A Model Theory for Nonmonotonic Multiple Value and Code Inheritance in Object-Oriented Knowledge Bases

A DISSERTATION PRESENTED
BY
GUIZHEN YANG

TO
THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE
STATE UNIVERSITY OF NEW YORK
AT STONY BROOK

December 2002

State University of New York
at Stony Brook
The Graduate School

Guizhen Yang

We, the dissertation committee for the above candidate for
the degree of Doctor of Philosophy,
hereby recommend acceptance of this dissertation.

Professor Michael Kifer, Advisor
Department of Computer Science

Professor I.V. Ramakrishnan, Chairman of Defense
Department of Computer Science

Professor Yanhong Annie Liu
Department of Computer Science

Professor C.R. Ramakrishnan
Department of Computer Science

Dr. Bertram Ludäscher, Director
Knowledge-Based Information Systems Laboratory
San Diego Supercomputer Center

This dissertation is accepted by the Graduate School.

Dean of the Graduate School

Abstract of the Dissertation

A Model Theory for Nonmonotonic Multiple Value and Code Inheritance in Object-Oriented Knowledge Bases

by
Guizhen Yang

Doctor of Philosophy
in
Computer Science

State University of New York
at Stony Brook

2002

We have developed a comprehensive model theory for nonmonotonic multiple value and code inheritance in object-oriented knowledge bases. Our new inheritance semantics, called optimistic object model semantics, supports implicit inference by inheritance as well as explicit deductive inference via rules. Inference by inheritance supports a multitude of features, such as overriding, nonmonotonic multiple value and code inheritance, meta programming, and dynamic class hierarchies — the important features that are fundamental to advanced object-oriented knowledge management.

In the setting of three-valued models, we formally define the inheritance postulates that capture the common intuition behind overriding and conflict resolution in nonmonotonic multiple value and code inheritance. These postulates specify the minimum requirements for object models.

We specify an extended alternating fixpoint procedure for computing object models. We define a unique object model, called optimistic object model, for any given program that is written in our rule-based query language. We prove three different characterizations of the optimistic object model semantics: an optimistic object model is the least fixpoint of the extended alternating fixpoint computation, is the least stable object model with respect to information ordering, and is a minimal object model with respect to truth ordering.

Our new inheritance semantics yields intuitively satisfactory results in all known benchmark cases, does not impose syntactic restrictions on programs, and has been implemented in the Flora-2 system. To the best of our knowledge, the optimistic object model semantics is currently the only model-theoretic semantics for nonmonotonic multiple value and code inheritance that applies to general, unrestricted object-oriented knowledge bases.

To my parents.

Contents

List of Figures

Acknowledgements

I would like to thank my advisor, Professor Michael Kifer, for his generous support. During my study at Stony Brook, I have learned and benefited tremendously from his rigorous, persistent research style and his insight into research problems. The quality of my work reflects the many long yet enjoyable hours that we spent together, either brainstorming, nailing down technical details, or challenging each other's ideas.

My thanks also go to my co-advisor, Professor I.V. Ramakrishnan, for his being so inspiring and so enthusiastic. He has a very acute sense about research as well as humor, and always brings a lot of fun to work. It is really my privilege to have worked with him on a number of projects and research papers.

I want to extend my thanks to the other members of my dissertation committee: Professor Yanhong Annie Liu, for her helpful discussions and suggestions; Professor C.R. Ramakrishnan, who unfolded and introduced the new world of logic programming to me; and Dr. Bertram Ludäscher at San Diego Supercomputer Center, whose early work on the Flip project greatly influenced our design of the Flora-2 system.

I would not have gone this far and have accomplished this much without the support from my great family: my parents, Xiangdong and Yingxia, my sister, Guili, and my brother, Guihua. I am so fortunate to have grown up in such a loving and caring family. My mom and dad sustained unimaginable hardships and made great personal sacrifices to raise their three children. They lightened my spirit when I was lost and soothed my soul when I was down. They are the pillars of my life and my ultimate source of strength. Thank you, mom and dad!

I am very grateful to my uncle, Zhenye, and my aunt, Juan, for their hearty support. My uncle taught me a lot of precious life lessons and never ceases encouraging me to pursue my dreams. My aunt cooked very delicious Chinese dishes and made me feel so at home every time I had a visit to her family and was away from my parents. I also owe my thanks to my cousin, Guangming, for being such a great buddy who shares my joys and sorrows.

My dissertation received a lot of support from the XSB research team at Stony Brook: Professor David Warren, Dr. Terrance Swift, Dr. Baoqiu Cui, and Luís Fernando Castro, who all took great patience in answering my questions about the XSB system, on which the Flora-2 system was built.

Thanks are also due to my wonderful colleagues and co-authors, Hasan Davulcu and Saikat Mukherjee, and to the hardworking system and support staff in our department, in particular, Brian Tria, Anne Kilarjian, Betty Knittweis, and Edwina Osmanski.

I am indebted to my high school mentor, Hongbing Zhang, for her invaluable guidance in the early stage of my education and her constant encouragement in these many years.

The past few years would have been short of joy without the many friends and officemates at Stony Brook: Yifei Dong, Shiyong Lu, Xianfeng Yuan, Ying Zhang, Haifeng Guo, Jing Hua, Gang Peng, Ningning Zhu, and many others. Their friendship is very much appreciated.

Last but not the least, I would like to say my thanks especially to Lan Huang, for all the delight she brings and her company and moral support.

Chapter 1

Introduction

Object-oriented knowledge bases combine the strengths of both object-oriented and deductive programming paradigms. Recent years have witnessed a great deal of success of such systems in projects ranging from data integration in neuroscience [23] to processing semistructured and semantic information on the Web [40, 12, 22, 51] to information mediation [20, 21, 24] to commercial and research prototypes of Web information and ontology management systems [11, 16, 13, 53, 41, 54].

Inheritance is one of the most fundamental features of object-oriented systems. Code inheritance, realized through instance methods definitions, is commonly used in imperative languages like C^{++} and Java, while value inheritance, realized through class and object method definitions, is commonly used in AI.

Nonmonotonic multiple inheritance is of great significance, due to its importance in object-oriented modeling, its crucial role in the emerging field of ontology management [15, 16, 53, 41], and its increasing use in the field of security policy management, especially discretionary and role-based access control [52, 46, 50, 3, 25, 10]. Thus, unifying inheritance and deduction opens up new, important application areas.

Most of the previous proposals in the literature fail to account for a clean, model-theoretic semantics for nonmonotonic multiple inheritance in the presence of dynamic class hierarchies, when inheritance and deduction closely interact. In addition, no object-oriented knowledge base system provides a satisfactory solution to the problem of incorporating value inheritance and code inheritance into one coherent system.

In this work, we have developed a natural model-theoretic semantics, called *optimistic object model semantics*, for value and code inheritance in object-oriented knowledge bases, which supports implicit inference by inheritance as well as explicit deductive inference via rules. Inference by inheritance supports a multitude of features, such as overriding, nonmonotonic multiple inheritance, meta programming, and dynamic inheritance hierarchies — the important features that are fundamental to advanced object-oriented knowledge management.

We adopt the well-founded semantics [18, 17] and extend it with the ideas of locality and context [26]. In the setting of three-valued models, we formalize the notions of locality, context, and inheritance candidacy, and formally define the inheritance postulates. These postulates capture the common intuition behind overriding and conflict resolution in nonmonotonic multiple value and code inheritance, and specify the minimum requirements for an object model of a program. We also specify an extended alternating fixpoint procedure to compute a unique object model, called *optimistic object model*, for any program. Moreover, we show the implementation of our semantics by a rewriting algorithm which translates programs written in our query language to general logic programs under the well-founded semantics, and formally prove that this implementation is correct with respect to the new semantics.

Our new semantics satisfies all the requirements for object-oriented knowledge base systems listed in Section 3.2, yields intuitively satisfactory results in all known benchmark cases, does not impose syntactic restrictions on programs (beyond requiring them to be rule-based), and has been implemented in the Flora-2 system [59]. This new semantics is robust in the sense that it has at least three different characterizations: the optimistic object model is the least fixpoint of an extended alternating fixpoint operator; it is the least three-valued stable object model with respect to information ordering; and it is a minimal object model with respect to truth ordering.

To the best of our knowledge, the optimistic object model semantics is currently the only model-theoretic semantics for nonmonotonic multiple value and code inheritance that applies to general, unrestricted object-oriented knowledge bases.

The rest of this dissertation is organized as follows. In Chapter 2 we first introduce the basic concepts of overriding, nonmonotonic multiple inheritance, value inheritance, and code inheritance. In this chapter we will also illustrate the problems that result from combining inheritance and deduction in object-oriented knowledge bases.

Chapter 3 summarizes our previous research and surveys the literature on value and code inheritance. Chapter 4 introduces the preliminaries including the basic fixpoint theory and the well-founded semantics for general logic programs.

In Chapter 5 we introduce the syntax of a simplified query language and define a three-valued semantics for F-logic programs written in this language. Chapter 6 formalizes the notion of an object model as well as the inheritance postulates that an object model should satisfy. The operators that can be used to compute object models are defined in Chapter 7.

In Chapter 8 we introduce stable object models which satisfy a certain computational property of the operators defined in Chapter 7. Chapter 9 introduces optimistic object models, which are uniquely defined for F-logic programs, and shows that the optimistic object model of an F-logic program is the least stable object model with respect to information ordering. In Chapter 10 we introduce truth ordering among

the object models of an F-logic program and formally prove that optimistic object models are minimal with respect to this truth ordering.

Chapter 11 presents a rewriting algorithm that translates a given F-logic program into a certain general logic program, and proves the isomorphism between the well-founded model of the rewritten program and the optimistic object model of the original F-logic program. The data complexity of the optimistic object model semantics is also discussed in this chapter. Finally, Chapter 12 discusses future work and concludes this dissertation.

Chapter 2

Inheritance in a Nutshell

In this chapter we will illustrate through examples the basic concepts that play an important role in object-oriented knowledge base systems: overriding, nonmonotonic multiple inheritance, value inheritance, and code inheritance. At the end, we will also discuss the issues of conflicts due to the close interaction between inheritance and deduction.

2.1 Methods and Class Hierarchies

To make the exposition easier to follow, we will use a subset of the F-logic syntax to present the examples in this chapter. The simplified language includes only three kinds of atomic statements: those that represent class memberships, subclass relationships, and (inheritable) multivalued method[1] specifications.

An atom of the form $\mathsf{o}:\mathsf{c}$ says that the object o is a member[2] of the class c; $\mathsf{s}::\mathsf{c}$ says that the class s is a subclass of the class c; and $\mathsf{e}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$ [3] specifies that e, either an object or a class, has a multivalued method, m, whose return value is a set, and v is one of the members in that set. The symbols o, c, s, e, m, and v in the above atomic formulas are first-order terms. They represent the IDs of objects, classes, methods, and values of methods.

[1] We use the words *method* and *attribute* interchangeably in this dissertation.

[2] We use the words *member* and *instance* interchangeably in this dissertation.

[3] To reduce clutter, we do not syntactically distinguish between inheritable and noninheritable methods. Moreover, since single-valued methods are a special case of multivalued methods (with the additional constraint that at most one return value is allowed), we usually use multivalued methods to model single-valued methods.

2.2 Overriding and Multiple Inheritance

In object-oriented languages, overriding means that definitions from a more specific class take precedence over definitions in a more general class. For instance, consider the following classical example.

Example 2.2.1 (Royal Elephants) The program here states that the color of elephants is gray and clyde is a royal elephant, which, of course, is an elephant.

```
elephant[color →→ gray].
royalElephant :: elephant.
clyde : royalElephant.
```

What is the color of clyde? Although its color is not given directly in the above program, we can infer clyde[color →→ gray] by inheritance, since clyde is an elephant and so it inherits the color, gray, of elephants.

Now suppose we learn that the color of royal elephants is white and the above program is updated to reflect this new information:

```
elephant[color →→ gray].
royalElephant[color →→ white].
royalElephant :: elephant.
clyde : royalElephant.
```

Although earlier we established that the color of clyde is gray, we must withdraw this conclusion because of the newly added information. Since more specific definitions override less specific ones, clyde should inherit the color, white, from the more specific class royalElephant. □

Clearly, overriding leads to *nonmonotonic* inheritance. In nonmonotonic inheritance, new base facts do not necessarily lead to new inherited facts and might even lead to withdrawal of previous conclusions made by inheritance. For instance, in the above Example 2.2.1, the addition of the base fact royalElephant[color →→ white] invalidates the fact clyde[color →→ gray] that was drawn previously. But overriding is not the only source of nonmonotonic inheritance. In cases where an object belongs to multiple incomparable classes, inheritance conflicts can arise and so their "canceling" effects can lead to nonmonotonic inheritance as well. This phenomenon is illustrated by another classical example that follows.

Example 2.2.2 (Nixon Diamond) The program here says that quakers in general are pacifists whereas republicans are usually hawks.

quaker[policy ⟶⟶ pacifist].
republican[policy ⟶⟶ hawk].
nixon : quaker.
nixon : republican.

Which policy should nixon inherit, from quaker or republican? There are three possible approaches to the problem. First, in the *monotonic* approach, nixon inherits the policies from the two classes, quaker and republican. So we could derive *both* nixon[policy ⟶⟶ pacifist] *and* nixon[policy ⟶⟶ hawk]. Second, in the *nondeterministic* approach, we require that inheritance should take place from a *unique* source. So we would randomly select one of the two classes for inheritance and derive *either* nixon[policy ⟶⟶ pacifist] *or* nixon[policy ⟶⟶ hawk], but *not* both. Third, in the *cautious* approach, we disallow inheritance for nixon from these two classes because they have different values defined for the same method policy — a multiple inheritance conflict. Thus we derive *neither* nixon[policy ⟶⟶ pacifist] *nor* nixon[policy ⟶⟶ hawk]. In this dissertation, we pursue the cautious approach to resolving multiple inheritance conflicts. □

2.3 Value and Code Inheritance

A traditional object-oriented database schema normally distinguishes between two kinds of methods: instance methods and class methods. While *instance method definitions* characterize all instances (members) of a class, *class method definitions* characterize the class itself as an object [46]. When we specify an instance method for a class, the *code* that defines the method is to be inherited by all instances of this class and the method value is computed for each instance. On the other hand, when we specify a class method for a class, the value of the method is computed for this class object and the result is then inherited by all instances of this class. Moreover, we might need to explicitly define methods for individual objects, which act as *local* definitions and override the definition for the same method inherited from a superclass. We will call these definitions *object method definitions*. They are similar to class method definitions except that they are not intended for inheritance.

Example 2.3.1 (Value Inheritance) Suppose we want to compute bonuses for employees in the software department. Our policy is to award bonus based on the overall sales of the entire department. For example, every employee gets a bonus of 1% of the total amount of sales. This idea can be encoded using the following program.

$$\mathsf{softwareDept}[\mathsf{bonus} \twoheadrightarrow \mathsf{N}] \impliedby \mathsf{softwareDept}[\mathsf{salesTotal} \twoheadrightarrow \mathsf{S}],\ \mathsf{N} = \mathsf{S} * 1\%.$$
$$\mathsf{softwareDept}[\mathsf{salesTotal} \twoheadrightarrow 1000].$$
$$\mathsf{john} : \mathsf{softwareDept}.$$
$$\mathsf{mary} : \mathsf{softwareDept}.$$

The first two clauses in the above program are class method definitions for the methods, bonus and salesTotal, of the class softwareDept, respectively. The first rule defines the method bonus, whose value depends on the method salesTotal, which is specified as a fact. According to these two clause, we can infer $\mathsf{softwareDept}[\mathsf{bonus} \twoheadrightarrow 10]$.

The last two clauses simply say that john and mary are members of the class softwareDept. Although the program does not explicitly define the method bonus for john, john will inherit the method bonus *and* its value (*i.e.*, 10) from the class softwareDept, since john has been known to be a member of it. Similarly, we can derive $\mathsf{mary}[\mathsf{bonus} \twoheadrightarrow 10]$. □

Example 2.3.2 (Code Inheritance) Now we want to compute bonuses for all employees in the hardware department, but this time using a policy that rewards individual performance. For example, every employee gets a bonus of 10% of the amount of sales he/she has achieved. This idea can be illustrated using the following program.

$$\mathsf{code}\quad \mathsf{hardwareDept}[\mathsf{bonus} \twoheadrightarrow \mathsf{N}] \impliedby \mathsf{hardwareDept}[\mathsf{sales} \twoheadrightarrow \mathsf{S}],\ \mathsf{N} = \mathsf{S} * 10\%.$$
$$\mathsf{mike} : \mathsf{hardwareDept}.$$
$$\mathsf{lucy} : \mathsf{hardwareDept}.$$
$$\mathsf{mike}[\mathsf{sales} \twoheadrightarrow 300].$$
$$\mathsf{lucy}[\mathsf{sales} \twoheadrightarrow 200].$$

Note that the first rule is marked with a special keyword, code, which says that it is an instance method definition. It defines the method bonus for all instances of the class hardwareDept. Intuitively, the name hardwareDept in this rule can be considered as a "placeholder" that stands for a member of the class hardwareDept. The rest of the program states that mike and lucy are members of hardwareDept, and defines sales figures for mike and lucy, respectively.

Let us see how the method bonus can be computed for mike. Since mike is a member of hardwareDept and the first rule in the above program defines the method bonus for all instances of hardwareDept, mike will inherit this rule to compute bonus for himself. However, when inherited, this rule becomes instantiated for the object mike as follows:

$$\mathsf{mike}[\mathsf{bonus} \twoheadrightarrow \mathsf{N}] \impliedby \mathsf{mike}[\mathsf{sales} \twoheadrightarrow \mathsf{S}],\ \mathsf{N} = \mathsf{S} * 10\%.$$

i.e., mike gets substituted for hardwareDept. This corresponds to the so called *late binding* in traditional object-oriented languages like C^{++} and Java. It then follows that mike[bonus $\twoheadrightarrow$ 30] must be true. Similarly, we can derive lucy[bonus $\twoheadrightarrow$ 20]. □

Inheritance via instance method definitions, as illustrated in Example 2.3.2, is called *code inheritance*, because it is the code that gets inherited as opposed to results of methods. Code inheritance is commonly used in imperative object-oriented languages such as C^{++} and Java. In contrast, inheritance via class method definitions, as illustrated in Example 2.3.1, is called *value inheritance*, because it is the results (when established) of methods that get inherited. This kind of inheritance is commonly used in AI [55, 36].

One of the fundamental differences between value inheritance and code inheritance is that value inheritance is *data-dependent* whereas code inheritance is not. The difference becomes apparent when class and object method definitions are specified by rules. If a class method is defined using a rule, then the "inheritability" of this definition hinges on the satisfiability of the rule body in the database. In other words, the value specified by the rule head becomes inheritable *only if* its truth can be established in the model for the program. Similarly, when an object method is specified using a rule, whether the method is actually locally defined or not depends on the satisfiability of the rule body.

In contrast, although instance method definitions are also specified using rules, inheritability of these rules is independent of whether the rule body is satisfied or not. After all, it is the code but not the value that is inherited. Therefore, code inheritance can be resolved when the class hierarchy is fixed. This is not true, however, for value inheritance, because the truth values of atoms depend on the current state of the database.

By combining the ideas of value and code inheritance together, we can design more interesting applications, as illustrated by the following example.

Example 2.3.3 In Example 2.3.2, the bonus plan in the hardware department was such that every employee will get a bonus of 10% of the amount of sales he/she has achieved. Suppose mike has a special deal and will get 15% if his sales exceeds 500; otherwise, his bonus is determined using the general department-wide policy. The program segment to accomplish this goal is as follows.

code
$$\begin{aligned} &\mathsf{hardwareDept[bonus} \twoheadrightarrow \mathsf{N]} \longleftarrow \mathsf{hardwareDept[sales} \twoheadrightarrow \mathsf{S]},\ \mathsf{N} = \mathsf{S} * 10\%. \\ &\mathsf{mike} : \mathsf{hardwareDept}. \\ &\mathsf{mike[bonus} \twoheadrightarrow \mathsf{N]} \longleftarrow \mathsf{mike[sales} \twoheadrightarrow \mathsf{S]},\ \mathsf{S} > 500,\ \mathsf{N} = \mathsf{S} * 15\%. \end{aligned}$$

The first rule here is an instance method definition for the class hardwareDept, while the last rule is an object method definition for mike.

It is instructive to see how the method bonus can be evaluated on mike under two different conditions. On one hand, if the database contains the fact mike[sales $\twoheadrightarrow$ 600], then according to the last rule we can directly derive the fact mike[bonus $\twoheadrightarrow$ 90]. This is a local property for mike which overrides the inheritance of the instance method definition for bonus in the first rule. No code inheritance takes place in this case. On the other hand, if the database contains the fact mike[sales $\twoheadrightarrow$ 300], then the body of the last rule cannot be satisfied and a value of bonus is not locally defined for mike. In this case, because mike belongs to hardwareDept, he can inherit the code for computing the method bonus from the class hardwareDept. Therefore, we derive mike[sales $\twoheadrightarrow$ 30]. □

2.4 Dynamic Class Hierarchies

When class memberships and/or subclass relationships are defined using rules, the class hierarchy is no longer fixed; it becomes data-dependent in the sense that class memberships and subclass relationships depend on the particular set of facts that the knowledge base contains. We call such class hierarchies *dynamic* because they can only be decided at runtime but not at compile time.

For instance, consider the following rule:

$$\mathsf{c1} :: \mathsf{c2} \longleftarrow \mathsf{c1}[\mathsf{m} \twoheadrightarrow \mathsf{v}].$$

This rule says that whether or not the class c1 is a subclass of the class c2 depends on the satisfiability of the fact c1[m $\twoheadrightarrow$ v]. If the database implies c1[m $\twoheadrightarrow$ v] then c1 is a subclass of c2; otherwise, it is not. Consequently, inheritance decisions concerning the class c1 have to be delayed until runtime.

Because of dynamic class hierarchies, complex interactions between inheritance and deduction can come into play, as illustrated by the following example.

Example 2.4.1 In program here, the class goods represents all products and the first rule says how to compute their prices, *i.e.*, price is cost plus profit margin. The class luxuryGoods is a subclass of goods and represents those products whose cost exceeds 500. In commerce, normal goods and luxury goods use different profit margins. We encode this information by assigning different values, 100% and 200%, to the class method margin of goods and luxuryGoods, respectively.

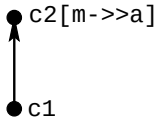
```
code  goods[price →→ P]  ⟵
            goods[cost →→ C], goods[margin →→ M], P = C * (1 + M).
      luxuryGoods :: goods.
      X : luxuryGoods  ⟵ X : goods, X[cost →→ P], P > 500.
      goods[margin →→ 100%].
      luxuryGoods[margin →→ 200%].
      steelWatch : goods.
      diamondRing : goods.
      steelWatch[cost →→ 100].
      diamondRing[cost →→ 600].
```

Consider the price for the product steelWatch. Since its cost is 100, it is not a luxury product. So its profit margin is 100%. Thus we derive steelWatch[price →→ 200]. On the other hand, the product diamondRing costs 600. So it is a luxury product. Therefore, it inherits its profit margin from luxuryGoods instead of goods, *i.e.*, 200%. Thus we derive diamondRing[price →→ 1800]. □

2.5 Inheritance and Deduction

The interactions between inheritance and deduction also bring up challenging problems in defining and computing inheritance semantics, especially in the presence of dynamic class hierarchies. We now illustrate some of the main difficulties. Although here we present the examples using value inheritance only, the same problems can also arise in the context of code inheritance. These issues were first explored in [32] but did not receive a satisfactory solution.

