$ is a model of $P$ iff $I \models P$ for any $I$ corresponding to $H$. From here on, these two dual notions of an intensional Herbrand interpretation will be used interchangeably. Recall that $||A||^I$ denotes the function (intension) assigned to a ground atom $A$ by an intensional Herbrand interpretation $I$. In addition, $||A||^I$ can be thought of as a subset of $U$ such that any $w \in U$ is in the set provided $I \models w \in A$. 

42
Given the semantics of formulas of IL, an intension \( \|A\|^I \) naturally extends to all formulas of IL. However, we are most interested in those formulas of IL of the form \( \forall A \) where \( \forall \) is a sequence of intensional operators and \( A \in B_P \) for some program \( P \). We also assume that the semantics of intensional operators are embedded in \( H \), i.e., the function which corresponds to \( I \). For example, the intension \( \|\text{first } A\|^I \) (or \( \|\text{first } A\|^H \)) is equal to \( \|\text{first}\|\|A\|^I \) (or \( \|\text{first}\|\|A\|^H \)).

Since intensional Herbrand interpretations can not be treated as subsets of \( B_P \), we must define three operations on intensional Herbrand interpretations which are analogous to set inclusion, intersection and union.

**Definition.** Let \( M = \{M_\alpha\}_{\alpha \in S} \) be a family of intensional Herbrand interpretations of an intensional logic program \( P \).

(a) An ordering relation on intensional Herbrand interpretations is defined as follows. \( I \sqsubseteq J \) iff \( \|A\|^I \subseteq \|A\|^J \) for all \( A \in B_P \) where \( I \) and \( J \) are intensional Herbrand interpretations as defined above.

(b) \( \sqcap \)-intersection : \( \sqcap M = \sqcap_\alpha M_\alpha \) is an intensional Herbrand interpretation for \( P \) that assigns to each \( A \in B_P \) a subset of \( U \) defined as \( \|A\|^\sqcap M = \bigcap_\alpha \|A\|^M_\alpha \).

(c) \( \sqcup \)-union : \( \sqcup M = \sqcup_\alpha M_\alpha \) is an intensional Herbrand interpretation for \( P \) that assigns to each \( A \in B_P \) a subset of \( U \) defined as \( \|A\|^\sqcup M = \bigcup_\alpha \|A\|^M_\alpha \).

The ordering relation \( \sqsubseteq \) defines a partial order on the intensional Herbrand interpretations of a given intensional Horn logic program \( P \), as does the set inclusion on Herbrand interpretations of a given Horn logic program. In fact, the family of all intensional Herbrand interpretations of \( P \) is a complete lattice under the partial order of \( \sqsubseteq \). The topmost element of the lattice is the intensional Herbrand interpretation \( H_P \) defined by the following : for all \( A \in B_P \), \( \|A\|^H_P = U \). The bottommost element \( \mathcal{H}_\emptyset \) of the lattice can be defined in a similar way : for all \( A \in B_P \), \( \|A\|^H_\emptyset = \emptyset \). The least upper bound (lub) of any family \( M \) of intensional Herbrand interpretations is \( \sqcup M \); and the greatest lower bound (glb) is \( \sqcap M \).

The following two propositions justify the claim that intensional Herbrand interpretations are sufficient for proving the unsatisfiability of a set of intensional Horn clauses. Note that we are not interested in arbitrary sets of intensional Horn clauses (see proposition 3.)

**Proposition 2.** Let \( S \) be a set of intensional Horn clauses and suppose \( S \) has an intensional model. Then \( S \) has an intensional Herbrand model.
Proof. Let \( I \) be an intensional interpretation of \( S \). Then the corresponding intensional Herbrand interpretation \( I_H \) can be defined as follows: The domain \( D \) of \( I_H \) is \( U_{IS} \) and for all \( P(e_1, \ldots, e_n) \) in \( B_{IP} \), \(<e_1, \ldots, e_n> \in I_H(P)(w) \) whenever \(<\|e_1\|, \ldots, \|e_n\|> \in I(P)(w)\). Clearly if \( I \) is a model of \( S \), so is \( I_H \). ■

Proposition 3. Let \( P \) be an intensional Horn logic program and \( \forall A \) be an intensional literal where \( A \in B_{IP} \) and \( \forall \) is a sequence of intensional operators. Then \( P \cup \{ \neg \forall A \} \) is unsatisfiable in any intensional model of \( P \) iff no intensional Herbrand model of \( P \) satisfies \( P \cup \{ \neg \forall A \} \).