In the following examples, a solid arrow from a node c_1 to another node c_2 means that c_1 is either a subclass or a member of c_2. All examples in this section are discussed informally. The formal treatment will be given in Chapters 6 and 7.

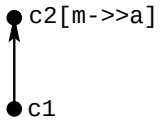


$c_1 : c_2$.
$c_2[m \twoheadrightarrow a]$.
$c_2[m \twoheadrightarrow b] \longleftarrow c_1[m \twoheadrightarrow a]$.

Figure 1: Inheritance through Context

Example 2.5.1 Consider the program in Figure 1. Without inheritance semantics, this program has a unique model, which consists of the first two facts. According to the common intuition behind inheritance, c_1 ought to inherit $m \twoheadrightarrow a$ from c_2. However, just adding the fact $c_1[m \twoheadrightarrow a]$ will not make the resulting set a model, since the last rule is no longer satisfied: The least model that contains the inherited

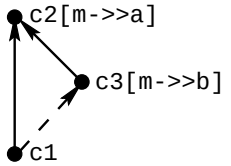
fact should also include $c_2[m \twoheadrightarrow b]$. However, this begs the question as to whether c1 should inherit $m \twoheadrightarrow b$ from c_2 as well. The intuition suggests that the intended model should be "stable" with respect to not only deduction but inheritance as well. Therefore, $c_1[m \twoheadrightarrow b]$ also should be in that model. This problem was recognized in [32], but the proposed solution was not stable in the above sense, because it was not based on semantic principles but rather on an ad hoc definition of a plausible fixpoint computation. □



$c_1 : c_2.$
$c_2[m \twoheadrightarrow a].$
$c_1[m \twoheadrightarrow b] \longleftarrow c_1[m \twoheadrightarrow a].$

Figure 2: Interaction between Derived and Inherited Facts

Example 2.5.2 Now consider the program in Figure 2, which is the same as the program in Figure 1 except for the head of the last rule. Again, the intuition suggests that $c_1[m \twoheadrightarrow a]$ ought to be derived via inheritance, and $c_1[m \twoheadrightarrow b]$ be derived to make the resulting set of facts into a model in the conventional sense. This, however, leads to the following observation. The method m of c_1 now has one value, a, which is inherited, and another value, b, which is derived via a rule. Although the traditional frameworks for inheritance were developed without deduction in mind, it is clear that derived facts like $c_1[m \twoheadrightarrow b]$ are akin to "local" method definitions and so should be treated similarly. In particular, local definitions always override inheritance. The conclusion is that although derivation is done "after" inheritance, its existence undermines the original reason for inheritance. This is similar to the known phenomenon where a reasoner rejects an assumption when it leads a contradiction. Again, the framework presented in this dissertation, which is based on semantic principles, differs from the ad hoc computation in [32] (which keeps both derived and inherited facts). □

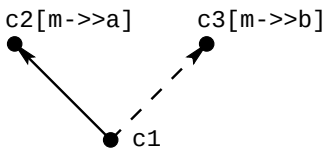


$c_1 : c_2.$
$c_3 :: c_2.$
$c_2[m \twoheadrightarrow a].$
$c_3[m \twoheadrightarrow b].$
$c_1 : c_3 \longleftarrow c_1[m \twoheadrightarrow a].$

Figure 3: Interaction between Inheritance and Derived Intervening Superclass

Example 2.5.3 The program in Figure 3 shows a case where inheritance changes the class hierarchy, which creates conditions that undermine the original reason for

inheritance. Initially, c_3 is not known to be a superclass of c_1. So, it seems that c_1 can inherit $m \twoheadrightarrow a$ from c_2. However, this makes the fact $c_1[m \twoheadrightarrow a]$ true, which in turn causes $c_1 : c_3$ to be derived by the last rule of the program. Since this makes c_3 a more specific superclass of c_1 than c_2, it appears that c_1 ought to inherit $m \twoheadrightarrow b$ from c_3 rather than $m \twoheadrightarrow a$ from c_2. However, this would make the fact $c_1 : c_3$ unsupported. Either way, the deductive inference enabled by the original inheritance undermines the support for the inheritance itself. Unlike [32], a logically correct solution in this case would be to leave the truth values of both $c_1 : c_3$ and $c_1[m \twoheadrightarrow a]$ undecided. The dashed arrow from c_1 to c_3 indicates that $c_1 : c_3$ is neither true nor false. □



$c_2[m \twoheadrightarrow a].$
$c_3[m \twoheadrightarrow b].$
$c_1 : c_2.$
$c_1 : c_3 \longleftarrow c_1[m \twoheadrightarrow a].$

Figure 4: Derived Multiple Inheritance

Example 2.5.4 The last program, in Figure 4, illustrates a similar problem, but this time it occurs in the context of multiple inheritance. Initially c_3 is not known to be a superclass of c_1. So there is no multiple inheritance conflict and the intuition suggests that c_1 should inherit $m \twoheadrightarrow a$ from c_2. But then $c_1 : c_3$ has to be added in order to satisfy the last rule, which makes c_3 a superclass of c_1 and introduces a multiple inheritance conflict. As in the previous example, although this conflict became apparent only after inheritance took place, it undermines the original reason for inheritance (which was based on the assumption that $c_2[m \twoheadrightarrow a]$ is the only source of inheritance for c_1). Therefore, the truth values of $c_1[m \twoheadrightarrow a]$ and $c_1 : c_3$ should be neither true nor false. Again, this conclusion differs from [32]. □

Chapter 3

Related Work

In this chapter, we first summarize our previous research and then survey the literature on value and code inheritance in object-oriented knowledge bases. Finally, we discuss issues related to simulating code inheritance using value inheritance.

3.1 Summary of Previous Research

The FLORA system was our first object-oriented knowledge base research prototype. The main ideas of this implementation could be traced back to the FLIP system [39]. Implemented using the efficient tabling inference engine of XSB [49, 9], the FLORA system demonstrated good performance comparable to other similar systems such as FLORID [40] that was implemented with C^{++}. This early work proved the feasibility of our implementation approach.

The original FLORA system played a central role in the first prototype of the commercial Web information management system XRover [11]. This success led us to embark on the more ambitious Flora-2 system [59], which incorporates F-logic [31, 32], HiLog [8], and Transaction Logic [4, 5] into a single, coherent logic language along the lines described in [30].

In [58], we studied some of the system design and optimization issues involving the Flora-2 system, including its novel module system, path expressions in rule heads, transactions in a tabling environment, and a specialization technique designed to improve indexing.

In the early versions of the Flora-2 system, implementation of inheritance was based on preliminary ideas, which led to the development of the formal semantics to be described later in this dissertation. Interestingly, although the ideas underlying this implementation "seemed" right, we later discovered that it was semantically incorrect and our subsequent theoretical study helped us fix this problem. The incorrect

behavior did, in fact, surface in real programs that users of Flora-2 wrote.

As the motivating examples in Chapter 2 have already shown, the correct solution for inheritance semantics is not at all obvious. Our first step towards tackling the problem of inheritance semantics was to develop a model-theoretic semantics for nonmonotonic multiple value inheritance [60]. Our current, comprehensive model theory for value and code inheritance is built on top of our previous work reported in [60], which dealt with model-theoretic semantics for value inheritance only.

3.2 Related Work in the Literature

We now briefly survey the literature on value and code inheritance in object-oriented knowledge bases. To make our comparison concrete, we first list the main features that, in our opinion, an object-oriented knowledge base system with value and code inheritance must possess:

(1) Implicit inference by inheritance, as well as explicit inference via rules.

(2) Dynamic class hierarchies, *i.e.*, the ability to define both class memberships and subclass relationships via rules. Although some proposals allow defining class hierarchies using rules, they do not allow queries on either class or object methods to appear in the bodies of these rules. So the class hierarchies can be decided independently of class and object methods. In such a case, we will still call these class hierarchies static.

(3) Data-dependent inheritance. This feature is closely related to value inheritance. If inheritability and locality of a method definition rely on the facts stored in the database, then we call such inheritance data-dependent; otherwise, it is data-independent.

(4) Overriding by intermediate superclasses. Here we are also interested in whether the semantics takes into account the interactions between value inheritance and code inheritance.

(5) Nonmonotonic inheritance from multiple superclasses that are incomparable with respect to subclass relationships. Some proposals avoid the need for resolving multiple inheritance conflicts by imposing syntactic restrictions on programs. In such cases, we will say that these proposals do not support nonmonotonic multiple inheritance.

(6) Meta-programming, by which variables can range over class and method names.

(7) Late binding. This feature is commonly found in imperative object-oriented languages such as C^{++} and Java that support code inheritance. Supporting

late binding requires resolving method names at runtime, when the class from which the code is inherited is decided.

There is a large body of work based on Touretzky's framework of Inheritance Nets [55]. On one hand, the overriding mechanism in this framework is more sophisticated than what is typically considered in the knowledge base context. On the other hand, this framework supports neither deductive inference via rules nor dynamic class hierarchies, which makes it too weak for many applications of knowledge bases. [36] surveys several different approaches to computing inheritance semantics based on Inheritance Nets. We will not discuss this framework any further here.

There is also a large body of work on extending traditional relational database systems with object-oriented features. But most proposals do not support deduction via inference rules, which, as we saw, prevents the main difficulty in defining inheritance semantics. Therefore we will not discuss such works. For a comprehensive survey on these works we refer the readers to [33].

Ullman surveyed several deductive object-oriented database systems in [56]. However, his main concerns were object identity and dynamic typing, which are orthogonal to our concerns.

Although F-logic [31, 32] resolved many semantic and proof-theoretic issues in object-oriented knowledge bases, the original semantics for inheritance in F-logic was defined through a nondeterministic inflationary fixpoint [32], which was not matched by a corresponding model theory. The original F-logic fixpoint procedure was known to produce questionable results (cf. Section 2.5) when inheritance and deduction interact. Moreover, only value inheritance was considered in the original F-logic.

Ordered Logic [35, 34] includes certain abstractions of the object-oriented paradigm. In this framework, both positive and negative literals are allowed in rule heads, and inference rules are grouped into a set of modules that collectively form a static class hierarchy. Although Ordered Logic supports overriding and propagation of rules among different modules, the idea of late binding is not built into the logic. Since it is primarily committed to resolving inconsistency between positive and negative literals, its semantics has a strong value-based value inheritance flavor.

Abiteboul et al. [1] propose a framework for implementing inheritance that is based on program rewriting using Datalog with negation. In spirit this implementation is close to our implementation in Flora-2. However, [1] lacks strong theoretical underpinnings, such as an independent model-theoretic formalization. On the practical side, this framework excludes nonmonotonic multiple inheritance and makes a strong assumption that programs rewritten by the algorithm in [1] must have a total (two-valued) well-founded model. This latter assumption does not generally hold without strong syntactic restrictions that force program stratification.

In [14], Dobbie and Topor develop a model-theoretic semantics for monotonic code inheritance in the object-oriented database language Gulog. A special feature of their language is that all variables in a program must be explicitly typed according to a separate signature declaration. However, they do not support data-dependent value inheritance and only consider positive programs with a static class hierarchy. More importantly, nonmonotonic multiple inheritance is not built into their semantics. Instead, syntactic restrictions are imposed on the programs to avoid multiple inheritance conflicts.

Jamil and Lakshmanan [29] introduce the deductive object-oriented database language ORLog, and propose a model theory for nonmonotonic multiple code inheritance. This work discusses the techniques for resolving inheritance conflicts. But it does not support data-dependent value inheritance and only considers static class hierarchies.

The work of Bugliesi and Jamil [7] attempts to develop a model theory that accounts for both value and code inheritance, which bears close resemblance to two-valued stable models [19]. However, their semantics applies only to programs *without* negation in rule bodies (a severe limitation) and does not handle multiple inheritance conflicts properly, making it monotonic instead. In addition, their framework does not support data-dependent value inheritance, and more importantly, it does not provide an algorithm to compute a canonical model under their semantics.

May et al. [43] propose to apply the alternating fixpoint procedure behind the well-founded semantics to evaluate F-logic programs. However, inheritance is still dealt with in the same way as in the original F-logic. Deduction and inheritance are computed in two separate stages and so the computation process has an inflationary fixpoint flavor. Apart from being ad hoc, this semantics is known to produce counter-intuitive results when dynamic class hierarchies interact with overriding and multiple inheritance (cf. Section 2.5).

Recently May and Kandzia [42] show that the original F-logic semantics can be described using the *inflationary* extension of Reiter's Default Logic [48]. In their framework, inheritance semantics is encoded using *defaults*. As in [32], F-logic programs still have a two-valued semantics. However, instead of adopting the full-blown semantics of Default Logic, which is not even semidecidable, they introduce the *inflationary* extension of it. Their inheritance strategy is inflationary in the sense that once a fact is derived through inheritance it is never undone. Therefore, a later inference might invalidate the original conditions (encoded as justifications of defaults) for inheritance (cf. Section 2.5). Moreover, nonmonotonic multiple inheritance is handled in such a way that when multiple incomparable inheritance sources exist, one of them is randomly selected for inheritance instead of none (as in our work). Finally, code inheritance is not considered in [42].

In [26], Jamil introduces the Datalog^{++} language which supports encapsulation and both value and code inheritance. He proposes a series of techniques to tackle the inheritance problem. Among these, the ideas of *locality* and *context* have influenced our approach the most. However, this work does not support dynamic class hierarchies or meta-programming. Data-dependent value inheritance is not supported either. Moreover, the inheritance semantics in this work is *ad hoc* and is defined using program rewriting (lack of theoretical underpinnings), although it does support late binding through an elegant *completion* technique.

A more recent proposal of Jamil [28] adopts a proof-theoretic approach to defining inheritance semantics. He extends the query language introduced in [26] and provides new syntax to denote rules with different inheritance types and different inheritance modes. However, only static class hierarchies are allowed in this framework and the proof theory does not account for nonmonotonic multiple inheritance. Finally, data-dependent value inheritance is not supported, since inheritability is defined basically the same way as in [26].

A number of other works partially address inheritance issues in knowledge bases. For instance, [38] defines signature-based inheritance, which does not provide an overriding mechanism. In [2], inheritance is defined using the framework for modularity in logic programming developed in [6]. However, this approach does not support multiple inheritance and dynamic class hierarchies.

3.3 Simulating Code Inheritance

The kind of inheritance considered in the original F-logic [32] as well as in the new model theory developed by us [60] is value inheritance. Instance method definitions are not supported at the language level and do not appear in the semantics. However, it has been shown that value inheritance can simulate code inheritance using the meta programming feature of F-logic [32], and so pure value inheritance systems are considered to be more general than pure code inheritance systems.

Example 3.3.1 To see how the simulation works, suppose we want to achieve the same effect of code inheritance as in the following program by using value inheritance only.

$$
\begin{array}{ll}
\mathsf{code} & \mathsf{c1}[\mathsf{m} \twoheadrightarrow \mathsf{V}] \longleftarrow \mathsf{c1}[\mathsf{f} \twoheadrightarrow \mathsf{V}]. \\
\mathsf{code} & \mathsf{c2}[\mathsf{m} \twoheadrightarrow \mathsf{V}] \longleftarrow \mathsf{c2}[\mathsf{g} \twoheadrightarrow \mathsf{V}]. \\
 & \mathsf{c2} :: \mathsf{c1}. \\
 & \mathsf{o} : \mathsf{c1}. \\
 & \mathsf{o} : \mathsf{c2}.
\end{array}
$$

The first two rules above define the method m for all instances of the class $\mathsf{c1}$ and

c2, respectively. Since c2 is a subclass of c1, its definition for m overrides that of c1. Therefore, the object o inherits the definition for m from c2. Following the general simulation techniques as described in [32], we can rewrite the above program as follows.

$$
\begin{aligned}
&\mathsf{c1}[\mathsf{m\_name} \twoheadrightarrow \mathsf{m\_c1}].\\
&\mathsf{X}[\mathsf{m\_c1} \twoheadrightarrow \mathsf{V}] \;\longleftarrow\; \mathsf{X} : \mathsf{c1},\ \mathsf{X}[\mathsf{f} \twoheadrightarrow \mathsf{V}].\\
&\mathsf{c2}[\mathsf{m\_name} \twoheadrightarrow \mathsf{m\_c2}].\\
&\mathsf{X}[\mathsf{m\_c2} \twoheadrightarrow \mathsf{V}] \;\longleftarrow\; \mathsf{X} : \mathsf{c2},\ \mathsf{X}[\mathsf{g} \twoheadrightarrow \mathsf{V}].\\
&\mathsf{c2} :: \mathsf{c1}.\\
&\mathsf{o} : \mathsf{c1}.\\
&\mathsf{o} : \mathsf{c2}.\\
&\mathsf{X}[\mathsf{m} \twoheadrightarrow \mathsf{V}] \;\longleftarrow\; \mathsf{X}[\mathsf{m\_name} \twoheadrightarrow \mathsf{M}],\ \mathsf{X}[\mathsf{M} \twoheadrightarrow \mathsf{V}].
\end{aligned}
$$

In the above program, the new method name m_c1 is introduced to implement the method m for the class c1, and m_c2 is introduced to implement the method m for c2. Moreover, the new method name, m_name, is needed, so that the same effect of code inheritance for the method m can be achieved by having an object inherit the appropriate method name through m_name and then evaluate this method on itself. This is shown by the last rule in the above program. Thus we can derive $\mathsf{o}[\mathsf{m\_name} \twoheadrightarrow \mathsf{m\_c2}]$ by value inheritance. And any call to $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{V}]$ will result in a call to $\mathsf{o}[\mathsf{m\_c2} \twoheadrightarrow \mathsf{V}]$, and then to $\mathsf{o}[\mathsf{g} \twoheadrightarrow \mathsf{V}]$. This exactly what the original code inheritance would have achieved. □

Although value inheritance is capable of simulating code inheritance, there are a number of disadvantages in the simulation:

(1) Programmers bear the burden of introducing new method names that must be *unique* in the knowledge base. This problem may be overcome by automatic program translation, however.

(2) The size of the simulation program increases. In the worst case it can double.

(3) The declarativeness of code inheritance is degraded after simulation, because it is hard to foresee all the consequences of rewriting.

(4) Redundant information is forced into the canonical model of the rewritten program. For instance, for the simulation program in Example 3.3.1, its model has to include atoms like $\mathsf{o}[\mathsf{m\_c1} \twoheadrightarrow \mathsf{x}]$, which are "meaningless" to users. Therefore, simulation is not friendly to bottom-up processors that may decide to materialize the entire program.

A much bigger problem is that the general simulation techniques do not naturally lend themselves to the integration of value and code inheritance. This problem is illustrated by the following example.

Example 3.3.2 Let us apply the general simulation techniques to the following program which is copied from the previous Example 2.3.3.

$$
\begin{array}{ll}
\mathsf{code} & \mathsf{hardwareDept}[\mathsf{bonus} \twoheadrightarrow \mathsf{N}] \ \longleftarrow \mathsf{hardwareDept}[\mathsf{sales} \twoheadrightarrow \mathsf{S}],\ \mathsf{N} = \mathsf{S} * 10\%. \\
& \mathsf{mike} : \mathsf{hardwareDept}. \\
& \mathsf{mike}[\mathsf{bonus} \twoheadrightarrow \mathsf{N}] \ \longleftarrow \mathsf{mike}[\mathsf{sales} \twoheadrightarrow \mathsf{S}],\ \mathsf{S} > 500,\ \mathsf{N} = \mathsf{S} * 15\%.
\end{array}
$$

We will get the following rewritten program, which contains class and object method definitions only.

$$
\begin{array}{l}
\mathsf{hardwareDept}[\mathsf{bonus\_name} \twoheadrightarrow \mathsf{bonus\_hardwareDept}]. \\
\mathsf{X}[\mathsf{bonus\_hardwareDept} \twoheadrightarrow \mathsf{N}] \ \longleftarrow \mathsf{X} : \mathsf{hardwareDept},\ \mathsf{X}[\mathsf{sales} \twoheadrightarrow \mathsf{S}],\ \mathsf{N} = \mathsf{S} * 10\%. \\
\mathsf{mike} : \mathsf{hardwareDept}. \\
\mathsf{X}[\mathsf{bonus} \twoheadrightarrow \mathsf{N}] \ \longleftarrow \mathsf{X}[\mathsf{bonus\_name} \twoheadrightarrow \mathsf{M}],\ \mathsf{X}[\mathsf{M} \twoheadrightarrow \mathsf{N}]. \\
\mathsf{mike}[\mathsf{bonus} \twoheadrightarrow \mathsf{N}] \ \longleftarrow \mathsf{mike}[\mathsf{sales} \twoheadrightarrow \mathsf{S}],\ \mathsf{S} > 500,\ \mathsf{N} = \mathsf{S} * 15\%.
\end{array}
$$

However, the rewritten program does not function correctly in all cases. The problem is that the intended overriding semantics between value and code inheritance is lost after this rewriting. For example, suppose the knowledge base contains the fact $\mathsf{mike}[\mathsf{sales} \twoheadrightarrow 600]$. Then according to our model theory for value inheritance, we can derive *both* $\mathsf{mike}[\mathsf{bonus} \twoheadrightarrow 90]$ and $\mathsf{mike}[\mathsf{bonus} \twoheadrightarrow 60]$, although *only* $\mathsf{mike}[\mathsf{bonus} \twoheadrightarrow 90]$ is expected. □

Chapter 4

Preliminaries

In this chapter we introduce the background knowledge that is essential to understanding the theoretical development in this dissertation. First we introduce the basic fixpoint theory. Then we cover the well-founded semantics for general logic programs. The materials in Section 4.1 are mostly borrowed from [37]. And most of the materials in Section 4.2 are borrowed from [18] and [17].

4.1 Fixpoint Theory

Given a set S, a *relation* R on S is a subset of $R \times R$. Normally we use the infix notation xRy to represent $(x, y) \in R$. A relation R on a set S is a *partial order* if the following conditions are satisfied:

(1) xRx for all $x \in S$.

(2) for all $x, y \in S$: if xRy and yRx, then $x = y$.

(3) for all $x, y \in S$: if xRy and yRz, then xRz.

For example, let S be a set and 2^S be the set of all subsets of S. Then set inclusion, $\subseteq$, is a partial order on 2^S.

We adopt the standard notation and use $\leq$ to denote a partial order. Let S be a set with a partial order $\leq$ and X be a subset of S. Then $u \in S$ is an *upper bound* of X if $x \leq u$ for all $x \in X$. Similarly, $l \in S$ is a *lower bound* of X if $l \leq x$ for all $x \in X$.

Let S be a set with a partial order $\leq$ and X be a subset of S. Then $a \in S$ is the *least upper bound* of X, if a is an upper bound of X and $a \leq c$ for all upper bound c of X. Similarly, $b \in S$ is the *greatest lower bound* of X, if b is a lower bound of X and $d \leq b$ for all lower bound d of X. Clearly, the least upper bound of X is unique if it exists, and is denoted by $\mathrm{lub}(X)$. Similarly, the greatest lower bound of X is unique

if it exists, and is denoted by $\mathrm{glb}(X)$.

A partially ordered set L is a *complete lattice* if $\mathrm{lub}(X)$ and $\mathrm{glb}(X)$ exist for every subset X of L. We use the symbol $\top$ to denote the *top element* $\mathrm{lub}(L)$ and $\bot$ to denote the *bottom element* $\mathrm{glb}(L)$.

For example, let S be a set and 2^S be the set of all subsets of S. Then 2^S under $\subseteq$ is a complete lattice. In fact, the least upper bound of a collection of subsets of S is their union and the greatest lower bound is their intersection. The top element is S itself and the bottom element is $\emptyset$.

Let L be a complete lattice and $\mathrm{T} : L \mapsto L$ be a mapping. We say T is *monotonic* if $\mathrm{T}(x) \leq \mathrm{T}(y)$ whenever $x \leq y$. Let $a \in L$. We say that a is the *least fixpoint* of T if a is a fixpoint of T, *i.e.*, $\mathrm{T}(a) = a$, and $a \leq b$ for all fixpoint b of T. Similarly, we can define the *greatest fixpoint* of T.

Proposition 4.1.1 Let L be a complete lattice and $\mathrm{T} : L \mapsto L$ be a monotonic mapping. Then T has a least fixpoint, denoted by $\mathrm{lfp}(\mathrm{T})$, and a greatest fixpoint, denoted by $\mathrm{gfp}(\mathrm{T})$. Furthermore, $\mathrm{lfp}(\mathrm{T}) = \mathrm{glb}(\{x \mid \mathrm{T}(x) = x\}) = \mathrm{glb}(\{x \mid \mathrm{T}(x) \leq x\})$ and $\mathrm{gfp}(\mathrm{T}) = \mathrm{lub}(\{x \mid \mathrm{T}(x) = x\}) = \mathrm{lub}(\{x \mid x \leq \mathrm{T}(x)\})$.

Proof. See [37].
□

Proposition 4.1.2 Let L be a complete lattice and $\mathrm{T} : L \mapsto L$ be a monotonic mapping. Suppose $a \in L$ and $a \leq \mathrm{T}(a)$. Then $a \leq \mathrm{gfp}(\mathrm{T})$. Similarly, if $b \in L$ and $\mathrm{T}(b) \leq b$, then $\mathrm{lfp}(\mathrm{T}) \leq b$.

Proof. See [37].
□

Now we recall some elementary properties of ordinal numbers, which we will simply refer to as ordinals. Intuitively, the ordinals are what we use to count with. The first ordinal 0 is defined to be $\emptyset$. Then we define $1 = \{\emptyset\} = \{0\}$, $2 = \{\emptyset, \{\emptyset\}\} = \{0, 1\}$, and so on. The first infinite ordinal is $\omega = \{1, 2, \ldots\}$. We can specify an ordering $<$ on the collection of all ordinals by defining $\alpha < \beta$ if $\alpha \in \beta$. If α is an ordinal, the *successor* of α is the ordinal $\alpha + 1 = \alpha \cup \{\alpha\}$, which is the least ordinal greater than α. We call $\alpha + 1$ a *successor ordinal*. An ordinal is called a *limit ordinal* if it is not the successor of any ordinal. The smallest limit ordinal (apart from 0) is ω. After ω comes $\omega + 1 = \omega \cup \{\omega\}$, $\omega + 2 = (\omega + 1) + 1$, and so on. The next limit ordinal is $\omega 2$, which is the set consisting of all $n \in \omega$ and all $\omega + m$ where $m \in \omega$. Then come $\omega 2 + 1$, $\omega 2 + 2$, …, $\omega 3$, $\omega 3 + 1$, $\omega 3 + 2$, …, and so on.

Let L be a complete lattice and $\mathrm{T} : L \mapsto L$ be a monotonic mapping. We define the ordinal powers of T as follows:

$$\begin{array}{lcll} \mathrm{T}^{\uparrow 0} & = & \bot & \text{for limit ordinal } 0 \\ \mathrm{T}^{\uparrow \alpha} & = & \mathrm{T}(\mathrm{T}^{\uparrow \alpha-1}) & \text{for successor ordinal } \alpha \\ \mathrm{T}^{\uparrow \alpha} & = & \mathrm{lub}(\{\mathrm{T}^{\uparrow \beta} \mid \beta < \alpha\}) & \text{for limit ordinal } \alpha \neq 0 \\ \mathrm{T}^{\downarrow 0} & = & \top & \text{for limit ordinal } 0 \\ \mathrm{T}^{\downarrow \alpha} & = & \mathrm{T}(\mathrm{T}^{\downarrow \alpha-1}) & \text{for successor ordinal } \alpha \\ \mathrm{T}^{\downarrow \alpha} & = & \mathrm{glb}(\{\mathrm{T}^{\downarrow \beta} \mid \beta < \alpha\}) & \text{for limit ordinal } \alpha \neq 0 \end{array}$$

Proposition 4.1.3 Let L be a complete lattice with a partial order $\leq$ and $\mathrm{T} : L \mapsto L$ be a monotonic mapping. Suppose α and β are ordinals. Then:

(1) for all α: $\mathrm{T}^{\uparrow \alpha} \leq \mathrm{lfp}(\mathrm{T})$.

(2) for all α, β: if $\alpha < \beta$ then $\mathrm{T}^{\uparrow \alpha} \leq \mathrm{T}^{\uparrow \beta}$.

(3) there exists α such that $\mathrm{T}^{\uparrow \beta} = \mathrm{lfp}(\mathrm{T})$ whenever $\beta \geq \alpha$.

(4) for all α: $\mathrm{gfp}(\mathrm{T}) \leq \mathrm{T}^{\downarrow \alpha}$.

(5) for all α, β: if $\alpha < \beta$ then $\mathrm{T}^{\downarrow \beta} \leq \mathrm{T}^{\downarrow \alpha}$.

(6) there exists α such that $\mathrm{T}^{\downarrow \beta} = \mathrm{gfp}(\mathrm{T})$ whenever $\beta \geq \alpha$.

Finally, we outline the *principle of transfinite induction*, which is frequently used in the proofs throughout this dissertation. Let $P(\alpha)$ be a property about ordinals. Assume that for all ordinal β, if $P(\gamma)$ holds for all $\gamma < \beta$, then $P(\beta)$ holds. Then $P(\alpha)$ holds for all ordinal α.

4.2 Well-Founded Semantics

A *general* logic program is a finite set of rules which may have both positive and negative subgoals (also called literals) in their bodies. For instance, the following rule has a positive subgoal, $\mathsf{p}(\mathsf{X})$, and a negative subgoal, $\neg\, \mathsf{r}(\mathsf{X})$.

$$\mathsf{p}(\mathsf{X}) \longleftarrow \mathsf{q}(\mathsf{X}), \neg\, \mathsf{r}(\mathsf{X})$$

$\mathsf{p}(\mathsf{X})$ is the *head* of the rule while the rest (*i.e.*, $\mathsf{q}(\mathsf{X}), \neg\, \mathsf{r}(\mathsf{X})$) is the *body* of the rule.

It is desirable to associate one Herbrand model with a general logic program and think of that model as the "meaning" of the program, or its "declarative semantics". Ideally, queries directed to the program would be answered in accordance with this model. The well-founded semantics for general logic programs was proposed by Van Gelder et al. [18]. It assigns a unique, three-valued Herbrand model, called *well-founded model*, to every general logic program. The alternating fixpoint

computation, which is widely used to compute well-founded models of general logic programs, was introduced by Van Gelder [17]. Przymusinski also gave different characterizations of the well-founded semantics [44] and later showed that the well-founded semantics coincides with the three-valued stable semantics [45].

Given a general logic program P, its *Herbrand universe*, $\mathcal{HU}_{\mathrm{P}}$, is the set of ground (*i.e.*, variable-free) terms that use the function symbols and constants that appear in the program. The *Herbrand base*, $\mathcal{HB}_{\mathrm{P}}$, of P is the set of atomic formulas formed by predicate symbols in the program whose arguments are in the Herbrand universe.

The *Herbrand instantiation*, *ground*(P), of a general logic program P is the set of rules obtained by substituting terms in the Herbrand universe for variables in every possible way. A *ground* rule is one in the Herbrand instantiation. Although general logic programs are assumed to be a finite set of rules, their Herbrand instantiations may well be infinite. We shall be considering Herbrand instantiations while defining the well-founded semantics.

A *three-valued interpretation* $\mathcal{I}$ of a general logic program P is a triple, $\langle\, \mathrm{T}; \mathrm{U}; \mathrm{F} \,\rangle$, where T, U, and F are subsets of $\mathcal{HB}_{\mathrm{P}}$ and *pairwise disjoint*. Moreover, $\mathrm{T} \cup \mathrm{U} \cup \mathrm{F} = \mathcal{HB}_{\mathrm{P}}$. The atoms in T are *true* while the atoms in U are *undefined* and the atoms in F are *false*. Intuitively, "undefined" means possibly true or possibly false. Clearly, if any two of the three sets T, U, and F are known, then the remaining set can be decided. Sometimes when we write down a three-valued interpretation, we will only show a pair of sets and omit the other one.

Given an interpretation $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U}; \mathrm{F} \,\rangle$ and a positive subgoal L, we say that L is true in $\mathcal{I}$ if $\mathsf{L} \in \mathrm{T}$ and L is false in $\mathcal{I}$ if $\mathsf{L} \in \mathrm{F}$. Similarly, for a negative subgoal $\neg\, \mathsf{L}$, we say that $\neg\, \mathsf{L}$ is true in $\mathcal{I}$ if $\mathsf{L} \in \mathrm{F}$ and $\neg\, \mathsf{L}$ is false in $\mathcal{I}$ if $\mathsf{L} \in \mathrm{T}$.

Well-founded models are three-valued interpretations. In the original well-founded semantics [18], well-founded models are defined in terms of the set of atoms that are true (T) and the set of atoms that are false (F). However, under the alternating fixpoint semantics [17], well-founded models can be defined in terms of the set of atoms that are true (T) and the set of atoms that are undefined (U).

First we will introduce the declarative semantics of well-founded models as defined in [18][1] . A very important notion is concerned with the so-called *unfounded sets*.

Definition 4.2.1 (Unfounded Sets) Let P be a general logic program and $\mathcal{I}$ be a three-valued interpretation. We say $\mathrm{A} \in \mathcal{HB}_{\mathrm{P}}$ is an *unfounded set* with respect to $\mathcal{I}$, if each atom $p \in \mathsf{A}$ satisfies the following condition: for each ground rule $\mathsf{R} \in \mathit{ground}(\mathrm{P})$ whose head is p, either

(1) Some (positive or negative) subgoal in the rule body of R is false in $\mathcal{I}$; or

(2) Some positive subgoal in the rule body of R belongs to A.

[1] We slight depart from the syntax of three-valued interpretations in [18].

Intuitively, the interpretation $\mathcal{I}$ in the above definition can be considered as the (partial) information that is known about the intended model of P. On one hand, rules satisfying condition (1) are not usable for further derivations since their hypotheses are already known to be false. On the other hand, condition (2) is the unfoundedness condition: among all the rules that might still be usable to derive something in the set A, each requires an atom in A be true — a *deadlock* situation. In other words, there is no one atom in A which can be *first* established as true by the rules of P (starting from "knowing" $\mathcal{I}$). Consequently, if we choose to infer that some or all atoms in A are false, there is no way we could later infer that one in A is true. Essentially, under the well-founded semantics, all atoms in an unfounded set are *simultaneously* inferred to be false.

We can define a union operator, $\cup$, an intersection operator, $\cap$, and a partial order, $\preceq$, on three-valued interpretations. Let $\mathcal{I}_1 = \langle \mathrm{P}_1; \mathrm{Q}_1 \rangle$ and $\mathcal{I}_2 = \langle \mathrm{P}_2; \mathrm{Q}_2 \rangle$ be two three-valued interpretations (in two-set notation), where P_1, P_2 are sets of atoms that are true and Q_1, Q_2 are sets of atoms that are false. Then $\mathcal{I}_1 \cup \mathcal{I}_2 = \langle \mathrm{P}_1 \cup \mathrm{P}_2; \mathrm{Q}_1 \cup \mathrm{Q}_2 \rangle$. Similarly, $\mathcal{I}_1 \cap \mathcal{I}_2 = \langle \mathrm{P}_1 \cap \mathrm{P}_2; \mathrm{Q}_1 \cap \mathrm{Q}_2 \rangle$. Finally, $\mathcal{I}_1 \preceq \mathcal{I}_2$ iff $\mathrm{P}_1 \subseteq \mathrm{P}_2$ and $\mathrm{Q}_1 \subseteq \mathrm{Q}_2$.

Clearly, given a general logic program P, the set of three-valued interpretations, whose elements are atoms in $\mathcal{HB}_\mathrm{P}$, constitutes a complete lattice with the partial order $\preceq$. For a set of three-valued interpretations, its least upper bound can be computed using the $\cup$ operator while its greatest lower bound can be computed using the $\cap$ operator.

Next we introduce the operators that are needed to define well-founded models.

Definition 4.2.2 The operators $\mathbf{T}_\mathrm{P}$, $\mathbf{U}_\mathrm{P}$, and $\mathbf{W}_\mathrm{P}$ are defined for a general logic program P. Both $\mathbf{T}_\mathrm{P}$ and $\mathbf{U}_\mathrm{P}$ take a three-valued interpretation as input and generate a set of atoms. The operator $\mathbf{W}_\mathrm{P}$ takes a three-value interpretation as input and generates a new three-valued interpretation as follows:

$$
\begin{aligned}
\mathbf{W}_\mathrm{P}(\mathcal{I}) &= \langle\, \mathbf{T}_\mathrm{P}(\mathcal{I}); \mathbf{U}_\mathrm{P}(\mathcal{I}) \,\rangle, \text{ where} \\
\mathbf{T}_\mathrm{P}(\mathcal{I}) &= \left\{ p \;\middle|\; \begin{array}{l}\text{there is a ground rule } \mathsf{R} \text{ in } \mathit{ground}(\mathrm{P}) \text{ such that the head} \\ \text{of } \mathsf{R} \text{ is } p, \text{ and each subgoal in the body of } \mathsf{R} \text{ is true in } \mathcal{I}\end{array} \right\} \\
\mathbf{U}_\mathrm{P}(\mathcal{I}) &= \text{the } \textit{greatest} \text{ unfounded set of P with respect to } \mathcal{I}
\end{aligned}
$$

Note that in the three-valued interpretation which is returned by $\mathbf{W}_\mathrm{P}(\mathcal{I})$, the set $\mathbf{T}_\mathrm{P}(\mathcal{I})$ contains those atoms that are true, whereas $\mathbf{U}_\mathrm{P}(\mathcal{I})$ contains those atoms that are false.

Lemma 4.2.1 $\mathbf{T}_\mathrm{P}$, $\mathbf{U}_\mathrm{P}$, and $\mathbf{W}_\mathrm{P}$ are monotonic when P is fixed.

It follows that there always exists a (unique) least fixpoint of $\mathbf{W}_\mathrm{P}$ by Proposition 4.1.1. Now we are ready to define well-founded models.

Definition 4.2.3 Let P be a general logic program. The well-founded model of P is defined as the least fixpoint of $\mathbf{W}_{\mathrm{P}}$.

Definition 4.2.4 Let P be a general logic program and α range over all countable ordinals. The interpretations $\mathcal{I}_\alpha$ and $\mathcal{I}_\infty$, whose elements are atoms in $\mathcal{HB}_{\mathrm{P}}$, are defined as follows:

$$\begin{array}{lcll} \mathcal{I}_0 & = & \langle\, \emptyset; \emptyset \,\rangle & \text{for limit ordinal } 0 \\ \mathcal{I}_\alpha & = & \mathbf{W}_{\mathrm{P}}(\mathcal{I}_{\alpha-1}) & \text{for successor ordinal } \alpha \\ \mathcal{I}_\alpha & = & \bigcup\limits_{\beta<\alpha} \mathcal{I}_\beta & \text{for limit ordinal } \alpha \neq 0 \\ \mathcal{I}_\infty & = & \bigcup\limits_{\alpha} \mathcal{I}_\alpha & \end{array}$$

Then it follows that $\mathbf{W}_{\mathrm{P}} = \mathcal{I}_\infty$, by Proposition 4.1.3. In other words, $\mathcal{I}_\infty$ is equivalent to the well-founded model.

Next we will present a different characterization of the well-founded semantics, which is based on the alternating fixpoint computation introduced by Van Gelder [17]. Note that in contrast to the previous characterization which is defined in terms of the set of atoms that are true and the set of atoms that are false, this characterization is defined in terms of the set of atoms that are true and the set of atoms that are undefined.

Definition 4.2.5 Let P be a general logic program and I be a subset of $\mathcal{HB}_{\mathrm{P}}$. The operator $\mathbf{C}_{\mathrm{P,I}}$ takes as input a set of atoms, J, and generates another set of atoms.

$$\mathbf{C}_{\mathrm{P,I}}(\mathrm{J}) = \left\{ \mathsf{H} \;\middle|\; \begin{array}{l} \text{There is } \mathsf{H} \longleftarrow \mathsf{A}_1, \ldots, \mathsf{A}_\mathsf{m}, \neg\, \mathsf{B}_1, \ldots, \neg\, \mathsf{B}_\mathsf{n} \in \mathit{ground}(\mathrm{P}), \\ \mathsf{m} \geq 0, \mathsf{n} \geq 0, \mathsf{A}_\mathsf{i}\ (1 \leq \mathsf{i} \leq \mathsf{m}) \text{ and } \mathsf{B}_\mathsf{j}\ (1 \leq \mathsf{j} \leq \mathsf{n}) \text{ are positive literals, and } \mathsf{A}_\mathsf{i} \in \mathrm{J} \text{ for all } 1 \leq \mathsf{i} \leq \mathsf{m},\ \mathsf{B}_\mathsf{j} \notin \mathrm{I} \text{ for all} \\ 1 \leq \mathsf{j} \leq \mathsf{n}. \end{array} \right\}$$

Lemma 4.2.2 $\mathbf{C}_{\mathrm{P,I}}$ is monotonic when P and I are fixed.

It follows that $\mathbf{C}_{\mathrm{P,I}}$ has a unique least fixpoint. Having defined $\mathbf{C}_{\mathrm{P,I}}$ we can introduce two more operators.

Definition 4.2.6 Let P be a general logic program and I be a subset of $\mathcal{HB}_{\mathrm{P}}$. The operators $\mathbf{S}_{\mathrm{P}}$ and $\mathbf{A}_{\mathrm{P}}$ are defined as follows:

$$\begin{array}{rcl} \mathbf{S}_{\mathrm{P}}(\mathrm{I}) & \stackrel{\mathrm{def}}{=} & \mathrm{lfp}(\mathbf{C}_{\mathrm{P,I}}) \\ \mathbf{A}_{\mathrm{P}}(\mathrm{I}) & \stackrel{\mathrm{def}}{=} & \mathbf{S}_{\mathrm{P}}(\mathbf{S}_{\mathrm{P}}(\mathrm{I})) \end{array}$$

Lemma 4.2.3 $\mathbf{S}_{\mathrm{P}}$ is antimonotonic and $\mathbf{A}_{\mathrm{P}}$ is monotonic when P is fixed.

It follows that $\mathbf{A}_{\mathrm{P}}$ has a unique least fixpoint, $\mathrm{lfp}(\mathbf{A}_{\mathrm{P}})$. The following proposition gives a different characterization of the well-founded semantics.

Proposition 4.2.4 The well-founded model, $\langle\, \mathrm{T}; \mathrm{U} \,\rangle$, of a general logic program P, where T is the set of atoms that are true and U is the set of atoms that are false, can be computed as follows:

$$\begin{aligned} \mathrm{T} &= \mathrm{lfp}(\mathbf{A}_{\mathrm{P}}) \\ \mathrm{U} &= \mathbf{S}_{\mathrm{P}}(\mathrm{lfp}(\mathbf{A}_{\mathrm{P}})) - \mathrm{lfp}(\mathbf{A}_{\mathrm{P}}) \end{aligned}$$

In the sequel, we will frequently refer to this characterization of the well-founded semantics that is based on the alternating fixpoint computation.

Chapter 5

Three-Valued Semantics

In this chapter, we will first introduce the syntax of our simple, rule-based query language, which is a subset of the original F-logic language but powerful enough to specify class hierarchies and multivalued method definitions. Then we will develop a three-valued semantics for F-logic programs written in this language.

5.1 Syntax

To develop our model theory for value and code inheritance, here we focus on a small subset of F-logic, which includes only three kinds of atoms: those that represent class memberships, those that represent subclass relationships, and those that represent multivalued method specifications.

An atom of the form $\mathsf{o}:\mathsf{c}$ says that o is a member of the class c, while $\mathsf{s}::\mathsf{c}$ says that s is a subclass of c (so c is a superclass of s, but not necessarily an immediate superclass of s), and $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$ specifies that s has a multivalued method, m, whose return value is a set, and v is one of the members in that set. If s represents a class, then $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$ represents an *inheritable*[1] multivalued class method specification (*i.e.*, the value of this method can be inherited by all members of this class). If s represents an object, then $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$ represents a multivalued object method specification.

The symbols o, c, s, m, and v in the above atomic formulas are first-order terms that represent object IDs. Moreover, the terms that represent these entities in a program can contain variables and thus they can represent multiple objects, one per variable instantiation. This design makes meta-programming in F-logic as natural as querying.

[1] Note that we slightly depart from the syntax of F-logic and use $\twoheadrightarrow$ instead of $\star\twoheadrightarrow$ to represent inheritable multivalued class methods.

Let A be any atom. A literal of the form A is called a *positive* literal while a literal of the form $\neg\,\mathsf{A}$ is called a *negative* literal. An F-logic program is a finite set of rules where all variables are universally quantified. There are two kinds of rules: V-rules and C-rules. In general, V-rules represent definitions for class memberships, subclass relationships, inheritable class methods, and object methods, while C-rules represent instance method definitions only.

A V-rule has the following form:

$$\forall(\mathsf{H} \longleftarrow \mathsf{L}_1 \ \wedge \ \ldots \ \wedge \ \mathsf{L}_\mathsf{n})$$

where $\mathsf{n} \geq 0$, H is a positive literal, and L_i $(0 \leq \mathsf{i} \leq \mathsf{n})$ is either a positive or a negative literal. H is called the *head* of the rule and can be any positive F-logic literal. The *conjunction* of L_i's is called the *body* of the rule. The symbol $\forall$ indicates that all variables appearing in this rule are universally quantified. Following the standard convention, we will omit universal quantifiers in the rules and simply write

$$\mathsf{H} \longleftarrow \mathsf{L}_1, \ldots, \mathsf{L}_\mathsf{n}$$

A C-rule represents a piece of *code* that specifies an instance method definition. A C-rule has the following form:

$$\mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{L}_1, \ldots, \mathsf{L}_\mathsf{n}$$

It is similar to a V-rule except that a C-rule is marked with the special keyword code and its rule head must be a multivalued method specification. Given the above C-rule, we will say that it specifies the *instance method* m for the class c.