Proof. If \( P \cup \{ \neg \forall A \} \) is satisfiable in some intensional model \( I \) of \( P \), then \( I \models_{I_w} P \cup \{ \neg \forall A \} \) for some \( w \in U \). By proposition 2, a construction from \( I \) to the corresponding intensional Herbrand model \( H \) can be given. ■

We would like to clarify the notion of unsatisfiability formulated above. Since we are interested in what can be proved from an intensional logic program \( P \), we must consider only those intensional interpretations of \( P \) which are models as well. There are certainly many other interpretations of \( P \) with respect to which \( P \cup \{ \neg \forall A \} \) may be true at some \( w \), that is \( P \cup \{ \neg \forall A \} \) is "satisfiable." Evidently satisfiability alone does not imply the existence of models.

We next show that every intensional Horn logic program has a model which is the topmost element in the lattice of the family of intensional Herbrand interpretations, therefore the set of models of any program is non-empty. The following proposition can not be proved if we drop the restriction that the denotations of operators be monotonic. One perfect example to operators with their denotations non-monotonic is of course the negation. (For any \( X \) and \( Y \subseteq U \), \( X \subseteq Y \) implies \( \| \neg \| (X) \supseteq \| \neg \| (Y) \), where \( \| \neg \| = \lambda X. U \setminus X \).

Proposition 4. Let \( H_{IP} \) be an intensional Herbrand interpretation of an intensional Horn logic program \( P \) such that for all \( A \in B_{IP} \), \( \| A \|_{H_P} = U \). Then \( H_{IP} \) is a model of \( P \).

Proof. Clearly \( H_{IP} \) is an intensional Herbrand interpretation of \( P \). Consider any ground instance of any clause in \( P \) and let \( A \leftarrow B_1, \ldots, B_n \) be one such ground instance. Each intensional literal in the instance is of the form \( "VF" \), where \( F \) is an intensional ground atom and \( V \) is a sequence of intensional operators with \( \| V \| = \Theta \). For any \( w \in U \), pick an element \( X \) of \( \Theta_w \) such that \( w \in \Theta(X) \). Since \( X \subseteq U \) and \( w \in \Theta(X) \), we have that \( w \in \Theta(U) \) by monotonicity. We also know that for any \( F \in B_{IP} \), \( \| F \|_{H_P} = U \) by construction, so \( w \in \| VF \|_{H_P} \) iff \( w \in \Theta(\| F \|_{H_P}) \) iff \( w \in \Theta(U) \). That implies all intensional literals in the ground instance must be true at \( w \). Therefore \( I_{H_P} \models A \leftarrow B_1, \ldots, B_n \) for any ground instance of any clause in \( P \). ■
We now relate conjunctivity with model intersection property. The following proposition shows that the $\bigcap$-intersection of a family of intensional Herbrand models of some formula of the form $\forall A$ is also a model, providing $\|\forall\|=\Theta$ is conjunctive. Recall that $\Theta_w = \{ X \subseteq U \mid w \in \Theta(X) \}$.

Proposition 5. Let $\forall$ be an operator with $\|\forall\|=\Theta$ conjunctive and $A \in B_I$. Let $M_{\forall A} = \{ M_\alpha \mid \forall \alpha \forall A \}$ be the family of intensional Herbrand models of $\forall A$. Then $\bigcap M_{\forall A} \subseteq M_{\forall A}$. Moreover $\|A\|\bigcap M_{\forall A}$ is equal to $\bigcup_{w \in U} \bigcap \Theta_w$.

Proof. We will outline the proof. Since $\Theta$ is conjunctive, for all $w \in U$, $\bigcap \Theta_w \subseteq \Theta_w$ by proposition 1. Given that $\|A\|^{M_\alpha} \subseteq \Theta_w$ for any $w \in U$ and for any $M_\alpha \in M_{\forall A}$, it can be shown that $\|A\|\bigcap M_{\forall A} = \bigcap_{\alpha} \|A\|^{M_\alpha} \subseteq \Theta_w$, which in turn implies $\bigcap M_{\forall A}$ is a model.