We will use uppercase names to denote variables and lowercase names to denote constants. A rule with an empty body is called a *fact*. So a V-rule with an empty body is called a V-fact and a C-rule with an empty body is called a C-fact. When writing down the facts, we will omit the implication symbol and simply show the head.

5.2 Three-Valued Interpretations

As illustrated by the motivating examples in Section 2.5, inheritance candidacy can be invalidated by a subsequent derivation, which suggests the use of the stable model semantics [19] or the well-founded semantics [18]. In this dissertation we adopt the latter. Since well-founded models are three-valued and the original F-logic models were two-valued [32], we need to define a suitable three-valued semantics for F-logic programs first.

The *Herbrand universe* of an F-logic program P, denoted $\mathcal{HU}_{\mathrm{P}}$, consists of all the *ground* (*i.e.*, variable-free) terms constructed using the function symbols and constants found in the program.

The *Herbrand instantiation* of an F-logic program P, denoted *ground*(P), is the set of rules obtained by consistently substituting all the terms in $\mathcal{HU}_{\mathrm{P}}$ for all variables in every rule of P. Although the program P is finite, its Herbrand instantiation may well be infinite. For the semantics to be discussed in this dissertation, we only consider the Herbrand instantiation of a program.

The *Herbrand base* of an F-logic program P, denoted $\mathcal{HB}_{\mathrm{P}}$, consists of the following sorts of atoms: $\mathsf{o}:\mathsf{c}$, $\mathsf{s}::\mathsf{c}$, $\mathsf{s}[\mathsf{m}\longrightarrow\mathsf{v}]^{\mathsf{s}}_{\mathrm{local}}$, $\mathsf{o}[\mathsf{m}\longrightarrow\mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}$, and $\mathsf{o}[\mathsf{m}\longrightarrow\mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}$, where o, c, s, m, and v are terms from $\mathcal{HU}_{\mathrm{P}}$.

A *three-valued interpretation* $\mathcal{I}$ of an F-logic program P is a pair $\langle\,\mathrm{T};\mathrm{U}\,\rangle$, where T and U are *disjoint* subsets of the Herbrand base $\mathcal{HB}_{\mathrm{P}}$ of P. The set T contains all atoms that are *true* in $\mathcal{I}$ and U contains all atoms that are *undefined* in $\mathcal{I}$. The set F of all atoms that are *false* in $\mathcal{I}$ is defined as $\mathrm{F}=\mathcal{HB}_{\mathrm{P}}-(\mathrm{T}\cup\mathrm{U})$. A three-valued interpretation $\mathcal{I}=\langle\,\mathrm{T};\mathrm{U}\,\rangle$ is called *two-valued* if $\mathrm{U}=\emptyset$.

5.3 Truth Valuation Functions

Following [44, 45], we will define the truth valuation functions for atoms, literals, and V-rules. The atoms in $\mathcal{HB}_{\mathrm{P}}$ can take one of the three values: $\mathbf{t}$, $\mathbf{f}$, and $\mathbf{u}$. Note that the truth value $\mathbf{u}$ means possibly true or possible false and so carries more "truth" than the truth value $\mathbf{f}$. Therefore, the ordering among truth values is defined as follows: $\mathbf{f}\ <\mathbf{u}\ <\mathbf{t}$.

Given an interpretation $\mathcal{I}=\langle\,\mathrm{T};\mathrm{U}\,\rangle$ of an F-logic program P, for any atom A from $\mathcal{HB}_{\mathrm{P}}$ we can define a truth valuation function $\mathcal{I}$ as follows:

$$\mathcal{I}(\mathsf{A})=\begin{cases}\mathbf{t}, & \text{if } \mathsf{A}\in\mathrm{T};\\ \mathbf{u}, & \text{if } \mathsf{A}\in\mathrm{U};\\ \mathbf{f}, & \text{otherwise.}\end{cases}$$

Moreover, for any $\mathsf{A}_{\mathsf{i}}\in\mathcal{HB}_{\mathrm{P}}$, $1\leq\mathsf{i}\leq\mathsf{n}$:

$$\mathcal{I}(\mathsf{A}_1\ \wedge\ \ldots\ \wedge\ \mathsf{A}_{\mathsf{n}})=min\{\mathcal{I}(\mathsf{A}_{\mathsf{i}})|1\leq\mathsf{i}\leq\mathsf{n}\}$$

Next we will extend the truth valuation function $\mathcal{I}$ to all V-rules in the Herbrand instantiation, *ground*(P), of P.

In an interpretation of an F-logic program, atoms of the form $\mathsf{s}[\mathsf{m}\longrightarrow\mathsf{v}]^{\mathsf{s}}_{\mathrm{local}}$ capture the idea that $\mathsf{m}\longrightarrow\mathsf{v}$ is locally defined at s via a V-rule, while atoms of the forms

$\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}$, where $\mathsf{o} \neq \mathsf{c}$, capture the idea that o inherits $\mathsf{m} \twoheadrightarrow \mathsf{v}$ from c by value and code inheritance, respectively.

Generally, the intuitive reading of a V-rule is as follows: the head of the rule functions as a *local definition* while the body of the rule functions as a *query*. In particular, if $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$ is in the head of a rule and the body of the rule is satisfied, it means that $\mathsf{m} \twoheadrightarrow \mathsf{v}$ is *locally defined* for s. If $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$ appears in the body of a rule, it is a query which tests whether s has a local definition of $\mathsf{m} \twoheadrightarrow \mathsf{v}$, or s inherits $\mathsf{m} \twoheadrightarrow \mathsf{v}$ from some superclass by either value or code inheritance.

Therefore, the truth valuation of a ground F-logic literal may be different depending on whether this literal appears in a rule head or in a rule body. The following definitions make the above discussion precise.

Definition 5.3.1 Given an interpretation $\mathcal{I}$ of an F-logic program P, the *truth valuation functions* $\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}$ and $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}$ (**h** stands for *head* and **b** for *body*) on ground F-logic literals are defined as follows:

$$\begin{aligned}
\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{o}:\mathsf{c}) &= \mathcal{I}(\mathsf{o}:\mathsf{c}) \\
\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{s}::\mathsf{c}) &= \mathcal{I}(\mathsf{s}::\mathsf{c}) \\
\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) &= \mathcal{I}(\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{s}}_{\text{local}}) \\
\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o}:\mathsf{c}) &= \mathcal{I}(\mathsf{o}:\mathsf{c}) \\
\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{s}::\mathsf{c}) &= \mathcal{I}(\mathsf{s}::\mathsf{c}) \\
\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) &= max \left\{ \begin{array}{l} \mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\text{local}}) \\ \mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) \\ \mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) \end{array} \;\middle|\; \mathsf{c} \in \mathcal{HU}_{\mathrm{P}} \right\}
\end{aligned}$$

Let L and L_i $(1 \leq \mathsf{i} \leq \mathsf{n})$ be variable-free literals. Then

$$\begin{aligned}
\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{L}) &= \neg\, \mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) \\
\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_1 \wedge \ldots \wedge \mathsf{L}_\mathsf{n}) &= min\{\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_\mathsf{i}) \mid 1 \leq \mathsf{i} \leq \mathsf{n}\}
\end{aligned}$$

where $\neg\, \mathbf{f} = \mathbf{t}$, $\neg\, \mathbf{u} = \mathbf{u}$, and $\neg\, \mathbf{t} = \mathbf{f}$.

We have the the following two lemmas regarding some properties of the truth valuation function $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}$.

Lemma 5.3.1 Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be an interpretation of an F-logic program P, L be a ground literal in *ground*(P), $\mathcal{J} = \langle\, \mathrm{T}; \emptyset \,\rangle$, and $\mathcal{K} = \langle\, \mathrm{T} \cup \mathrm{U}; \emptyset \,\rangle$:

(1) If L is a positive literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$ iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}) = \mathbf{t}$.

(2) If L is a negative literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$ iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{L}) = \mathbf{t}$.

(3) If L is a positive literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) \geq \mathbf{u}$ iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{L}) = \mathbf{t}$.

(4) If L is a negative literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) \geq \mathbf{u}$ iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}) = \mathbf{t}$.

Proof.

(1) If L is a positive literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$ iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}) = \mathbf{t}$.

If $\mathsf{L} = \mathsf{o} : \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o} : \mathsf{c}) = \mathbf{t}$, iff $\mathcal{I}(\mathsf{o} : \mathsf{c}) = \mathbf{t}$, iff $\mathsf{o} : \mathsf{c} \in \mathrm{T}$, iff $\mathcal{J}(\mathsf{o} : \mathsf{c}) = \mathbf{t}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{o} : \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \mathsf{s} :: \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{s} :: \mathsf{c}) = \mathbf{t}$, iff $\mathcal{I}(\mathsf{s} :: \mathsf{c}) = \mathbf{t}$, iff $\mathsf{s} :: \mathsf{c} \in \mathrm{T}$, iff $\mathcal{J}(\mathsf{s} :: \mathsf{c}) = \mathbf{t}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{s} :: \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$, iff $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\mathrm{local}}) = \mathbf{t}$ or there exists c such that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}) = \mathbf{t}$ or $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{t}$, iff $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\mathrm{local}} \in \mathrm{T}$ or there exists c such that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \mathrm{T}$ or $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathrm{T}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$.

(2) If L is a negative literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$ iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{L}) = \mathbf{t}$.

If $\mathsf{L} = \neg\, \mathsf{o} : \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{o} : \mathsf{c}) = \mathbf{t}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o} : \mathsf{c}) = \mathbf{f}$, iff $\mathcal{I}(\mathsf{o} : \mathsf{c}) = \mathbf{f}$, iff $\mathsf{o} : \mathsf{c} \notin \mathrm{T} \cup \mathrm{U}$, iff $\mathcal{K}(\mathsf{o} : \mathsf{c}) = \mathbf{f}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{o} : \mathsf{c}) = \mathbf{f}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\neg\, \mathsf{o} : \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \neg\, \mathsf{s} :: \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{s} :: \mathsf{c}) = \mathbf{t}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{s} :: \mathsf{c}) = \mathbf{f}$, iff $\mathcal{I}(\mathsf{s} :: \mathsf{c}) = \mathbf{f}$, iff $\mathsf{s} :: \mathsf{c} \notin \mathrm{T} \cup \mathrm{U}$, iff $\mathcal{K}(\mathsf{s} :: \mathsf{c}) = \mathbf{f}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{s} :: \mathsf{c}) = \mathbf{f}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\neg\, \mathsf{s} :: \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \neg\, \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{f}$, iff $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\mathrm{local}}) = \mathbf{f}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}) = \mathbf{f}$, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{f}$ for all c, iff $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\mathrm{local}} \notin \mathrm{T} \cup \mathrm{U}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \notin \mathrm{T} \cup \mathrm{U}$, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \notin \mathrm{T} \cup \mathrm{U}$ for all c, iff $\mathcal{K}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\mathrm{local}}) = \mathbf{f}$ and $\mathcal{K}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}) = \mathbf{f}$, $\mathcal{K}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{f}$ for all c, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{f}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\neg\, \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$.

(3) If L is a positive literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) \geq \mathbf{u}$ iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{L}) = \mathbf{t}$.

If $\mathsf{L} = \mathsf{o} : \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o} : \mathsf{c}) \geq \mathbf{u}$, iff $\mathcal{I}(\mathsf{o} : \mathsf{c}) \geq \mathbf{u}$, iff $\mathsf{o} : \mathsf{c} \in \mathrm{T} \cup \mathrm{U}$, iff $\mathcal{K}(\mathsf{o} : \mathsf{c}) = \mathbf{t}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{o} : \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \mathsf{s} :: \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{s} :: \mathsf{c}) \geq \mathbf{u}$, iff $\mathcal{I}(\mathsf{s} :: \mathsf{c}) \geq \mathbf{u}$, iff $\mathsf{s} :: \mathsf{c} \in \mathrm{T} \cup \mathrm{U}$, iff $\mathcal{K}(\mathsf{s} :: \mathsf{c}) = \mathbf{t}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{s} :: \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) \geq \mathbf{u}$, iff $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\mathrm{local}}) \geq \mathbf{u}$ or there exists c such that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}) \geq \mathbf{u}$ or $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) \geq \mathbf{u}$, iff $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\mathrm{local}} \in \mathrm{T} \cup \mathrm{U}$ or there exists c such that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \mathrm{T} \cup \mathrm{U}$ or $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathrm{T} \cup \mathrm{U}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$.

(4) If L is a negative literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) \geq \mathbf{u}$ iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}) = \mathbf{t}$.

If $\mathsf{L} = \neg\, \mathsf{o} : \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{o} : \mathsf{c}) \geq \mathbf{u}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o} : \mathsf{c}) = \mathcal{I}(\mathsf{s} : \mathsf{c}) \leq \mathbf{u}$, iff $\mathsf{o} : \mathsf{c} \notin \mathrm{T}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{o} : \mathsf{c}) = \mathcal{J}(\mathsf{o} : \mathsf{c}) = \mathbf{f}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\neg\, \mathsf{o} :: \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \neg\, \mathsf{s} :: \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{s} :: \mathsf{c}) \geq \mathbf{u}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{s} :: \mathsf{c}) = \mathcal{I}(\mathsf{s} :: \mathsf{c}) \leq \mathbf{u}$, iff $\mathsf{s} :: \mathsf{c} \notin \mathrm{T}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{s} :: \mathsf{c}) = \mathcal{J}(\mathsf{s} :: \mathsf{c}) = \mathbf{f}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\neg\, \mathsf{s} :: \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \neg\, \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) \geq \mathbf{u}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) \leq \mathbf{u}$, iff $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{o}}_{\mathrm{local}}) \leq \mathbf{u}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{value}}) \leq \mathbf{u}$, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{code}}) \leq \mathbf{u}$ for all c, iff $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{o}}_{\mathrm{local}} \notin \mathrm{T}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{value}} \notin \mathrm{T}$, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{code}} \notin \mathrm{T}$ for all c, iff $\mathcal{J}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{o}}_{\mathrm{local}}) = \mathbf{f}$ and $\mathcal{J}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{value}}) = \mathbf{f}$, $\mathcal{J}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{f}$ for all c, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{f}$, iff $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\neg\, \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$.

□

Lemma 5.3.2 Let $\mathcal{I} = \langle\, \mathrm{A}; \emptyset \,\rangle$ and $\mathcal{J} = \langle\, \mathrm{B}; \emptyset \,\rangle$ be interpretations of an F-logic program P, $\mathrm{A} \subseteq \mathrm{B}$, and L be a ground literal in *ground*(P):

(1) If L is a positive literal and $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}) = \mathbf{t}$.

(2) If L is a negative literal and $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}) = \mathbf{t}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$.

Proof.

(1) If L is a positive literal and $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}) = \mathbf{t}$.

If $\mathsf{L} = \mathsf{o} : \mathsf{c}$, then $\mathsf{o} : \mathsf{c} \in \mathrm{A} \subseteq \mathrm{B}$. So $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{o} : \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \mathsf{s} :: \mathsf{c}$, then $\mathsf{s} :: \mathsf{c} \in \mathrm{A} \subseteq \mathrm{B}$. So $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{s} :: \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$, then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{o}}_{\mathrm{local}}) = \mathbf{t}$ or there exists c such that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{value}}) = \mathbf{t}$ or $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{t}$. Thus $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{o}}_{\mathrm{local}} \in \mathrm{A} \subseteq \mathrm{B}$ or there exists c such that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{value}} \in \mathrm{A} \subseteq \mathrm{B}$ or $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{code}} \in \mathrm{A} \subseteq \mathrm{B}$. It follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$.

(2) If L is a negative literal and $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}) = \mathbf{t}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$.

If $\mathsf{L} = \neg\, \mathsf{o} : \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{o} : \mathsf{c}) = \mathbf{f}$. Therefore, $\mathsf{o} : \mathsf{c} \notin \mathrm{B}$. So $\mathsf{o} : \mathsf{c} \notin \mathrm{A}$ and $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o} : \mathsf{c}) = \mathbf{f}$. It follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{o} : \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \neg\, \mathsf{s} :: \mathsf{c}$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{s} :: \mathsf{c}) = \mathbf{f}$. So $\mathsf{s} :: \mathsf{c} \notin \mathrm{B}$ and $\mathsf{s} :: \mathsf{c} \notin \mathrm{A}$. Thus $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{s} :: \mathsf{c}) = \mathbf{f}$. It follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{s} :: \mathsf{c}) = \mathbf{t}$.

If $\mathsf{L} = \neg\, \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{f}$. Thus $\mathcal{J}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{o}}_{\mathrm{local}}) = \mathbf{f}$ and $\mathcal{J}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{value}}) = \mathbf{f}$, $\mathcal{J}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{f}$ for all c. It follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{o}}_{\mathrm{local}} \notin \mathrm{B}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{value}} \notin \mathrm{B}$, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{code}} \notin \mathrm{B}$ for all c. Therefore, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{o}}_{\mathrm{local}} \notin \mathrm{A}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{value}} \notin \mathrm{A}$, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]\,^{\mathsf{c}}_{\mathrm{code}} \notin \mathrm{A}$ for all c. So $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{f}$ and $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\neg\, \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$.

□

5.4 V-Rule Satisfaction

With the definitions of $\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}$ and $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}$, we can define the truth valuation function $\mathcal{I}$ on ground V-rules. We should note that although the truth valuation function $\mathcal{I}$ is

three-valued when applied to ground atoms, it becomes two-valued when applied to ground V-rules. Intuitively, a ground V-rule is evaluated to be true if and only if the truth value of rule head is greater than or equal to the truth value of the rule body. Formally, we have the following definition.

Definition 5.4.1 Given an interpretation $\mathcal{I}$ of an F-logic program P, the truth valuation function $\mathcal{I}$ on a ground V-rule, $\mathsf{H} \longleftarrow \mathsf{B}$, in *ground*(P), is defined as follows:

$$\mathcal{I}(\mathsf{H} \longleftarrow \mathsf{B}) = \begin{cases} \mathbf{t}, & \text{if } \mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) \geq \mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}); \\ \mathbf{f}, & \text{otherwise.} \end{cases}$$

And the truth valuation function $\mathcal{I}$ on a ground V-fact, H, in *ground*(P), is defined as follows:

$$\mathcal{I}(\mathsf{H}) = \begin{cases} \mathbf{t}, & \text{if } \mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) = \mathbf{t}; \\ \mathbf{f}, & \text{otherwise.} \end{cases}$$

We will say that a three-valued interpretation satisfies the V-rules of an F-logic program, if it satisfies all the ground V-rules of this program.

Definition 5.4.2 (V-Rule Satisfaction) A three-valued interpretation $\mathcal{I}$ *satisfies* the V-rules of an F-logic program P, if for every V-rule R in *ground*(P), $\mathcal{I}(\mathsf{R}) = \mathbf{t}$.

Chapter 6

Inheritance Postulates

Even if an interpretation $\mathcal{I}$ satisfies all the V-rules of an F-logic program P, it does not necessarily mean that $\mathcal{I}$ is an intended *object model* of P, because $\mathcal{I}$ must also include facts that are derived by inheritance. F-logic programs specify only class hierarchies and method definitions — what needs to be inherited is not explicitly stated. In fact, as we saw in Section 2.5, defining exactly what should be inherited is a subtle issue. In our framework, it is the job of the *inheritance postulates*, which embody the common intuition behind nonmonotonic multiple inheritance. The purpose of this chapter is to define these postulates and the associated notion of an object model.

Intuitively, $\mathsf{c}[\mathsf{m}]$ is an *inheritance context* for o, if o is a member of the class c, and $\mathsf{m} \twoheadrightarrow \mathsf{v}$ is locally defined at c for some value v (*i.e.*, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$ is defined as a fact or derived via a V-rule) or there is a C-rule which specifies the instance method m for the class c. Inheritance context is necessary for inheritance to take place, but is not sufficient. Indeed, inheritance of the values of m from c might be overridden by a more specific inheritance context that sits below c along the inheritance path. If an inheritance context is not overridden by any other inheritance context, then we call it an *inheritance candidate*. Inheritance candidates represent potential sources for inheritance. But there must be exactly one inheritance candidate for inheritance to take place — having more just leads to a multiple inheritance conflict.

The various concepts to be defined in this chapter come in with two flavors: *strong* or *weak*. The "strong" flavor of a concept requires that all relevant facts be positively established while the "weak" flavor allows some or all facts to be undefined.

6.1 Inheritance Candidates

Definition 6.1.1 (Local Context) Given an interpretation $\mathcal{I}$ of an F-logic program P, $\mathsf{s}[\mathsf{m}]$ is a *strong local context*, if $max\{\mathcal{I}(\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{s}}_{\mathrm{local}}) | \mathsf{v} \in \mathcal{HU}_{\mathrm{P}}\} = \mathbf{t}$. Similarly, $\mathsf{s}[\mathsf{m}]$ is a *weak local context* if $max\{\mathcal{I}(\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{s}}_{\mathrm{local}}) | \mathsf{v} \in \mathcal{HU}_{\mathrm{P}}\} = \mathbf{u}$.

Definition 6.1.2 (Value Inheritance Context) Given an interpretation $\mathcal{I}$ of an F-logic program P, $\mathsf{c}[\mathsf{m}]$ is a *strong value inheritance context* for o, if $\mathsf{c} \neq \mathsf{o}$[1] and $min\{\mathcal{I}(\mathsf{o}:\mathsf{c}), max\{\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{local}} | \mathsf{v} \in \mathcal{HU}_{\mathrm{P}}\}\} = \mathbf{t}$. (*i.e.*, the object o is a proper member of the class c and $\mathsf{c}[\mathsf{m}]$ is a strong local context). Similarly, $\mathsf{c}[\mathsf{m}]$ is a *weak value inheritance context* for o if $\mathsf{c} \neq \mathsf{o}$ and $min\{\mathcal{I}(\mathsf{o}:\mathsf{c}), max\{\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{local}} | \mathsf{v} \in \mathcal{HU}_{\mathrm{P}}\}\} = \mathbf{u}$.

Definition 6.1.3 (Code Inheritance Context) Given an interpretation $\mathcal{I}$ of an F-logic program P, $\mathsf{c}[\mathsf{m}]$ is a *strong code inheritance context* for o, if $\mathsf{c} \neq \mathsf{o}$, $\mathcal{I}(\mathsf{o}:\mathsf{c}) = \mathbf{t}$, and there is a C-rule in P which specifies the instance method m for the class c. Similarly, $\mathsf{c}[\mathsf{m}]$ is a *weak code inheritance context* for o if $\mathsf{c} \neq \mathsf{o}$, $\mathcal{I}(\mathsf{o}:\mathsf{c}) = \mathbf{u}$, and there is a C-rule in P which specifies the instance method m for the class c.

Note that local contexts can only be established via V-rules but not C-rules. The difference between a value and a code inheritance context is that the former requires at least one value be established for its class method via a V-rule, whereas the latter only requires the presence of at least one C-rule which specifies its instance method.

Therefore, if inheritance takes place from a value inheritance context, it is the values of this class method that will be directly inherited by its members. On the contrary, when inheritance takes place from a code inheritance context, it is the definitions of this instance method that will be inherited. Furthermore, these definitions will be evaluated in the context of individual members, which does not necessarily entail that a value of this instance method be derived for a member of this class.