As regards the second part of the proposition, clearly $\bigcap_{\alpha} \|A\|^{M_\alpha} \supseteq \bigcup_{w \in U} \bigcap \Theta_w$. In addition, one of the models of $\forall A$ must assign $\bigcup_{w \in U} \bigcap \Theta_w$ to $A$. Thus we have the set inclusion in the reverse direction as well. ■

As a corollary, the first part of proposition 5 can be extended to arbitrary families of models of any formula of the form $\forall A$. We next show that model intersection property smoothly extends to a family of intensional Herbrand models of an intensional Horn logic program.

Proposition 6. (Model $\bigcap$-intersection property.) Let $P$ be an intensional Horn logic program and $M = \{ M_\alpha \}_{\alpha \in S}$ be a non-empty family of intensional Herbrand models for $P$. Then $\bigcap M$ is an intensional Herbrand model for $P$.

Proof. Clearly $\bigcap M$ is an intensional Herbrand interpretation of $P$. Suppose $\bigcap M$ is not a model of $P$. Then there is a ground instance of an intensional clause in $P$ of the form $A \leftarrow B_1, \ldots, B_n$ which is false in $\bigcap M$ at some $w \in U$. That means all $B_i$'s are true, but $A$ is false in $\bigcap M$ at $w$, which in turn implies $A$ is false in some $M_\alpha$ at $w$. But we know that $A$ is true at $w$ in all $M_\alpha \in M$ by definition. Therefore $A$ is true in $\bigcap M$ at $w$ by proposition 5. ■

The following theorem states that there is a model of an intensional Horn logic program called the minimum intensional Herbrand model, which as far as declarative semantics is concerned, is all we need to know about the program.

Theorem 7. Every intensional Horn logic program $P$ has a $\bigcap$-minimum intensional Herbrand model $M_{IP}$, which is the $\bigcap$-intersection of all intensional Herbrand models of $P$. ■

45
Proof. Let $M = \{ M_\alpha | M_\alpha \models P \}$ be the family of all intensional Herbrand models of $P$. Then $M$ is non-empty by proposition 4. Therefore the $\bigcap M = M_p$ is the minimum model of $P$ by proposition 6. \qquad \square

The theorem given below states that an intensional literal $\forall A$ is a logical consequence of an intensional Horn logic program $P$ iff $w \vDash \forall A | M_p^w$ and $\models P$ at any world $w$. (Here $A$ is an intensional ground atom in $B_p$ and $\forall$ is a sequence of intensional operators.) It is a stronger characterisation of the minimum Herbrand model $M_p$.

**Theorem 8.** Let $P$ be an intensional Horn logic program. Then $\forall A$ with $A \in B_p$ is a logical consequence of $P$ iff $P \models \forall A$.

**Proof.** $\forall A$ is logical consequence of $P$ iff

$P \cup \{ \neg \forall A \}$ is unsatisfiable in any intensional model of $P$ iff

no intensional Herbrand model of $P$ satisfies $P \cup \{ \neg \forall A \}$ by proposition 3 iff

$P \not\models \forall A$ for all models $M_\alpha$ of $P$ iff

$P \models \forall A$ for all models $M_\alpha$ of $P$ iff

$P \models \forall A$. \qquad \square

The fixpoint theory of Horn logic programs can be modified for further extensions of the theory of ILP as well. We will give the outline. The continuous mapping originally given in [VK76] provides the basis for fixpoint semantics and therefore establishes the connection between the model-theoretic and operational semantics of Horn logic programs. $T_p$ which is one step modus ponens function for intensional Horn logic is defined as follows: Let $P$ be an intensional Horn logic program and $H$ be an intensional Herbrand interpretation of $P$. Then $T_p(H)$ is an intensional Herbrand interpretation given by

$w \vDash A_l | T_p(H)$ iff $w \vDash B_i | H$ for all $1 \leq i \leq n$

where $A_l \leftarrow B_1, \ldots, B_n$ is a ground instance of an intensional clause in $P$.