When the difference between value and code inheritance is not important, we will generally use the term *inheritance context* to refer to either a value or a code inheritance context. In the following definitions we will see that value and code inheritance contexts are treated equally as far as overriding is concerned.

Definition 6.1.4 (Overriding) Given an interpretation $\mathcal{I}$ of an F-logic program P, the class s *strongly overrides* $\mathsf{c}[\mathsf{m}]$ for o, if $\mathsf{s} \neq \mathsf{c}$, $\mathcal{I}(\mathsf{s}::\mathsf{c}) = \mathbf{t}$, and $\mathsf{s}[\mathsf{m}]$ is either a strong value inheritance context or a strong code inheritance context for o.

The class s *weakly overrides* $\mathsf{c}[\mathsf{m}]$ for o if the above conditions are relaxed by allowing $\mathsf{s}::\mathsf{c}$ to be undefined and/or allowing $\mathsf{s}[\mathsf{m}]$ to be a weak inheritance context. Formally this means that either

(1) $\mathcal{I}(\mathsf{s}::\mathsf{c}) = \mathbf{t}$ and $\mathsf{s}[\mathsf{m}]$ is a weak inheritance context for o; or

[1] $\mathsf{c} \neq \mathsf{o}$ means that c and o are distinct terms.

(2) $\mathcal{I}(\mathsf{s} :: \mathsf{c}) = \mathbf{u}$ and $\mathsf{s}[\mathsf{m}]$ is either a weak or a strong inheritance context for o.

Definition 6.1.5 (Value Inheritance Candidate) Given an interpretation $\mathcal{I}$ of an F-logic program P, $\mathsf{c}[\mathsf{m}]$ is a *strong value inheritance candidate* for o, denoted $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, if $\mathsf{c}[\mathsf{m}]$ is a strong value inheritance context for o, and there is no s that strongly or weakly overrides $\mathsf{c}[\mathsf{m}]$ for o.

$\mathsf{c}[\mathsf{m}]$ is a *weak value inheritance candidate* for o, denoted $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, if the above conditions are relaxed by allowing $\mathsf{c}[\mathsf{m}]$ to be a weak value inheritance context and/or allowing weak overriding. Formally, this means that there is no s that strongly overrides $\mathsf{c}[\mathsf{m}]$ for o, and either

(1) $\mathsf{c}[\mathsf{m}]$ is a weak value inheritance context for o; or

(2) $\mathsf{c}[\mathsf{m}]$ is a strong value inheritance context for o and there is s that weakly overrides $\mathsf{c}[\mathsf{m}]$ for o.

Definition 6.1.6 (Code Inheritance Candidate) Given an interpretation $\mathcal{I}$ of an F-logic program P, $\mathsf{c}[\mathsf{m}]$ is a *strong code inheritance candidate* for o, denoted $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, if $\mathsf{c}[\mathsf{m}]$ is a strong code inheritance context for o, and there is no s that strongly or weakly overrides $\mathsf{c}[\mathsf{m}]$ for o.

$\mathsf{c}[\mathsf{m}]$ is a *weak code inheritance candidate* for o, denoted $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, if the above conditions are relaxed by allowing $\mathsf{c}[\mathsf{m}]$ to be a weak code inheritance context and/or allowing weak overriding. Formally, this means that there is no s that strongly overrides $\mathsf{c}[\mathsf{m}]$ for o, and either

(1) $\mathsf{c}[\mathsf{m}]$ is a weak code inheritance context for o; or

(2) $\mathsf{c}[\mathsf{m}]$ is a strong code inheritance context for o and there is s that weakly overrides $\mathsf{c}[\mathsf{m}]$ for o.

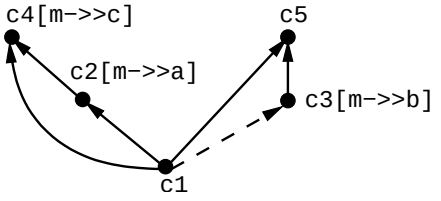
Example 6.1.1 As an example, consider an interpretation $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ of an F-logic program P, where

$$\begin{aligned} \mathrm{T} &= \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_1 : \mathsf{c}_4, \mathsf{c}_1 : \mathsf{c}_5, \mathsf{c}_2 :: \mathsf{c}_4, \mathsf{c}_3 :: \mathsf{c}_5\} \cup \\ &\quad \{\mathsf{c}_2[\mathsf{m} \longrightarrow\!\!\!\!\!\rightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{local}}, \mathsf{c}_3[\mathsf{m} \longrightarrow\!\!\!\!\!\rightarrow \mathsf{b}]^{\mathsf{c}_3}_{\mathrm{local}}, \mathsf{c}_4[\mathsf{m} \longrightarrow\!\!\!\!\!\rightarrow \mathsf{c}]^{\mathsf{c}_4}_{\mathrm{local}}\} \\ \mathrm{U} &= \{\mathsf{c}_1 : \mathsf{c}_3\} \end{aligned}$$

$\mathcal{I}$ and P are illustrated in Figure 5, where solid and dashed arrows represent true and undefined values, respectively.

In the interpretation $\mathcal{I}$, $\mathsf{c}_2[\mathsf{m}]$ and $\mathsf{c}_4[\mathsf{m}]$ are strong value inheritance contexts for c_1. $\mathsf{c}_5[\mathsf{m}]$ is a strong code inheritance context for c_1. On the other hand, $\mathsf{c}_3[\mathsf{m}]$ is a weak value inheritance context for c_1. The class c_2 strongly overrides $\mathsf{c}_4[\mathsf{m}]$, while c_3 weakly overrides $\mathsf{c}_5[\mathsf{m}]$. The context $\mathsf{c}_2[\mathsf{m}]$ is a strong value inheritance candidate for c_1, while $\mathsf{c}_3[\mathsf{m}]$ is a weak value inheritance candidate and $\mathsf{c}_5[\mathsf{m}]$ is a weak code

inheritance candidate for c_1. Finally, $c_4[m]$ is neither a strong nor a weak value inheritance candidate for c_1. □



$c_1 : c_2$.
$c_1 : c_5$.
$c_2 :: c_4$.
$c_3 :: c_5$.
$c_2[m \twoheadrightarrow a]$.
$c_3[m \twoheadrightarrow b]$.
$c_4[m \twoheadrightarrow c]$.
code $c_5[m \twoheadrightarrow v] \longleftarrow c_5[f \twoheadrightarrow x]$.

Figure 5: Inheritance Context, Overriding, and Inheritance Candidate

For convenience, we will simply write $c[m] \rightsquigarrow_{\mathcal{I}} o$ when it does not matter whether $c[m]$ is a strong or a weak value or code inheritance candidate. Now we are ready to introduce our postulates for nonmonotonic multiple value and code inheritance.

6.2 Core Inheritance Postulates

Definition 6.2.1 (Positive ISA Transitivity) An interpretation $\mathcal{I}$ of an F-logic program P satisfies the *positive ISA transitivity constraint* if the positive part of the class hierarchy is transitively closed, formally, if the following two conditions hold:

(1) for all s, c: if there is x such that $\mathcal{I}(s :: x) = \mathbf{t}$ and $\mathcal{I}(x :: c) = \mathbf{t}$, then $\mathcal{I}(s :: c) = \mathbf{t}$;

(2) for all o, c: if there is x such that $\mathcal{I}(o : x) = \mathbf{t}$ and $\mathcal{I}(x :: c) = \mathbf{t}$, then $\mathcal{I}(o : c) = \mathbf{t}$.

Definition 6.2.2 (Context Consistency) An interpretation $\mathcal{I}$ of an F-logic program P satisfies the *context consistency constraint*, if the following conditions hold:

(1) for all o, m, v: $\mathcal{I}(o[m \twoheadrightarrow v]^{o}_{\text{value}}) = \mathbf{f}$ and $\mathcal{I}(o[m \twoheadrightarrow v]^{o}_{\text{code}}) = \mathbf{f}$;

(2) for all c, m, v: if $\mathcal{I}(c[m \twoheadrightarrow v]^{c}_{\text{local}}) = \mathbf{f}$, then $\mathcal{I}(o[m \twoheadrightarrow v]^{c}_{\text{value}}) = \mathbf{f}$ for all o;

(3) for all c, m: if there is no C-rule in *ground*(P) which specifies the instance method m for the class c, then $\mathcal{I}(o[m \twoheadrightarrow v]^{c}_{\text{code}}) = \mathbf{f}$ for all o, v;

(4) for all o, m: if $o[m]$ is a strong local context, then $\mathcal{I}(o[m \twoheadrightarrow v]^{c}_{\text{value}}) = \mathbf{f}$ and $\mathcal{I}(o[m \twoheadrightarrow v]^{c}_{\text{code}}) = \mathbf{f}$ for all v, c.

The context consistency constraint captures the implications of locality and specificity. The first condition rules out self inheritance. The second condition states that if $\mathsf{m} \twoheadrightarrow \mathsf{v}$ is not locally defined at c, then no class should inherit $\mathsf{m} \twoheadrightarrow \mathsf{v}$ from c by value inheritance. The third condition states that if a class c does not specify an instance method m, then no object should inherit any value of m from c by code inheritance. The fourth condition states that if $\mathsf{m} \twoheadrightarrow \mathsf{v}$ is locally defined at o, then this definition should prevent o from inheriting any value of m from other classes.

The following constraint captures the meaning of nonmonotonic multiple value and code inheritance. Intuitively, we want our semantics for inheritance to have such a property that if inheritance is allowed, then it should take place from a *unique* source.

Definition 6.2.3 (Unique Source Inheritance) An interpretation $\mathcal{I}$ of an F-logic program P satisfies the *unique source inheritance constraint*, if the following two conditions hold:

(1) for all $\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}$: if $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{t}$ or $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) = \mathbf{t}$, then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{x}}_{\text{value}}) = \mathbf{f}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{x}}_{\text{code}}) = \mathbf{f}$ for all z, x such that $\mathsf{x} \neq \mathsf{c}$.

(2) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{value}}) = \mathbf{f}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{code}}) = \mathbf{f}$ for all v, x such that $\mathsf{x} \neq \mathsf{c}$.

(3) for all $\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}$: $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{t}$ *iff*

- (i) $\mathsf{o}[\mathsf{m}]$ is neither a strong nor a weak local context; and
- (ii) $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$; and
- (iii) $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}}) = \mathbf{t}$; and
- (iv) there is no x such that $\mathsf{x} \neq \mathsf{c}$ and $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$.

The first condition above states that if a positive value or code inheritance takes place from a class, then no value and code inheritance can take place from other classes.

The second condition states that if a strong value or code inheritance candidate, $\mathsf{c}[\mathsf{m}]$, exists, then inheritance of the method m cannot take place from any other source (because this would be a multiple inheritance conflict). However, inheritance of the method m can take place from c, if there are no other inheritance candidates and no local contexts.

The third condition specifies when "positive" value inheritance takes place. An object o *must* inherit $\mathsf{m} \twoheadrightarrow \mathsf{v}$ from a class c by value inheritance if and only if: (i) no value is locally defined for the method m at o; (ii) $\mathsf{c}[\mathsf{m}]$ is a strong value inheritance candidate for o; (iii) $\mathsf{m} \twoheadrightarrow \mathsf{v}$ is locally defined at c and is positive; and (iv) there are no other inheritance candidates — weak or strong — from which o could inherit the method m.

6.3 C-Rule Satisfaction

Intuitively, a model of an F-logic program should satisfy all the rules in this program. In Section 5.4 we have defined the truth valuation function on ground V-rules and the associated notion of V-rule satisfaction. Now we will extend the truth valuation function to ground C-rules.

Satisfaction of C-rules, however, is different from that of V-rules, because C-rules specify instance method definitions for classes. Only when these definitions are inherited by individual members of a class should they be satisfied.

When code inheritance takes place, an object inherits the instance method definitions from the class to which it belongs. Once inherited, these instance method definitions are evaluated in the context of individual objects. This corresponds to the idea of *late binding* in imperative object-oriented languages like C^{++} and Java.

In a ground C-rule which specifies an instance method m for a class c, the name c can be considered as a *placeholder* that stands for any member of the class c. When this C-rule is inherited, the name c will be substituted by the oids of individual objects that belong to this class.

Definition 6.3.1 (Binding) Let $\mathsf{R} \equiv (\mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{B})$ be a ground C-rule which specifies the instance method m for the class c. The *binding* of R with respect to o, denoted $\mathsf{R}||\mathsf{o}$, is obtained from R by substituting o for every occurrence of c in R. We will use $\mathsf{X}_{\mathsf{c}\backslash\mathsf{o}}$ to represent the term that is obtained from X by substituting o for every occurrence of c in X.

Therefore, the truth valuation function will be defined on bindings of ground C-rules instead of on C-rules directly. Intuitively, when an object inherits the C-rules from a class, the bindings of these C-rules would act like local definitions for this object and so should be satisfied similarly to V-rules. However, because only those C-rules which are inherited need to be satisfied, satisfaction of C-rules depends on how they are inherited: strongly or weakly.

Definition 6.3.2 (Strong Code Inheritance) Let $\mathcal{I}$ be an interpretation of an F-logic program P and let $\mathsf{R} \equiv (\mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{B})$ be a C-rule in *ground*(P). An object o *strongly inherits* R, if the following conditions hold:

(1) $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$;

(2) $\mathsf{o}[\mathsf{m}]$ is neither a strong nor a weak local context;

(3) there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$.

Given a ground C-rule which specifies the instance method m for the class c, we say that an object o strongly inherits R, if: (i) $\mathsf{c}[\mathsf{m}]$ is a strong code inheritance

candidate for o, *i.e.*, $\mathsf{c[m]}$ is not overridden by any intermediate classes; (ii) $\mathsf{o[m]}$ is neither a strong nor a weak local context, *i.e.*, o has no locally defined values for the method m; and (iii) there are no other strong or weak inheritance candidates, *i.e.*, there is no multiple inheritance conflict at all.

Definition 6.3.3 (Weak Code Inheritance) Let $\mathcal{I}$ be an interpretation of an F-logic program P and $\mathsf{R} \equiv \mathsf{code}\ \ \mathsf{c[m \twoheadrightarrow v]} \longleftarrow \mathsf{B}$ be a C-rule in *ground*(P). An object o *weakly inherits* R, if the following conditions hold:

(1) $\mathsf{c[m]} \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c[m]} \overset{wc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$;

(2) $\mathsf{o[m]}$ is not a strong local context;

(3) there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x[m]} \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x[m]} \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$;

(4) o does not strongly inherit R.

Given a ground C-rule which specifies the instance method m for the class c, we say that an object o weakly inherits R, if: (i) $\mathsf{c[m]}$ is either a strong or a weak code inheritance candidate for o, *i.e.*, there is no strong evidence that $\mathsf{c[m]}$ is overridden by an intermediate class; (ii) $\mathsf{o[m]}$ is not a strong local context, *i.e.*, there is no strong evidence that o has locally defined values for the method m; (iii) there are no other strong value or code inheritance candidates, *i.e.*, there is no strong evidence for a multiple inheritance conflict; and (iv) o does not strongly inherit R.

Given an interpretation $\mathcal{I}$ of an F-logic program P, let R be a C-rule in *ground*(P) and $\mathsf{R||o}$ be the binding of R with respect to o. We can define a function, $imode_{\mathcal{I}}$, on bindings of ground C-rules, which returns the "inheritance mode" of a binding:

$$imode_{\mathcal{I}}(\mathsf{R||o}) = \begin{cases} \mathbf{t}, & \text{if } \mathsf{o} \text{ strongly inherits } \mathsf{R}; \\ \mathbf{u}, & \text{if } \mathsf{o} \text{ weakly inherits } \mathsf{R}; \\ \mathbf{f}, & \text{otherwise.} \end{cases}$$

When $imode_{\mathcal{I}}(\mathsf{R||o}) = \mathbf{t}$, we will say that $\mathsf{R||o}$ is in strong code inheritance mode. Similarly, we will say $\mathsf{R||o}$ is in weak code inheritance mode if $imode_{\mathcal{I}}(\mathsf{R||o}) = \mathbf{u}$.

Note that in an interpretation atoms of the form $\mathsf{o[m \twoheadrightarrow v]}\,{}^{\mathsf{c}}_{\mathrm{code}}$ represent those facts that we can derive after binding a C-rule, which specifies the instance method m for the class c, with the object o via code inheritance. The truth valuation function can be extended to ground C-rules as follows.

Definition 6.3.4 Let $\mathcal{I}$ be an interpretation, $\mathsf{R} \equiv (\mathsf{code}\ \ \mathsf{c[m \twoheadrightarrow v]} \longleftarrow \mathsf{B})$ be a ground C-rule, and $\mathsf{F} \equiv (\mathsf{code}\ \ \mathsf{c[m \twoheadrightarrow v]})$ be a ground C-fact. The truth valuation function $\mathcal{I}$ on $\mathsf{R||o}$ and $\mathsf{F||o}$ (the bindings of R and F with respect to o, respectively)

is defined as follows:

$$\mathcal{I}(\mathsf{R}||\mathsf{o}) = \begin{cases} \mathbf{t}, & \text{if } \mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) \geq \mathbf{u} \text{ and} \\ & \mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}) \geq \mathit{min}\{\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}), \mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o})\}; \\ \mathbf{t}, & \text{if } \mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{f} \text{ and } \mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}) = \mathbf{f}; \\ \mathbf{f}, & \text{otherwise.} \end{cases}$$

$$\mathcal{I}(\mathsf{F}||\mathsf{o}) = \begin{cases} \mathbf{t}, & \text{if } \mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) \geq \mathbf{u} \text{ and } \mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}) \geq \mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o}); \\ \mathbf{t}, & \text{if } \mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{f} \text{ and } \mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}) = \mathbf{f}; \\ \mathbf{f}, & \text{otherwise.} \end{cases}$$

Note that when $\mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{f}$, *i.e.*, o neither strongly nor weakly inherits R, it is required that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}) = \mathbf{f}$ to satisfy $\mathsf{R}||\mathsf{o}$. The intuition behind this is that if an object cannot inherit a C-rule at all, then the effect of code inheritance should not be seen in the model. Therefore, strong or weak code inheritance mode should be a necessary condition for code inheritance to take place.

There is an interesting observation. In the case of strong code inheritance, the truth valuation function on C-rules will be defined essentially the same way as on V-rules. Clearly, $\mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{t}$ under strong code inheritance. It follows that: (i) $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}) \geq \mathit{min}\{\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}), \mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o})\}$ iff $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}) \geq \mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}})$; and (ii) $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}) \geq \mathit{imode}_{\mathcal{I}}(\mathsf{R}||\mathsf{o})$ iff $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}) = \mathbf{t}$.

The idea of C-rule satisfaction can be simply stated as follows.

Definition 6.3.5 (C-Rule Satisfaction) A three-valued interpretation $\mathcal{I}$ *satisfies* the C-rules of an F-logic program P, if $\mathcal{I}(\mathsf{R}||\mathsf{o}) = \mathbf{t}$ for all C-rule $\mathsf{R} \in \mathit{ground}(\mathrm{P})$ and all $\mathsf{o} \in \mathcal{HU}_{\mathrm{P}}$.

6.4 Object Models

We are now ready to state formally what it means to be an *object model* of an F-logic program.

Definition 6.4.1 (Object Model) An interpretation $\mathcal{I}$ of an F-logic program P is called an *object model* of P, if $\mathcal{I}$ satisfies both the V-rules and the C-rules in P, plus the following three postulates: the positive ISA transitivity constraint, the context consistency constraint, and the unique source inheritance constraint.

Example 6.4.1 Consider the two programs in Figures 6(b) and 6(c) which share the same class hierarchy as shown in Figure 6(a). Let

$$\mathrm{C} = \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_1 : \mathsf{c}_3, \mathsf{c}_2 :: \mathsf{c}_4, \mathsf{c}_3 :: \mathsf{c}_4\}$$

and $\mathcal{I}_1 = \langle\, \mathrm{T}_1; \mathrm{U}_1 \,\rangle$ be an interpretation for the program in Figure 6(b), where

$$\begin{aligned} \mathrm{T}_1 &= \mathrm{C} \cup \{\mathsf{c_4}[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c_4}}_{\text{local}}, \mathsf{c_1}[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c_4}}_{\text{value}}\} \\ \mathrm{U}_1 &= \emptyset \end{aligned}$$

One can verify that $\mathcal{I}_1$ is an object model for the program in Figure 6(b). From $\mathcal{I}_1$ we can see that $\mathsf{c_4}[\mathsf{m}]$ is the *unique* strong value inheritance candidate for $\mathsf{c_1}$ and $\mathsf{m} \twoheadrightarrow \mathsf{a}$ is locally defined at $\mathsf{c_4}$. Therefore, $\mathsf{c_1}$ can inherit $\mathsf{m} \twoheadrightarrow \mathsf{a}$ from $\mathsf{c_4}$.

Let C be the same set of ISA atoms as before and consider the interpretation $\mathcal{I}_2 = \langle\, \mathrm{T}_2; \mathrm{U}_2 \,\rangle$ for the program in Figure 6(c), where

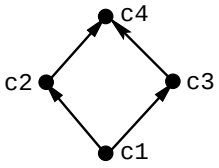
$$\begin{aligned} \mathrm{T}_2 &= \mathrm{C} \cup \{\mathsf{c_2}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_2}}_{\text{local}}, \mathsf{c_3}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_3}}_{\text{local}}, \mathsf{c_1}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_2}}_{\text{value}}, \mathsf{c_1}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_3}}_{\text{value}}\} \\ \mathrm{U}_2 &= \emptyset \end{aligned}$$

Clearly, $\mathcal{I}_2$ satisfies the program in Figure 6(c). But it is not an object model — the presence of each one of $\mathsf{c_1}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_2}}_{\text{value}}$ and $\mathsf{c_1}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_3}}_{\text{value}}$ in $\mathcal{I}_2$ violates the first condition of the unique source inheritance constraint.

Finally, consider $\mathcal{I}_3 = \langle\, \mathrm{T}_3; \mathrm{U}_3 \,\rangle$ for the program in Figure 6(c), where

$$\begin{aligned} \mathrm{T}_3 &= \mathrm{C} \cup \{\mathsf{c_2}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_2}}_{\text{local}}, \mathsf{c_3}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_3}}_{\text{local}}, \mathsf{c_1}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_2}}_{\text{value}}\} \\ \mathrm{U}_3 &= \emptyset \end{aligned}$$

However, $\mathcal{I}_3$ is not an object model either — the presence of $\mathsf{c_1}[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c_2}}_{\text{value}}$ in $\mathcal{I}_3$ violates both the second and the third condition of the unique source inheritance constraint, because both $\mathsf{c_2}[\mathsf{m}]$ and $\mathsf{c_3}[\mathsf{m}]$ are strong value inheritance candidates for $\mathsf{c_1}$ and $\mathsf{c_2} \neq \mathsf{c_3}$. □



(a)

$\mathsf{c_1} :: \mathsf{c_2}.$
$\mathsf{c_1} :: \mathsf{c_3}.$
$\mathsf{c_2} :: \mathsf{c_4}.$
$\mathsf{c_3} :: \mathsf{c_4}.$
$\mathsf{c_4}[\mathsf{m} \twoheadrightarrow \mathsf{a}].$

(b)

$\mathsf{c_1} : \mathsf{c_2}.$
$\mathsf{c_1} : \mathsf{c_3}.$
$\mathsf{c_2} :: \mathsf{c_4}.$
$\mathsf{c_3} :: \mathsf{c_4}.$
$\mathsf{c_2}[\mathsf{m} \twoheadrightarrow \mathsf{b}].$
$\mathsf{c_3}[\mathsf{m} \twoheadrightarrow \mathsf{b}].$

(c)

Figure 6: Unique Source Inheritance

It is worth pointing out the difference between *source-based* and *value-based* approaches to nonmonotonic multiple inheritance. Suppose $\mathsf{c_2}[\mathsf{m}]$ and $\mathsf{c_3}[\mathsf{m}]$ are both strong inheritance candidates for $\mathsf{c_1}$, where $\mathsf{c_2} \neq \mathsf{c_3}$. In the source-based approach $\mathsf{c_1}$ has a multiple inheritance conflict on the method m regardless of the return values

of the method m in c_1 and c_2. On the contrary, in the value-based approach, no conflict would occur if m returns the *same* set of values in both classes c_2 and c_3. For instance, the above interpretation $\mathcal{I}_2$ for the program in Figure 6(c) would be an object model under the value-based approach, since m returns the same set of values, $\{\mathsf{b}\}$, in c_2 and c_3. However, value-based nonmonotonic multiple inheritance requires higher-order reasoning and is expensive to compute. In this dissertation we consider only source-based inheritance.

6.5 Optimistic Inheritance Postulates

The constraints introduced so far capture the intuition behind the "definite" part of an object model, *i.e.*, the true and the false components. We view them as *core postulates* that any reasonable object model must obey. However, we still need to assign a meaning to the undefined part of an object model. Since "undefined" means either possibly true or possibly false, intuitively we want the conclusions drawn from undefined facts to remain undefined. In other words, the semantics should be "closed" with regard to undefined facts. As a consequence, although it might seem tempting to "jump" to negative conclusions from undefined facts in some cases (*e.g.*, if there are multiple weak inheritance candidates), our semantics is biased towards undefined conclusions, which is why we call it "optimistic".

Definition 6.5.1 (Optimistic ISA Transitivity) An interpretation $\mathcal{I}$ of an F-logic program P satisfies the *optimistic ISA transitivity constraint* if the undefined part of the class hierarchy is transitively closed, formally, if the following two conditions hold:

(1) for all s, c: if there is x such that $\mathcal{I}(\mathsf{s}::\mathsf{x} \wedge \mathsf{x}::\mathsf{c}) = \mathbf{u}$ and $\mathcal{I}(\mathsf{s}::\mathsf{c}) \neq \mathbf{t}$, then $\mathcal{I}(\mathsf{s}::\mathsf{c}) = \mathbf{u}$;

(2) for all o, c: if there is x such that $\mathcal{I}(\mathsf{o}:\mathsf{x} \wedge \mathsf{x}::\mathsf{c}) = \mathbf{u}$ and $\mathcal{I}(\mathsf{o}:\mathsf{c}) \neq \mathbf{t}$, then $\mathcal{I}(\mathsf{o}:\mathsf{c}) = \mathbf{u}$.

Definition 6.5.2 (Optimistic Inheritance) An interpretation $\mathcal{I}$ of an F-logic program P satisfies the *optimistic inheritance constraint*, if for all o, m, v, c: $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{u}$ *iff*

(i) $\mathsf{o}[\mathsf{m}]$ is not a strong local context; and

(ii) $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$; and

(iii) $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}}) \geq \mathbf{u}$; and

(iv) there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$; and

(v) $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) \neq \mathbf{t}$.

The optimistic inheritance constraint captures the intuition behind multiple inheritance based on undefined knowledge. The first condition above states when optimistic value inheritance takes place while the second condition states when optimistic code inheritance takes place.

An object o *optimistically* inherits $\mathsf{m} \twoheadrightarrow \mathsf{v}$ from a class c by value inheritance if and only if: (i) there is no strong evidence that the method m has a locally defined value at o; (ii) c[m] is either a strong or a weak value inheritance candidate for o; (iii) $\mathsf{m} \twoheadrightarrow \mathsf{v}$ is locally defined at c; (iv) there are no other strong inheritance candidates that can invalidate value inheritance from c (by the unique source inheritance constraint); and (v) o cannot positively inherit $\mathsf{m} \twoheadrightarrow \mathsf{v}$ from c by value inheritance.

Chapter 7

Computation

In this chapter we will define a series of operators. These operators form the basis of a bottom-up computation procedure which will be used to compute object models for F-logic programs.

7.1 Extended Atom Sets

First we need to extend the definition of an interpretation in Section 5.2 to include book-keeping information used by the computation. The book-keeping information will be projected out when the final object model is produced.

The *extended Herbrand base* of an F-logic program P, denoted $\widehat{\mathcal{HB}}_{\mathrm{P}}$, consists of atoms from $\mathcal{HB}_{\mathrm{P}}$ and *auxiliary* atoms of the forms $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o}$ and $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o}$, where c, m, and o are terms from $\mathcal{HU}_{\mathrm{P}}$. During the computation, we will use auxiliary atoms of the forms $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o}$ and $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o}$ to approximate value and code inheritance candidates, respectively.

An *extended atom set* is a subset of $\widehat{\mathcal{HB}}_{\mathrm{P}}$. In the sequel, we will use symbols with a hat (*e.g.*, $\widehat{\mathrm{I}}$) to denote extended atom sets. The *projection* of an extended atom set $\widehat{\mathrm{I}}$, denoted $\pi(\widehat{\mathrm{I}})$, is $\widehat{\mathrm{I}}$ with the auxiliary atoms removed.

Lemma 7.1.1 Let $\widehat{\mathrm{I}}$ and $\widehat{\mathrm{J}}$ be extended atom sets:

(1) If $\widehat{\mathrm{I}} \subseteq \widehat{\mathrm{J}}$, then $\pi(\widehat{\mathrm{I}}) \subseteq \pi(\widehat{\mathrm{J}})$.

(2) $\pi(\widehat{\mathrm{I}} \cup \widehat{\mathrm{J}}) = \pi(\widehat{\mathrm{I}}) \cup \pi(\widehat{\mathrm{J}})$

(3) $\pi(\widehat{\mathrm{I}} - \widehat{\mathrm{J}}) = \pi(\widehat{\mathrm{I}}) - \pi(\widehat{\mathrm{J}})$

Frequently we will need to compare a normal atom set with the projection of an extended atom set to test for set inclusion. Without complicating the presentation

we will usually omit the project function and just write the extended atom set, when its intended usage is clear from the context.

It is straightforward to extend the definitions of the truth valuation functions in Section 5.3 to extended atom sets, since the auxiliary atoms do not occur in F-logic programs. Formally, given an extended atom set $\widehat{\mathrm{I}}$, let $\mathcal{I} = \langle\, \pi(\widehat{\mathrm{I}}); \emptyset \,\rangle$. We define:

$$\begin{aligned}
val^{\mathbf{h}}_{\widehat{\mathrm{I}}}(\mathsf{H}) &\stackrel{\text{def}}{=} \mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}), && \text{for a ground rule head}\\
val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\mathsf{B}) &\stackrel{\text{def}}{=} \mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}), && \text{for a ground rule body}\\
val_{\widehat{\mathrm{I}}}(\mathsf{R}) &\stackrel{\text{def}}{=} \mathcal{I}(\mathsf{R}), && \text{for a ground V-rule}\\
val_{\widehat{\mathrm{I}}}(\mathsf{R}||\mathsf{o}) &\stackrel{\text{def}}{=} \mathcal{I}(\mathsf{R}||\mathsf{o}), && \text{for a binding of a ground C-rule}
\end{aligned}$$

7.2 Operators

The computation to be defined in this section extends the alternating fixpoint computation in [17]. The new element here is the book-keeping mechanism for recording inheritance information.

Definition 7.2.1 Given a ground literal L of an F-logic program P and an atom $\mathsf{A} \in \mathcal{HB}_{\mathrm{P}}$, we say that L *matches* A, if one of the following conditions is true:

(1) $\mathsf{L} = \mathsf{o}:\mathsf{c}$ and $\mathsf{A} = \mathsf{o}:\mathsf{c}$

(2) $\mathsf{L} = \mathsf{s}::\mathsf{c}$ and $\mathsf{A} = \mathsf{s}::\mathsf{c}$

(3) $\mathsf{L} = \mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$ and $\mathsf{A} = \mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{s}}_{\text{local}}$

Definition 7.2.2 (V-Rule Consequence Operator $\mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}$) The *V-rule consequence operator*, $\mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}$, is defined for an F-logic program P and an extended atom set $\widehat{\mathrm{I}}$. It takes as input an extended atom set, $\widehat{\mathrm{J}}$, and generates a new extended atom set as follows:

$$\mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}}) = \left\{ \mathsf{A} \;\middle|\; \begin{array}{l} \text{There is a V-rule, } \mathsf{H} \longleftarrow \mathsf{L}_1, \ldots, \mathsf{L}_n, \text{ in } \mathit{ground}(\mathrm{P}), \text{ such that} \\ \mathsf{H} \text{ matches } \mathsf{A} \text{ and for every literal } \mathsf{L}_i \;\; (1 \le i \le n): \\ \quad \text{(i) if } \mathsf{L}_i \text{ is positive, then } val^{\mathbf{b}}_{\widehat{\mathrm{J}}}(\mathsf{L}_i) = \mathbf{t}; \quad \text{and} \\ \quad \text{(ii) if } \mathsf{L}_i \text{ is negative, then } val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\mathsf{L}_i) = \mathbf{t}. \end{array} \right\}$$

The V-rule consequence operator is adopted from the usual alternating fixpoint computation. It derives new facts, including class memberships, subclass relationships, and local method definitions for classes and objects, from the V-rules in an F-logic program.

Definition 7.2.3 (Inheritance Blocking Operator $\mathbf{IB}_{\mathrm{P}}$) The *inheritance blocking operator*, $\mathbf{IB}_{\mathrm{P}}$, is defined for an F-logic program P. It takes as input an extended atom set, $\widehat{\mathrm{I}}$, and generates the following set of atoms:

$$\begin{aligned}\mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}}) \;=\; & \left\{ lc(\mathsf{o},\mathsf{m}) \mid \exists\, \mathsf{v}, \text{ such that } \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{local}} \in \widehat{\mathrm{I}} \right\} \cup \\ & \left\{ mc(\mathsf{c},\mathsf{m},\mathsf{o}) \;\middle|\; \exists\, \mathsf{x} \neq \mathsf{c} \text{ such that } \mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{I}} \text{ or } \mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{I}} \right\} \cup \\ & \left\{ ov(\mathsf{c},\mathsf{m},\mathsf{o}) \;\middle|\; \begin{array}{l} \exists \mathsf{x} \text{ such that: (i) } \mathsf{x} \neq \mathsf{c},\ \mathsf{x} \neq \mathsf{o},\ \mathsf{x} :: \mathsf{c} \in \widehat{\mathrm{I}},\ \mathsf{o} : \mathsf{x} \in \widehat{\mathrm{I}}; \\ \text{and (ii) } \exists \mathsf{v} \text{ such that } \mathsf{x}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\mathrm{local}} \in \widehat{\mathrm{I}} \text{ or there is} \\ \text{a C-rule in } \mathit{ground}(\mathrm{P}) \text{ which specifies the instance} \\ \text{method } \mathsf{m} \text{ for the class } \mathsf{x}. \end{array} \right\} \end{aligned}$$

The inheritance blocking operator is an auxiliary operator used in defining the C-rule consequence operator and the inheritance consequence operator below. It returns the book-keeping information that is needed in deciding what can be inherited and which ones are the inheritance candidates.

Intuitively, $lc(\mathsf{o},\mathsf{m})$ means that the method m is locally defined at o; $mc(\mathsf{c},\mathsf{m},\mathsf{o})$ means that inheritance of the method m from c to o is not possible due to a multiple inheritance conflict (as manifested by the existence of either a value or a code inheritance candidate that is different from c); $ov(\mathsf{c},\mathsf{m},\mathsf{o})$ means that inheritance of the method m from c to o would be overridden by another class that stands between o and c in the class hierarchy. From the definition we can see that a class must have a locally defined value for a method or have an instance method definition to be able to override inheritance from its superclasses.

Lemma 7.2.1 Given an interpretation $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ of an F-logic program P:

(1) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: there is x such that x strongly overrides $\mathsf{c}[\mathsf{m}]$ for o *iff* $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$.

(2) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: there is x such that x strongly or weakly overrides $\mathsf{c}[\mathsf{m}]$ for o *iff* $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\mathrm{T} \cup \mathrm{U})$.

Proof.

By Definition 6.1.4 and Definition 7.2.3.

□

Lemma 7.2.2 Given an interpretation $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ of an F-logic program P:

(1) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ *iff* (i) $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o} : \mathsf{c} \in \mathrm{T}$; (ii) $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \mathrm{T}$ for some value v; and (iii) $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T} \cup \mathrm{U})$.

(2) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ *iff* (i) $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o} : \mathsf{c} \in \mathrm{T}$; (ii) there is a C-rule in $\mathit{ground}(\mathrm{P})$ which specifies the instance method m for the class c; and (iii) $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T} \cup \mathrm{U})$.

(3) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ *iff* (i) $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o}:\mathsf{c} \in \mathrm{T} \cup \mathrm{U}$; (ii) $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \mathrm{T} \cup \mathrm{U}$ for some value v; and (iii) $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$.

(4) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ *iff* (i) $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o}:\mathsf{c} \in \mathrm{T} \cup \mathrm{U}$; (ii) there is a C-rule in $ground(\mathrm{P})$ which specifies the instance method m for the class c; and (iii) $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$.

(5) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$ *iff* (i) $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o}:\mathsf{c} \in \mathrm{T} \cup \mathrm{U}$; (ii) $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \mathrm{T} \cup \mathrm{U}$ for some value v or there is a C-rule in $ground(\mathrm{P})$ which specifies the instance method m for the class c; and (iii) $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$.

Proof.

By Definitions 6.1.5 and 6.1.6 and Lemma 7.2.1.

□

Definition 7.2.4 (C-Rule Consequence Operator $\mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}$) The *C-rule consequence operator*, $\mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}$, is defined for an F-logic program P and an extended atom set $\widehat{\mathrm{I}}$. It takes as input an extended atom set, $\widehat{\mathrm{J}}$, and generates a new extended atom set as follows:

$$\mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}}) = \left\{ \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \;\middle|\; \begin{array}{l} \mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}},\ lc(\mathsf{o}, \mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}}), \\ mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}}), \text{ and there is a} \\ \text{C-rule, } \mathsf{code} \quad \mathsf{H} \longleftarrow \mathsf{B}, \text{ in } ground(\mathrm{P}) \\ \text{such that } \mathsf{H} \equiv \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \text{ and for every} \\ \text{literal } \mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}: \\ \quad \text{(i) if } \mathsf{L} \text{ is positive, then} \\ \quad\quad val^{\mathbf{b}}_{\widehat{\mathrm{J}}}(\mathsf{L}) = \mathbf{t}; \quad \text{and} \\ \quad \text{(ii) if } \mathsf{L} \text{ is negative, then} \\ \quad\quad val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\mathsf{L}) = \mathbf{t}. \end{array} \right\}$$

The C-rule consequence operator is used to derive new facts as a result of code inheritance. It is similar to the V-rule consequence operator except that the V-rule consequence operator is applied to all V-rules whereas the C-rule consequence operator is applied to only those selected C-rules that could be inherited according to our inheritance semantics.

Given an object o and a C-rule, $\mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{B}$, which specifies the instance method m for the class c, in $ground(\mathrm{P})$, we first need to decide whether o can inherit this instance method definition from c. If so, then we will bind this instance method definition with respect to o and evaluate it (note that $\mathsf{L}_{\mathsf{c}\backslash\mathsf{o}}$ is obtained from L by substituting o for every occurrence of c in L). If the rule body is satisfied in the context of o, we will derive $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}$ to represent the fact that $\mathsf{m} \twoheadrightarrow \mathsf{v}$ is established for o by inheritance of an instance method definition from c.

We can decide whether o can inherit code from c by looking up the two sets $\widehat{\mathrm{J}}$ and $\mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$. The object o can inherit the instance method definition of m from the class c only if the following conditions are true: (i) $\mathsf{c}[\mathsf{m}]$ is a code inheritance candidate for o ($\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$); (ii) the method m is not locally defined at o ($lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$); and (iii) there is no multiple inheritance conflict ($mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$).

Definition 7.2.5 (Inheritance Consequence Operator $\mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}$) The *inheritance consequence operator*, $\mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}$, where P is an F-logic program and $\widehat{\mathrm{I}}$ is an extended atom set, takes as input an extended atom set, $\widehat{\mathrm{J}}$, and generates a new extended atom set as follows:

$$\mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}}) \overset{\text{def}}{=} \mathbf{IC}^{t}(\widehat{\mathrm{J}}) \cup \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}}) \cup \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}})$$

$$\mathbf{IC}^{t}(\widehat{\mathrm{J}}) = \left\{ \mathsf{o}:\mathsf{c} \;\middle|\; \exists \mathsf{x}, \text{ such that } \mathsf{o}:\mathsf{x} \in \widehat{\mathrm{J}}, \mathsf{x}::\mathsf{c} \in \widehat{\mathrm{J}} \right\} \cup \left\{ \mathsf{s}::\mathsf{c} \;\middle|\; \exists \mathsf{x}, \text{ such that } \mathsf{s}::\mathsf{x} \in \widehat{\mathrm{J}}, \mathsf{x}::\mathsf{c} \in \widehat{\mathrm{J}} \right\}$$

$$\mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}}) = \left\{ \mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \;\middle|\; \begin{array}{l} \mathsf{o}:\mathsf{c} \in \widehat{\mathrm{J}},\ \mathsf{c} \neq \mathsf{o},\ \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}} \in \widehat{\mathrm{J}}, \\ \text{and } ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}}) \end{array} \right\} \cup \left\{ \mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \;\middle|\; \begin{array}{l} \mathsf{o}:\mathsf{c} \in \widehat{\mathrm{J}},\ \mathsf{c} \neq \mathsf{o}, \text{ there is a C-rule in } \mathit{ground}(\mathrm{P}) \\ \text{which specifies the instance method } \mathsf{m} \text{ for the} \\ \text{class } \mathsf{c}, \text{ and } ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}}) \end{array} \right\}$$

$$\mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}}) = \left\{ \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \;\middle|\; \begin{array}{l} \mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}},\ \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}} \in \widehat{\mathrm{J}}, \\ lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}}), \text{ and } mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}}) \end{array} \right\}$$

The inheritance consequence operator, $\mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}$, is the union of three operators: $\mathbf{IC}^{t}$, $\mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{I}}}$, and $\mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{I}}}$. The operator $\mathbf{IC}^{t}$ is used to perform transitive closure of the class hierarchy, including class memberships and subclass relationships. Value and code inheritance candidates are computed by the operator $\mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{I}}}$, which relies on the overriding information provided by $\mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$. Finally, the operator $\mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{I}}}$ derives new facts by value inheritance. This operator also relies on information provided by $\mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$ to make inheritance decisions.

Definition 7.2.6 (Program Completion Operator $\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}$) The *program completion operator*, $\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}$, where P is an F-logic program and $\widehat{\mathrm{I}}$ an extended atom set, takes as input an extended atom set, $\widehat{\mathrm{J}}$, and generates a new extended atom set as follows:

$$\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}}) \overset{\text{def}}{=} \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}}) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}}) \cup \mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}})$$

The program completion operator is simply the union of the V-rule consequence operator, the C-rule consequence operator, and the inheritance consequence operator.

It derives new "local" method definitions (via V-rules in the program), new "contextual" method definitions as a result of binding instance method specifications with class members (via C-rules in the program), new inherited facts (by value and code inheritance), plus inheritance candidacy information that is used to decide which facts to inherit in the future.

We have the following lemma regarding the *monotonicity* property of the operators that we have defined so far.

Lemma 7.2.3

(1) $\mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}$ is monotonic when P and $\widehat{\mathrm{I}}$ are fixed.

(2) $\mathbf{IB}_{\mathrm{P}}$ is monotonic when P is fixed.

(3) $\mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}$ is monotonic when P and $\widehat{\mathrm{I}}$ are fixed.

(4) $\mathbf{IC}^{t}$ is monotonic. $\mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{I}}}$ and $\mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{I}}}$ are monotonic when P and $\widehat{\mathrm{I}}$ are fixed.

(5) $\mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}$ is monotonic when P and $\widehat{\mathrm{I}}$ are fixed.

(6) $\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}$ is monotonic when P and $\widehat{\mathrm{I}}$ are fixed.

Given an F-logic program P, the set of all subsets of the extended Herbrand base $\widehat{\mathcal{HB}}_{\mathrm{P}}$ constitutes a complete lattice where the partial ordering is defined by set inclusion. Therefore, any monotonic operator, Φ, defined on this lattice has a unique least fixpoint $\mathrm{lfp}(\Phi)$ [37].

Definition 7.2.7 (Alternating Fixpoint Operator Ψ_{P}) The *alternating fixpoint operator*, Ψ_{P}, for an F-logic program P takes as input an extended atom set, $\widehat{\mathrm{I}}$, and generates a new extended atom set as follows:

$$\Psi_{\mathrm{P}}(\widehat{\mathrm{I}}) \stackrel{\mathrm{def}}{=} \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}})$$

Definition 7.2.8 (F-logic Fixpoint Operator $\mathbf{F}_{\mathrm{P}}$) The *F-logic fixpoint operator*, $\mathbf{F}_{\mathrm{P}}$, where P is an F-logic program, takes as input an extended atom set, $\widehat{\mathrm{I}}$, and generates a new extended atom set as follows:

$$\mathbf{F}_{\mathrm{P}}(\widehat{\mathrm{I}}) \stackrel{\mathrm{def}}{=} \Psi_{\mathrm{P}}(\Psi_{\mathrm{P}}(\widehat{\mathrm{I}}))$$

Lemma 7.2.4 Let $\widehat{\mathrm{I}}$ be an extended atom set of an F-logic program P, $\widehat{\mathrm{J}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{I}})$. Then:

(1) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \stackrel{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$ then $\mathsf{c} \neq \mathsf{o}$.

(2) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \stackrel{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$ then $\mathsf{c} \neq \mathsf{o}$.

(3) for all $\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}$: $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \widehat{\mathrm{J}}$ *iff* $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}})$.

(4) for all $\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}$: $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}} \in \widehat{\mathrm{J}}$ *iff* $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{J}})$.

(5) for all $\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}$: if $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \widehat{\mathrm{J}}$ then $\mathsf{c} \neq \mathsf{o}$.

(6) for all $\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}$: if $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}} \in \widehat{\mathrm{J}}$ then $\mathsf{c} \neq \mathsf{o}$.

Proof.

By Definitions 7.2.7, 7.2.6, 7.2.5, and 7.2.4.
□

Lemma 7.2.5 Ψ_{P} is antimonotonic when P is fixed.

Proof.