Let us reflect. Each $B_i$ (and $A$ as well) is possibly of the form $\forall A_j$ where $A_j \in B_p$ and $\forall$ is a sequence of intensional operators (not necessarily the same sequence for different $B_i$'s.) Then $T_p(H)$ actually reads as follows:

$w \vDash A_l | T_p(H)$ iff $w \vDash B_i | H$ for all $1 \leq i \leq n$

iff $w \vDash \forall A_j | H$ for all $1 \leq j \leq n$

iff $w \vDash \forall \{B_i | H\}$ for all $1 \leq i \leq n$

When $A$ is of the form $\forall F$ for some $F \in B_p$, we must explain what $T_p(H)$ assigns to $F$. Let $\Theta = \| \forall \|$ and $S = \{ w \in U | w \vDash B_i | H$ for all $1 \leq i \leq n \}$. Then $\|F\| | T_p(H) = \bigcup_{w \in S} \Theta_w$, so that $w \vDash A_l | T_p(H)$ for all $w \in S$. 46
If $F$ appears on the left-hand side of any other ground instances, the intension assigned to $F$ is the union of all the intensions induced by each ground instance, which is assumed by the definition of $T_F$. Now it can be shown that $\text{f.p}(T_F) = T_F \uparrow \omega = M_F$ provided that all intensional operators have the properties.

For example, consider the operator necessary $\Box$ whose denotation is non-finitary. Suppose we used $\Box$ in the naive intensional Horn logic program given as: $P = \{ p(a) \leftarrow \Box q(a), q(X) \leftarrow \}$. Since $||\Box||$ is non-finitary, it can be shown that $T_P$ is no longer continuous (we omit the details.) Then $P \models p(a)$ but $M_P$ is not computable: given $||\Box||=\lambda X.\{ \text{we} \in U \mid X=U \}$, any attempt to satisfy $p(a)$ at any given world will result in an endless series of attempts to satisfy $q(a)$ at all worlds, which will never converge.

We can show that the results still hold even if the denotations of all intensional operators are just monotonic and conjunctive - provided they are used only on the left-hand side of the intensional clauses in the program. Similarly, if intensional operators appear on the right-hand side, and their denotations are monotonic and finitary, then the theorems remain valid. We can also conclude that the temporal logic programming language Templog introduced by Abadi and Manna [AM87] has the minimum model property since they use $\Box$ only on the left-hand side; and $\Diamond$ only on the right-hand side (as "sometime in the future operator"). The intensional operators of the language we have introduced in this paper and those of Chronolog can be used anywhere in programs, since their denotations have all three of the properties.

6. Conclusions

The theoretical basis introduced in this paper has enabled us to develop a language-independent model theory for ILP. We have also shown that ILP languages such as Chronolog [Wad86] and Templog [AM87] as well as the one described in this paper enjoy our results, because the intensional/temporal operators they offer have the desired properties, i.e., their denotations are monotonic, conjunctive and finitary. We are also considering an extension to Chronolog which employs the set of integers as the set of possible worlds and another temporal operator pre which refers to the previous moment in time. It can be shown that $||\text{pre}||$ has all the three properties, therefore we can immediately conclude that Chronolog with negative time has the minimum model property as well.

We have not addressed any implementation techniques or rules of inferences for intensional Horn logic. Our colleagues A.A. Faustini and D. Rolston [Rol87] of Arizona State University are investigating efficient implementation techniques for Chronolog based on the combined features of a Horn logic
programming implementation and a dataflow language implementation such as that of [FW85]. As regards the language described in this paper, a display-oriented tool similar to spreadsheets can be readily incorporated in an implementation such that queries can be entered in the cells and the answers to the queries (i.e., intensional ground atoms or answer substitutions) can be displayed back in the same cells at any given moment. This approach would resemble that of [Du86] where the implementation of an intensional programming language and its application to an intensional 3-D spreadsheet are sought.

We are also exploring other extensions to intensional logic programming, including intensional Horn logic axiomatizability, i.e., what kind of intensional operators can be made available in terms of primitive ones and by rules of intensional Horn logic; and choice predicates in Chronolog as defined in [Wa86]. For example, given any moment in time, a unary choice predicate is only true of a certain ground term, but the choice of that term among all perfectly valid terms is totally arbitrary. Intensional Horn logic alone may not suffice to develop rigorous model-theoretic semantics for Chronolog with choice predicates (a logic with equality may be needed.)

References


[Du86] Du, W.; An Intensional 3-D Spreadsheet and Its Implementation, Master's Thesis, Department of Computer Science, University of Victoria, B.C., Canada, 1986.