We want to show that for any extended atom sets $\widehat{\mathrm{I}}$ and $\widehat{\mathrm{J}}$: if $\widehat{\mathrm{I}} \supseteq \widehat{\mathrm{J}}$, then $\Psi_{\mathrm{P}}(\widehat{\mathrm{I}}) \subseteq \Psi_{\mathrm{P}}(\widehat{\mathrm{J}})$. Let α range over all countable ordinals, define:

$$\begin{array}{lcllcll}
\widehat{\mathrm{X}}_0 & = & \emptyset & \widehat{\mathrm{Y}}_0 & = & \emptyset & \text{for limit ordinal } 0 \\
\widehat{\mathrm{X}}_\alpha & = & \mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) & \widehat{\mathrm{Y}}_\alpha & = & \mathbf{T}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1}) & \text{for successor ordinal } \alpha \\
\widehat{\mathrm{X}}_\alpha & = & \bigcup_{\beta<\alpha} \widehat{\mathrm{X}}_\beta & \widehat{\mathrm{Y}}_\alpha & = & \bigcup_{\beta<\alpha} \widehat{\mathrm{Y}}_\beta & \text{for limit ordinal } \alpha \neq 0
\end{array}$$

By Definition 7.2.7 $\Psi_{\mathrm{P}}(\widehat{\mathrm{I}}) = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}})$ and $\Psi_{\mathrm{P}}(\widehat{\mathrm{J}}) = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{J}}})$. So, by Proposition 4.1.2, to show that $\Psi_{\mathrm{P}}(\widehat{\mathrm{I}}) \subseteq \Psi_{\mathrm{P}}(\widehat{\mathrm{J}})$ it suffices to show that $\widehat{\mathrm{X}}_\alpha \subseteq \widehat{\mathrm{Y}}_\alpha$ for any ordinal α.

The case is trivial for a limit ordinal α. Now suppose α is a successor ordinal. Then $\widehat{\mathrm{X}}_\alpha = \mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) = \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) \cup \mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1})$ and $\widehat{\mathrm{Y}}_\alpha = \mathbf{T}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1}) = \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1}) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1}) \cup \mathbf{IC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1})$, by Definition 7.2.6. Therefore, to show that $\widehat{\mathrm{X}}_\alpha \subseteq \widehat{\mathrm{Y}}_\alpha$, it suffices to show that $\mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) \subseteq \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1})$, $\mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) \subseteq \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1})$, and $\mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) \subseteq \mathbf{IC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1})$.

Let $\widehat{\mathcal{I}} = \langle\, \widehat{\mathrm{I}}; \emptyset \,\rangle$, $\widehat{\mathcal{X}}_{\alpha-1} = \langle\, \widehat{\mathrm{X}}_{\alpha-1}; \emptyset \,\rangle$, $\widehat{\mathcal{J}} = \langle\, \widehat{\mathrm{J}}; \emptyset \,\rangle$, $\widehat{\mathcal{Y}}_{\alpha-1} = \langle\, \widehat{\mathrm{Y}}_{\alpha-1}; \emptyset \,\rangle$.

First we will show that $\mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) \subseteq \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1})$. Let A be any atom such that $\mathsf{A} \in \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1})$. Then by Definition 7.2.2, there must exit a V-rule, $\mathsf{H} \longleftarrow \mathsf{L}_1, \ldots, \mathsf{L}_{\mathsf{n}}$, in *ground*(P), such that H matches A and for all $\mathsf{L}_{\mathsf{i}}, 1 \leq \mathsf{i} \leq \mathsf{n}$: (i) if L_{i} is a positive literal then $\mathcal{V}^{\mathbf{b}}_{\widehat{\mathcal{X}}_{\alpha-1}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$; and (ii) if L_{i} is a negative literal then $\mathcal{V}^{\mathbf{b}}_{\widehat{\mathcal{I}}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$. Note that $\widehat{\mathrm{X}}_{\alpha-1} \subseteq \widehat{\mathrm{Y}}_{\alpha-1}$ by the induction hypothesis and $\widehat{\mathrm{I}} \supseteq \widehat{\mathrm{J}}$. So, by Lemma 5.3.2, for all $\mathsf{L}_{\mathsf{i}}, 1 \leq \mathsf{i} \leq \mathsf{n}$: (i) if L_{i} is a positive literal then $\mathcal{V}^{\mathbf{b}}_{\widehat{\mathcal{Y}}_{\alpha-1}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$; and (ii) if L_{i} is a negative literal then $\mathcal{V}^{\mathbf{b}}_{\widehat{\mathcal{J}}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$. It follows that $\mathsf{A} \in \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1})$.

Next we will show that $\mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) \subseteq \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1})$. Let $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}$ be any atom in $\mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1})$. Then by Definition 7.2.4, $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{X}}_{\alpha-1}$, $lc(\mathsf{o}, \mathsf{m}) \notin$

$\mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$, and there must exit a C-rule, $\mathsf{code}\quad \mathsf{c}[\mathsf{m} \rightarrow\!\!\!\!\rightarrow \mathsf{v}] \longleftarrow \mathsf{B}$, in *ground*(P), such that for all literal $\mathsf{L} \in \mathsf{B}$: (i) if L is positive then $\mathcal{V}^{\mathbf{b}}_{\widehat{\mathcal{X}}_{\alpha-1}}(\mathsf{L}_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{t}$; and (ii) if L is negative then $\mathcal{V}^{\mathbf{b}}_{\widehat{\mathcal{I}}}(\mathsf{L}_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{t}$. Note that $\widehat{\mathrm{X}}_{\alpha-1} \subseteq \widehat{\mathrm{Y}}_{\alpha-1}$ by the induction hypothesis and $\mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}}) \supseteq \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$ by the monotonicity of $\mathbf{IB}_{\mathrm{P}}$. So $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{Y}}_{\alpha-1}$, $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$, and by Lemma 5.3.2, for all literal $\mathsf{L} \in \mathsf{B}$: (i) if L is positive then $\mathcal{V}^{\mathbf{b}}_{\widehat{\mathcal{Y}}_{\alpha-1}}(\mathsf{L}) = \mathbf{t}$; and (ii) if L is negative then $\mathcal{V}^{\mathbf{b}}_{\widehat{\mathcal{J}}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$. It follows that $\mathsf{o}[\mathsf{m} \rightarrow\!\!\!\!\rightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1})$.

Finally we will show that $\mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1}) \subseteq \mathbf{IC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{Y}}_{\alpha-1})$. Note that $\widehat{\mathrm{X}}_{\alpha-1} \subseteq \widehat{\mathrm{Y}}_{\alpha-1}$ by the induction hypothesis and $\mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}}) \supseteq \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$ by the monotonicity of $\mathbf{IB}_{\mathrm{P}}$. Clearly, for any atom A, if $\mathsf{A} \in \mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{X}}_{\alpha-1})$ then $\mathsf{A} \in \mathbf{IC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{Y}}_{\alpha-1})$, by Definition 7.2.5.
□

Lemma 7.2.6 $\mathbf{F}_{\mathrm{P}}$ is monotonic when P is fixed.

Proof.

By Definition 7.2.8 and Lemma 7.2.5.
□

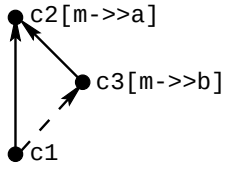
Chapter 8

Stable Object Models

In this chapter we will introduce a special type of object model, called *stable object model*, which does not exhibit some of the anomalies in inference. We will formally prove that a stable object model indeed satisfies all the requirements of an object model. At the end of this chapter, we will illustrate the relationship between stable object models and fxipoints through some interesting examples.

8.1 Stable Interpretations

Although V-rule satisfaction, the inheritance postulates, and C-rule satisfaction have ruled out a large number of unintended interpretations of an F-logic program which do not satisfy the necessary requirements of an object model, they still do not restrict object models tightly enough. In fact, for an F-logic program there may exist *unfounded* object models that do not match the common intuition behind inference. This problem is illustrated by the following example.



$c_1 : c_2.$
$c_3 :: c_2.$
$c_2[m \twoheadrightarrow a].$
$c_3[m \twoheadrightarrow b].$
$c_1 : c_3 \longleftarrow c_1[m \twoheadrightarrow b].$

Figure 7: Unfounded Inference

Example 8.1.1 Consider the program in Figure 7 and the *two-valued* object model $\mathcal{I} = \langle\, P; Q \,\rangle$, where

$$P \;=\; \{c_1 : c_2, c_3 :: c_2, c_1 : c_3, c_2[m \twoheadrightarrow a]^{c_2}_{\text{local}}, c_3[m \twoheadrightarrow b]^{c_3}_{\text{local}}, c_1[m \twoheadrightarrow b]^{c_3}_{\text{value}}\},$$

$$\mathrm{Q} \;=\; \emptyset.$$

Clearly, $\mathcal{I}$ satisfies the V-rules of the program in Figure 7 and all the inheritance postulates introduced in Chapter 6, including the optimistic ISA transitivity constraint and the optimistic inheritance constraint. However, we should note that in $\mathcal{I}$ the truths of $\mathsf{c_1 : c_3}$ and $\mathsf{c_1[m \twoheadrightarrow b]}^{\mathsf{c_3}}_{\mathrm{value}}$ are not *well-founded* in that they mutually rely on the truth of each other as the necessary inference premise. Indeed, the truth of $\mathsf{c_1 : c_3}$ depends on the literal $\mathsf{c_1[m \twoheadrightarrow b]}$ being satisfied in the body of the last rule. Since $\mathsf{c_1[m \twoheadrightarrow b]}$ does not appear in the head of any rule, there is no way for $\mathsf{m \twoheadrightarrow b}$ to be locally defined for $\mathsf{c_1}$. So the satisfaction of the body literal $\mathsf{c_1[m \twoheadrightarrow b]}$ depends on $\mathsf{c_1}$ inheriting $\mathsf{m \twoheadrightarrow b}$ from $\mathsf{c_3}$, the only class that has locally defined $\mathsf{m \twoheadrightarrow b}$. However, $\mathsf{c_1}$ can inherit $\mathsf{m \twoheadrightarrow b}$ from $\mathsf{c_3}$ only if the truth of $\mathsf{c_1 : c_3}$ can be established. We can see that the inference of $\mathsf{c_1 : c_3}$ and the inference of $\mathsf{c_1[m \twoheadrightarrow b]}^{\mathsf{c_3}}_{\mathrm{value}}$ are at a *deadlock*. Therefore, we should not automatically conclude that both $\mathsf{c_1 : c_3}$ and $\mathsf{c_1[m \twoheadrightarrow b]}^{\mathsf{c_3}}_{\mathrm{value}}$ are true as implied by the program and our semantics for inheritance. □

Now we will introduce a special class of object models, namely the stable object models, which do not exhibit the aforementioned anomaly.

Definition 8.1.1 Given an interpretation $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ of an F-logic program P, let $\widehat{\mathrm{T}}_{\mathcal{I}}$ be the extended atom set constructed by the union of T and the set of auxiliary atoms corresponding to the strong inheritance candidates in $\mathcal{I}$, and $\widehat{\mathrm{U}}_{\mathcal{I}}$ be the extended atom set constructed by the union of T, U, and the set of auxiliary atoms corresponding to the strong and weak inheritance candidates in $\mathcal{I}$, *i.e.*,

$$\begin{aligned}
\widehat{\mathrm{T}}_{\mathcal{I}} \;\stackrel{\mathrm{def}}{=}\; & \mathrm{T} \cup \\
& \{\mathsf{c[m]} \overset{v}{\rightsquigarrow} \mathsf{o} \mid \mathsf{c[m]} \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}\} \cup \\
& \{\mathsf{c[m]} \overset{c}{\rightsquigarrow} \mathsf{o} \mid \mathsf{c[m]} \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}\} \\
\widehat{\mathrm{U}}_{\mathcal{I}} \;\stackrel{\mathrm{def}}{=}\; & \mathrm{T} \cup \mathrm{U} \cup \\
& \{\mathsf{c[m]} \overset{v}{\rightsquigarrow} \mathsf{o} \mid \mathsf{c[m]} \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o} \text{ or } \mathsf{c[m]} \overset{wv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}\} \cup \\
& \{\mathsf{c[m]} \overset{c}{\rightsquigarrow} \mathsf{o} \mid \mathsf{c[m]} \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o} \text{ or } \mathsf{c[m]} \overset{wc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}\}
\end{aligned}$$

Definition 8.1.2 (Stable Interpretation) Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be an interpretation of an F-logic program P. $\mathcal{I}$ is called a *stable interpretation* of P, if $\widehat{\mathrm{T}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ and $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$.

Our definition of stable interpretation is closely related to that of stable model introduced in [19, 45]. The idea is that given an interpretation $\mathcal{I}$ of an F-logic program P, we first resolve all the negative premises both in P and in our semantics for inheritance using the information in $\mathcal{I}$. The result is a residual positive program without negation. Then $\mathcal{I}$ is called stable if and only if $\mathcal{I}$ can reproduce itself by

resolving the positive premises both in the residual program and in our inheritance semantics via least fixpoint computation. This is how stable interpretations can prevent the kind of unfounded inference illustrated in Example 8.1.1.

We should note that in Definition 8.1.2 it is only required that a stable interpretation $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ satisfy a certain computational property with respect to Ψ_{P}, *i.e.*, $\widehat{\mathrm{T}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ and $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$. In fact, it turns out that a stable interpretation of an F-logic program P satisfies all the V-rules and C-rules in P as well as all the core and optimistic inheritance postulates. We will present the formal proofs in the next section.

8.2 Properties

Lemma 8.2.1 Let P be an F-logic program and $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be a stable interpretation of P:

$$\begin{aligned}
\widehat{\mathrm{T}}_{\mathcal{I}} &= \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}}) \cup \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}}) \cup \mathbf{IC}^{t}(\widehat{\mathrm{T}}_{\mathcal{I}}) \cup \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}}) \cup \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}}) \\
\widehat{\mathrm{U}}_{\mathcal{I}} &= \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}}) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}}) \cup \mathbf{IC}^{t}(\widehat{\mathrm{U}}_{\mathcal{I}}) \cup \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}}) \cup \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})
\end{aligned}$$

Proof.

By Definitions 8.1.2, 7.2.7, 7.2.6, and 7.2.5.
□

Proposition 8.2.2 Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be a stable interpretation of an F-logic program P. Then $\mathcal{I}$ satisfies the V-rules of P.

Proof. By contradiction.

Suppose on the contrary $\mathcal{I}$ does not satisfy the V-rules of P. Then by Definitions 5.4.2 and 5.4.1, there is a ground V-rule, $\mathsf{H} \longleftarrow \mathsf{L}_1, \ldots, \mathsf{L}_n$, in *ground*(P), such that $\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) < \mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_1 \wedge \ \ldots \ \wedge \mathsf{L}_n)$. It follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_1 \wedge \ \ldots \ \wedge \mathsf{L}_n) = \mathbf{t}$ and $\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) \neq \mathbf{t}$, or $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_1 \wedge \ \ldots \ \wedge \mathsf{L}_n) = \mathbf{u}$ and $\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) = \mathbf{f}$.

(1) $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_1 \wedge \ \ldots \ \wedge \mathsf{L}_n) = \mathbf{t}$ and $\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) \neq \mathbf{t}$

It follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_i) = \mathbf{t}$ for all $\mathsf{L}_i, 1 \leq i \leq n$, by Definition 5.3.1. So by Lemma 5.3.1: (i) if L_i is a positive literal then $val^{\mathbf{b}}_{\widehat{\mathrm{T}}_{\mathcal{I}}}(\mathsf{L}_i) = \mathbf{t}$; and (ii) if L_i is a negative literal then $val^{\mathbf{b}}_{\widehat{\mathrm{U}}_{\mathcal{I}}}(\mathsf{L}_i) = \mathbf{t}$. Therefore, for the atom $\mathsf{A} \in \mathcal{HB}_{\mathrm{P}}$ such that H matches A, it follows that $\mathsf{A} \in \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}}) \subseteq \widehat{\mathrm{T}}_{\mathcal{I}}$, by Definition 7.2.2 and Lemma 8.2.1. Thus $\mathcal{I}(\mathsf{A}) = \mathbf{t}$, and so $\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) = \mathcal{I}(\mathsf{A}) = \mathbf{t}$ by Definitions 7.2.1 and 5.3.1, a contradiction.

(2) $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_1 \wedge \ \ldots \ \wedge \mathsf{L}_\mathsf{n}) = \mathbf{u}$ and $\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) = \mathbf{f}$

It follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_\mathsf{i}) \geq \mathbf{u}$ for all $\mathsf{L}_\mathsf{i}, 1 \leq \mathsf{i} \leq \mathsf{n}$, by Definition 5.3.1. So by Lemma 5.3.1: (i) if L_i is a positive literal then $val^{\mathbf{b}}_{\widehat{\mathrm{U}}_\mathcal{I}}(\mathsf{L}_\mathsf{i}) = \mathbf{t}$; and (2) if L_i is a negative literal then $val^{\mathbf{b}}_{\widehat{\mathrm{T}}_\mathcal{I}}(\mathsf{L}_\mathsf{i}) = \mathbf{t}$. Therefore, for the atom $\mathsf{A} \in \mathcal{HB}_\mathrm{P}$ such that H matches A, it follows that $\mathsf{A} \in \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{T}}_\mathcal{I}}(\widehat{\mathrm{U}}_\mathcal{I}) \subseteq \widehat{\mathrm{U}}_\mathcal{I}$, by Definition 7.2.2 and Lemma 8.2.1. Thus $\mathcal{I}(\mathsf{A}) \geq \mathbf{u}$, and so $\mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) = \mathcal{I}(\mathsf{A}) \geq \mathbf{u}$ by Definitions 7.2.1 and 5.3.1, a contradiction.

□

Proposition 8.2.3 Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be a stable interpretation of an F-logic program P. Then $\mathcal{I}$ satisfies the positive ISA transitivity constraint.

Proof.

By Definition 6.2.1, we need to show that the following conditions hold:

(1) for all s, c: if there is x such that $\mathcal{I}(\mathsf{s} :: \mathsf{x}) = \mathbf{t}$ and $\mathcal{I}(\mathsf{x} :: \mathsf{c}) = \mathbf{t}$, then $\mathcal{I}(\mathsf{s} :: \mathsf{c}) = \mathbf{t}$;

(2) for all o, c: if there is x such that $\mathcal{I}(\mathsf{o} : \mathsf{x}) = \mathbf{t}$ and $\mathcal{I}(\mathsf{x} :: \mathsf{c}) = \mathbf{t}$, then $\mathcal{I}(\mathsf{o} : \mathsf{c}) = \mathbf{t}$.

Note that for all s, c: $\mathcal{I}(\mathsf{s} :: \mathsf{c}) = \mathbf{t}$ iff $\mathsf{s} :: \mathsf{c} \in \mathrm{T} \subseteq \widehat{\mathrm{T}}_\mathcal{I}$ and for all o, c: $\mathcal{I}(\mathsf{o} : \mathsf{c}) = \mathbf{t}$ iff $\mathsf{o} : \mathsf{c} \in \mathrm{T} \subseteq \widehat{\mathrm{T}}_\mathcal{I}$. Suppose $\mathsf{s} :: \mathsf{x} \in \mathrm{T} \subseteq \widehat{\mathrm{T}}_\mathcal{I}$ and $\mathsf{x} :: \mathsf{c} \in \mathrm{T} \subseteq \widehat{\mathrm{T}}_\mathcal{I}$. Then $\mathsf{s} :: \mathsf{c} \in \mathbf{IC}^t(\widehat{\mathrm{T}}_\mathcal{I})$ by Definition 7.2.5. It follows that $\mathsf{s} :: \mathsf{c} \in \mathbf{IC}^t(\widehat{\mathrm{T}}_\mathcal{I}) \subseteq \widehat{\mathrm{T}}_\mathcal{I}$, by Lemma 8.2.1. Similarly, if $\mathsf{o} : \mathsf{x} \in \widehat{\mathrm{T}}_\mathcal{I}$ and $\mathsf{x} :: \mathsf{c} \in \widehat{\mathrm{T}}_\mathcal{I}$, then $\mathsf{o} : \mathsf{c} \in \mathbf{IC}^t(\widehat{\mathrm{T}}_\mathcal{I}) \subseteq \widehat{\mathrm{T}}_\mathcal{I}$.
□

Proposition 8.2.4 Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be a stable interpretation of an F-logic program P. Then $\mathcal{I}$ satisfies the context consistency constraint.

Proof.

By Definition 6.2.2, we need to show that the following conditions hold:

(1) for all o, m, v: $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{value}}) = \mathbf{f}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{code}}) = \mathbf{f}$.

Note that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{value}}) = \mathbf{f}$ iff $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{value}} \notin \mathrm{T} \cup \mathrm{U}$ iff $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{value}} \notin \widehat{\mathrm{U}}_\mathcal{I}$ by Definition 8.1.1. Similarly, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{code}}) = \mathbf{f}$ iff $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{code}} \notin \widehat{\mathrm{U}}_\mathcal{I}$. Since $\mathcal{I}$ is a stable interpretation of P and so $\widehat{\mathrm{U}}_\mathcal{I} = \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\mathcal{I})$ by Definition 8.1.2, it follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{value}} \notin \widehat{\mathrm{U}}_\mathcal{I}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{code}} \notin \widehat{\mathrm{U}}_\mathcal{I}$ for all $\mathsf{o}, \mathsf{m}, \mathsf{v}$, by Lemma 7.2.4.

(2) for all c, m, v: if $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}}) = \mathbf{f}$, then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}) = \mathbf{f}$ for all o.

Let $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}}) = \mathbf{f}$. Then $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}} \notin \widehat{\mathrm{U}}_{\mathcal{I}}$. We need to show that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \notin \widehat{\mathrm{U}}_{\mathcal{I}}$ for all o. Suppose on the contrary there exists o such that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \widehat{\mathrm{U}}_{\mathcal{I}}$. Because $\mathcal{I}$ is a stable interpretation of P, $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$. It follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Lemma 7.2.4. Thus $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$ by Definition 7.2.5. So $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Lemma 8.2.1. It follows that $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}} \in \widehat{\mathrm{U}}_{\mathcal{I}}$ by Definition 7.2.5, which contradicts the premise.

(3) for all c, m: if there is no C-rule in *ground*(P) which specifies the instance method m for the class c, then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) = \mathbf{f}$ for all o, v.

Suppose on the contrary there exist o, v such that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) \neq \mathbf{f}$. Then $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}} \in \mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$. It follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Lemma 7.2.4. Thus $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$ by Definition 7.2.5 and so $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Lemma 8.2.1. So by Definition 7.2.5 there must exist a C-rule in *ground*(P) which specifies the instance method m for the class c, a contradiction.

(4) for all o, m: if $\mathsf{o}[\mathsf{m}]$ is a strong local context, then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{f}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) = \mathbf{f}$ for all v, c.

Let $\mathsf{o}[\mathsf{m}]$ be a strong local context. Then there must exist v such that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\text{local}} \in \mathrm{T} \subseteq \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 6.1.1, and so $lc(\mathsf{o}, \mathsf{m}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definition 7.2.3. Suppose on the contrary there exist v, c such that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) \neq \mathbf{f}$. Then $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$. It follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Lemma 7.2.4. Thus $lc(\mathsf{o}, \mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definition 7.2.5, a contradiction. Similarly, we can show that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) = \mathbf{f}$ for all v, c.

□

Proposition 8.2.5 Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be a stable interpretation of an F-logic program P. Then $\mathcal{I}$ satisfies the unique source inheritance constraint.

Proof.

By Definition 6.2.3, we need to show that the following conditions hold:

(1) for all $\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}$: if $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{t}$ or $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) = \mathbf{t}$, then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{x}}_{\text{value}}) = \mathbf{f}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{x}}_{\text{code}}) = \mathbf{f}$ for all z, x such that $\mathsf{x} \neq \mathsf{c}$.

Because $\mathcal{I}$ is a stable interpretation of P, $\widehat{\mathrm{T}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ and $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definition 8.1.2. If $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{t}$, then $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \mathrm{T} \subseteq \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 8.1.1. So $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Lemma 7.2.4. It follows that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 7.2.5. On the other hand, if

$\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) = \mathbf{t}$, then $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}} \in \mathrm{T} \subseteq \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 8.1.1. So $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Lemma 7.2.4. It follows that $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 7.2.5. Therefore, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{t}$ or $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) = \mathbf{t}$ implies $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ or $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$.

Suppose on the contrary there are z, x such that $\mathsf{x} \neq \mathsf{c}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{x}}_{\text{value}}) \geq \mathbf{u}$. Then $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{x}}_{\text{value}} \in \mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$ by Definition 8.1.1. So $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{value}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Lemma 7.2.4. Therefore, $mc(\mathsf{x}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definition 7.2.5. Since $\mathsf{x} \neq \mathsf{c}$, it follows that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \notin \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 7.2.3, which is a contradiction. Therefore, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{x}}_{\text{value}}) = \mathbf{f}$ for all z, x such that $\mathsf{x} \neq \mathsf{c}$. Similarly, we can also show that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{x}}_{\text{code}}) = \mathbf{f}$ for all z, x such that $\mathsf{x} \neq \mathsf{c}$.

(2) for all c, m, o: if $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{value}}) = \mathbf{f}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{code}}) = \mathbf{f}$ for all v, x such that $\mathsf{x} \neq \mathsf{c}$.

Let $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$. Suppose on the contrary there exist v, x such that $\mathsf{x} \neq \mathsf{c}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{value}} \neq \mathbf{f}$. Then $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{value}} \in \mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$ by Definition 8.1.1. Because $\mathcal{I}$ is a stable interpretation of P, $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definition 8.1.2. So $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{value}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Lemma 7.2.4. It follows that $mc(\mathsf{x}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definition 7.2.5. However, $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ or $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 8.1.1. Since $\mathsf{x} \neq \mathsf{c}$, it follows that $mc(\mathsf{x}, \mathsf{m}, \mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definition 7.2.3, which is a contradiction. Therefore, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{value}}) = \mathbf{f}$ for all v, x such that $\mathsf{x} \neq \mathsf{c}$. Similarly, we can also show that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\text{code}}) = \mathbf{f}$ for all v, x such that $\mathsf{x} \neq \mathsf{c}$.

(3) for all o, m, v, c: $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{t}$ *iff*

(i) $\mathsf{o}[\mathsf{m}]$ is neither a strong nor a weak local context; and
(ii) $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$; and
(iii) $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}}) = \mathbf{t}$; and
(iv) there is no x such that $\mathsf{x} \neq \mathsf{c}$ and $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$.

"$\Rightarrow$". Because $\mathcal{I}$ is a stable interpretation of P, $\widehat{\mathrm{T}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Definition 8.1.2. Because $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{t}$, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \mathrm{T} \subseteq \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 8.1.1. Thus $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Lemma 7.2.4, and so $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}} \in \widehat{\mathrm{T}}_{\mathcal{I}}$, $lc(\mathsf{o}, \mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$, and $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$, by Definition 7.2.5. Because $lc(\mathsf{o}, \mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$, it follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{x}]^{\mathsf{o}}_{\text{local}} \notin \widehat{\mathrm{U}}_{\mathcal{I}}$ for all x, by Definition 7.2.3. Thus $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{x}]^{\mathsf{o}}_{\text{local}}) = \mathbf{f}$ for all x and so $\mathsf{o}[\mathsf{m}]$ is neither a strong nor a weak local context, by Definition 6.1.1. Because $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$, it follows that $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$

by Definition 8.1.1. $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ implies $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}}) = \mathbf{t}$. Because $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$, it follows that there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$, by Definition 7.2.3. So there is no x such that $\mathsf{x} \neq \mathsf{c}$ and $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$, by Definition 8.1.1.

"$\Leftarrow$". Because $\mathsf{o}[\mathsf{m}]$ is neither a strong nor a weak local context, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{x}]^{\circ}_{\text{local}}) = \mathbf{f}$ for all x, by Definition 6.1.1. It follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{x}]^{\circ}_{\text{local}} \notin \mathrm{T} \cup \mathrm{U}$ for all x, and so $lc(\mathsf{o}, \mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$, by Definitions 8.1.1 and 7.2.3. Because $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, therefore $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 8.1.1. Since $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}}) = \mathbf{t}$, it follows that $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{local}} \in \mathrm{T} \subseteq \widehat{\mathrm{T}}_{\mathcal{I}}$. Because $\mathcal{I}$ is a stable interpretation of P, therefore $\widehat{\mathrm{T}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$, by Definition 8.1.2. So if we can show that $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$, then it follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \mathbf{IC}^{i}_{\mathrm{P}, \widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}}) \subseteq \widehat{\mathrm{T}}_{\mathcal{I}}$, by Definition 7.2.5 and Lemma 8.2.1.

Suppose on the contrary $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$. Then by Definition 7.2.3, there is $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$. It follows that $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$ by Definition 8.1.1, which contradicts the premise. Therefore, $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$, and so $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}} \in \widehat{\mathrm{T}}_{\mathcal{I}}$, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{value}}) = \mathbf{t}$.

□

Proposition 8.2.6 Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be a stable interpretation of an F-logic program P. Then $\mathcal{I}$ satisfies the C-rules of P.

Proof. By contradiction.

Because $\mathcal{I}$ is a stable interpretation of P, $\widehat{\mathrm{T}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ and $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definition 8.1.2. Suppose on the contrary $\mathcal{I}$ does not satisfy the C-rules of P. Then by Definition 6.3.5, there are an object $\mathsf{o} \in \mathcal{HU}_{\mathrm{P}}$ and a C-rule, $\mathsf{R} \equiv \mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{B}$ or $\mathsf{R} \equiv \mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$, in $ground(\mathrm{P})$, such that $\mathcal{I}(\mathsf{R}||\mathsf{o}) = \mathbf{f}$. Let us assume that $\mathsf{R} \equiv \mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{B}$ (the case of $\mathsf{R} \equiv \mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$ is similar). By Definition 6.3.4, we have the following cases to consider:

(1) $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{t}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) < \mathcal{V}^{\mathfrak{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}})$

Because $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{t}$, therefore by Definition 6.3.2: (i) $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ and so $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 8.1.1; (ii) $lc(\mathsf{o}, \mathsf{m})$ is neither a strong nor a weak local context and so $lc(\mathsf{o}, \mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Definitions 7.2.3 and 6.1.1; and (iii) there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$. It follows that there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \notin \widehat{\mathrm{U}}_{\mathcal{I}}$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \notin \widehat{\mathrm{U}}_{\mathcal{I}}$ by Definition 8.1.1. Thus $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$. Since $\widehat{\mathrm{T}}_{\mathcal{I}} \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$, it also follows that $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$, $lc(\mathsf{o}, \mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, and $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, by the monotonicity of $\mathbf{IB}_{\mathrm{P}}$.

First let us assume that $\mathcal{V}^{\mathfrak{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{t}$. Then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\text{code}}) \neq \mathbf{t}$. Since $\mathcal{V}^{\mathfrak{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{t}$, it follows that $\mathcal{V}^{\mathfrak{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$ for all $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$, by Definition 5.3.1. So by Lemma 5.3.1: (i) if L is a positive literal then

$val^{\mathbf{b}}_{\widehat{\mathrm{T}}_{\mathcal{I}}}(\mathsf{L}) = \mathbf{t}$; and (ii) if L is a negative literal then $val^{\mathbf{b}}_{\widehat{\mathrm{U}}_{\mathcal{I}}}(\mathsf{L}) = \mathbf{t}$. Therefore, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{U}}_{\mathcal{I}}}(\widehat{\mathrm{T}}_{\mathcal{I}}) \subseteq \widehat{\mathrm{T}}_{\mathcal{I}}$, by Definition 7.2.4 and Lemma 8.2.1. Thus $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{t}$, a contradiction.

On the other hand, if $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{u}$, then $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{f}$. Since $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{u}$, it follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) \geq \mathbf{u}$ for all $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$, by Definition 5.3.1. So by Lemma 5.3.1: (i) if L is a positive literal then $val^{\mathbf{b}}_{\widehat{\mathrm{U}}_{\mathcal{I}}}(\mathsf{L}) = \mathbf{t}$; and (2) if L is a negative literal then $val^{\mathbf{b}}_{\widehat{\mathrm{T}}_{\mathcal{I}}}(\mathsf{L}) = \mathbf{t}$. Therefore, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}}) \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$, by Definition 7.2.2 and Lemma 8.2.1. Thus $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) \geq \mathbf{u}$, a contradiction.

(2) $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{u}$, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{f}$, and $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}) \geq \mathbf{u}$

Because $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{u}$, therefore by Definition 6.3.3: (i) $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, and so $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$ by Definition 8.1.1; (ii) $lc(\mathsf{o},\mathsf{m})$ is not a strong local context and so $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definitions 7.2.3 and 6.1.1; and (iii) there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$. It follows that there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \notin \widehat{\mathrm{T}}_{\mathcal{I}}$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \notin \widehat{\mathrm{T}}_{\mathcal{I}}$ by Definition 8.1.1. Thus $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$.

Since $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}) \geq \mathbf{u}$, it follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) \geq \mathbf{u}$ for all $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$, by Definition 5.3.1. So by Lemma 5.3.1: (i) if L is a positive literal then $val^{\mathbf{b}}_{\widehat{\mathrm{U}}_{\mathcal{I}}}(\mathsf{L}) = \mathbf{t}$; and (2) if L is a negative literal then $val^{\mathbf{b}}_{\widehat{\mathrm{T}}_{\mathcal{I}}}(\mathsf{L}) = \mathbf{t}$. Therefore, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}}) \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$, by Definition 7.2.2 and Lemma 8.2.1. Thus $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) \geq \mathbf{u}$, a contradiction.

(3) $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{f}$ and $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) \geq \mathbf{u}$

Because $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) \geq \mathbf{u}$, therefore $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$. Thus $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Lemma 7.2.4. So by Definition 7.2.4, $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$, $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, and $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$. Because $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$, so $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, by Definition 8.1.1. Since $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, therefore $lc(\mathsf{o},\mathsf{m})$ is not a strong local context, by Definitions 7.2.3 and 6.1.1. Because $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, so there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ or $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$. It follows that there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, by Definition 8.1.1. Thus o must either weakly or strongly inherit R, by Definitions 6.3.3 and 6.3.2. Therefore, $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) \geq \mathbf{u}$, a contradiction.

□

Proposition 8.2.7 Let $\mathcal{I} = \langle \mathrm{T}; \mathrm{U} \rangle$ be a stable interpretation of an F-logic program P. Then $\mathcal{I}$ satisfies the optimistic ISA transitivity constraint.

Proof.

By Definition 6.5.1, we need to show that the following conditions hold:

(1) for all s, c: if there is x such that $\mathcal{I}(\mathsf{s}::\mathsf{x} \wedge \mathsf{x}::\mathsf{c}) = \mathbf{u}$ and $\mathcal{I}(\mathsf{s}::\mathsf{c}) \neq \mathbf{t}$, then $\mathcal{I}(\mathsf{s}::\mathsf{c}) = \mathbf{u}$;

(2) for all o, c: if there is x such that $\mathcal{I}(\mathsf{o}:\mathsf{x} \wedge \mathsf{x}::\mathsf{c}) = \mathbf{u}$ and $\mathcal{I}(\mathsf{o}:\mathsf{c}) \neq \mathbf{t}$, then $\mathcal{I}(\mathsf{o}:\mathsf{c}) = \mathbf{u}$.

Suppose $\mathcal{I}(\mathsf{s}::\mathsf{x} \wedge \mathsf{x}::\mathsf{c}) = \mathbf{u}$. Then $\mathsf{s}::\mathsf{x} \in \mathrm{T} \cup \mathrm{U}$ and $\mathsf{x}::\mathsf{c} \in \mathrm{T} \cup \mathrm{U}$. It follows that $\mathsf{s}::\mathsf{x} \in \widehat{\mathrm{U}}_{\mathcal{I}}$ and $\mathsf{x}::\mathsf{c} \in \widehat{\mathrm{U}}_{\mathcal{I}}$, by Definition 8.1.1. So $\mathsf{s}::\mathsf{c} \in \mathbf{IC}^t(\widehat{\mathrm{U}}_{\mathcal{I}})$ by Definition 7.2.5. Since $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ by Definition 8.1.2, it follows that $\mathsf{s}::\mathsf{c} \in \mathbf{IC}^t(\widehat{\mathrm{U}}_{\mathcal{I}}) \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$, by Lemma 8.2.1. Thus $\mathcal{I}(\mathsf{s}::\mathsf{c}) \geq \mathbf{u}$. But $\mathcal{I}(\mathsf{s}::\mathsf{c}) \neq \mathbf{t}$. It follows that $\mathcal{I}(\mathsf{s}::\mathsf{c}) = \mathbf{u}$. Similarly, if $\mathcal{I}(\mathsf{o}:\mathsf{x} \wedge \mathsf{x}::\mathsf{c}) = \mathbf{u}$ and $\mathcal{I}(\mathsf{o}:\mathsf{c}) \neq \mathbf{t}$, then $\mathcal{I}(\mathsf{o}:\mathsf{c}) = \mathbf{u}$.
□

Proposition 8.2.8 Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be a stable interpretation of an F-logic program P. Then $\mathcal{I}$ satisfies the optimistic inheritance constraint.

Proof.

By Definition 6.5.2, we need to show that for all o, m, v, c: $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}) = \mathbf{u}$ *iff* the following conditions hold:

(i) $\mathsf{o}[\mathsf{m}]$ is not a strong local context;

(ii) $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$;

(iii) $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}}) \geq \mathbf{u}$;

(iv) there is no $\mathsf{x} \neq$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$;

(v) $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}) \neq \mathbf{t}$.

"$\Longrightarrow$". Because $\mathcal{I}$ is a stable interpretation of P, $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, by Definition 8.1.2. Because $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} = \mathbf{u}$, therefore $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$, by Definition 8.1.1. Thus $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \mathbf{IC}^i_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}})$, by Lemma 7.2.4. So $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{U}}_{\mathcal{I}}$, $lc(\mathsf{o}, \mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, and $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, by Definition 7.2.5. Because $lc(\mathsf{o}, \mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, it follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{x}]^{\mathsf{o}}_{\mathrm{local}} \notin \widehat{\mathrm{T}}_{\mathcal{I}}$ for all x, by Definition 7.2.3. So $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\mathrm{local}}) \neq \mathbf{t}$ for all x. Thus $\mathsf{o}[\mathsf{m}]$ is not a strong local context by Definition 6.1.1. Because $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$, it follows that $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, by Definition 8.1.1. $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{U}}_{\mathcal{I}}$ implies $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}}) \geq \mathbf{u}$. Because $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, it follows that there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ or $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$. So there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, by Definition 8.1.1.

"$\Longleftarrow$". Because $\mathsf{o}[\mathsf{m}]$ is not a strong local context, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{x}]^{\mathsf{o}}_{\mathrm{local}}) \neq \mathbf{t}$ for all x, by Definition 6.1.1. It follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{x}]^{\mathsf{o}}_{\mathrm{local}} \notin \mathrm{T}$ for all x, and so

$lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, by Definitions 8.1.1 and 7.2.3. Because $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, therefore $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_{\mathcal{I}}$ by Definition 8.1.1. Since $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{local}}) \geq \mathbf{u}$, it follows that $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{local}} \in \mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$. Because $\mathcal{I}$ is a stable interpretation of P, therefore $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, by Definition 8.1.2. So if we can show that $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, then it follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{value}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{T}}_{\mathcal{I}}}(\widehat{\mathrm{U}}_{\mathcal{I}}) \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$, by Definition 7.2.5 and Lemma 8.2.1. Suppose on the contrary $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$. Then by Definition 7.2.3, there is $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_{\mathcal{I}}$. It follows that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, by Definition 8.1.1, which contradicts the premise. Therefore, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, and so $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{value}} \in \widehat{\mathrm{U}}_{\mathcal{I}}$, $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{value}}) \geq \mathbf{u}$. But $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{value}}) \neq \mathbf{t}$. So $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{value}}) = \mathbf{u}$.
□

Theorem 8.2.9 Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be a stable interpretation of an F-logic program P. Then $\mathcal{I}$ is an object model of P. Moreover, $\mathcal{I}$ satisfies the optimistic ISA transitivity constraint and the optimistic inheritance constraint.

Proof.

By Definition 6.4.1, Propositions 8.2.2, 8.2.3, 8.2.4, 8.2.5, 8.2.6, 8.2.7, and 8.2.8.

□

Clearly, by Theorem 8.2.9, a stable interpretation indeed satisfies all the requirements of an object model. Therefore, from now on a stable interpretation is also called a *stable object model*.

8.3 Stable Object Models and Fixpoints

There is an interesting correspondence between stable object models and fixpoints of $\mathbf{F}_{\mathrm{P}}$. On one hand, stable object models are essentially fixpoints of $\mathbf{F}_{\mathrm{P}}$. Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be a stable object model of an F-logic program P. Then $\widehat{\mathrm{T}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ and $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$, by Definition 8.1.2. It follows that $\widehat{\mathrm{T}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}}) = \Psi_{\mathrm{P}}(\Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})) = \mathbf{F}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$ and so $\widehat{\mathrm{T}}_{\mathcal{I}}$ is a fixpoint of $\mathbf{F}_{\mathrm{P}}$. Similarly, $\widehat{\mathrm{U}}_{\mathcal{I}}$ is also a fixpoint of $\mathbf{F}_{\mathrm{P}}$. Moreover, $\widehat{\mathrm{T}}_{\mathcal{I}} \subseteq \widehat{\mathrm{U}}_{\mathcal{I}}$ by Definition 8.1.1.

The following proposition shows that stable object models can be constructed using certain fixpoints of $\mathbf{F}_{\mathrm{P}}$.

Proposition 8.3.1 Given an F-logic program P, let $\widehat{\mathrm{J}}$ be a fixpoint of $\mathbf{F}_{\mathrm{P}}$, $\widehat{\mathrm{K}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{J}})$, and $\widehat{\mathrm{J}} \subseteq \widehat{\mathrm{K}}$. Then $\mathcal{I} = \langle\, \pi(\widehat{\mathrm{J}}); \pi(\widehat{\mathrm{K}}) - \pi(\widehat{\mathrm{J}}) \,\rangle$, where π is the projection function defined in Section 7.1, is a stable object model of P.

Proof.

Let $\mathrm{T} = \pi(\widehat{\mathrm{J}})$ and $\mathrm{U} = \pi(\widehat{\mathrm{K}}) - \pi(\widehat{\mathrm{J}})$. Thus $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$. Since $\widehat{\mathrm{J}} \subseteq \widehat{\mathrm{K}}$, it follows that $\pi(\widehat{\mathrm{J}}) \subseteq \pi(\widehat{\mathrm{K}})$ by Lemma 7.1.1. So $\mathrm{T} \cup \mathrm{U} = \pi(\widehat{\mathrm{K}})$.

To show that $\mathcal{I}$ is a stable object model of P, we need to show that $\widehat{\mathrm{T}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\mathcal{I}})$ and $\widehat{\mathrm{U}}_{\mathcal{I}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\mathcal{I}})$. Since $\widehat{\mathrm{J}}$ is a fixpoint of $\mathbf{F}_{\mathrm{P}}$ and $\widehat{\mathrm{K}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{J}})$, it follows that $\widehat{\mathrm{J}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{K}})$, by Definition 7.2.8. Therefore, if we can show that $\widehat{\mathrm{T}}_{\mathcal{I}} = \widehat{\mathrm{J}}$ and $\widehat{\mathrm{U}}_{\mathcal{I}} = \widehat{\mathrm{K}}$, then it follows that $\mathcal{I}$ is a stable object model of P.

Since $\widehat{\mathrm{J}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{K}}) = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{K}}})$ and $\widehat{\mathrm{K}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{J}}) = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{J}}})$, by Definitions 7.2.6 and 7.2.5 it follows that

$$\begin{aligned}
\widehat{\mathrm{J}} &= \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{K}}}(\widehat{\mathrm{J}}) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{K}}}(\widehat{\mathrm{J}}) \cup \mathbf{IC}^{t}(\widehat{\mathrm{J}}) \cup \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{K}}}(\widehat{\mathrm{J}}) \cup \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{K}}}(\widehat{\mathrm{J}}) \\
\widehat{\mathrm{K}} &= \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{K}}) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{K}}) \cup \mathbf{IC}^{t}(\widehat{\mathrm{K}}) \cup \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{K}}) \cup \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{J}}}(\widehat{\mathrm{K}})
\end{aligned}$$

First we will show that for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$ iff $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$. Indeed, $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$, iff $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{K}}}(\widehat{\mathrm{J}})$, iff $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o} : \mathsf{c} \in \widehat{\mathrm{J}}$, $\mathsf{c}[\mathsf{m} \longrightarrow\!\!\!\!\!\rightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{J}}$ for some v, and $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{K}})$, by Definition 7.2.5, iff $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o} : \mathsf{c} \in \pi(\widehat{\mathrm{J}})$, $\mathsf{c}[\mathsf{m} \longrightarrow\!\!\!\!\!\rightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \pi(\widehat{\mathrm{J}})$ for some v, and $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\pi(\widehat{\mathrm{K}}))$, iff $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o} : \mathsf{c} \in \mathrm{T}$, $\mathsf{c}[\mathsf{m} \longrightarrow\!\!\!\!\!\rightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \mathrm{T}$ for some v, and $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T} \cup \mathrm{U})$, iff $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, by Lemma 7.2.2.

Similarly, we can also show that (i) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$ iff $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$; and (ii) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}$ or $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}$ iff $\mathsf{c}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$. Therefore, it follows that $\widehat{\mathrm{T}}_{\mathcal{I}} = \widehat{\mathrm{J}}$ and $\widehat{\mathrm{U}}_{\mathcal{I}} = \widehat{\mathrm{K}}$ by Definition 8.1.1 and so completes the proof. □

It is worth pointing out that the condition $\widehat{\mathrm{J}} \subseteq \widehat{\mathrm{K}}$ in the above Proposition 8.3.1 is *not* necessary to construct a stable object model out of the extended sets $\widehat{\mathrm{J}}$ and $\widehat{\mathrm{K}}$. In fact, as illustrated by the following example, we can have an F-logic program P such that $\widehat{\mathrm{J}}$ is a fixpoint of $\mathbf{F}_{\mathrm{P}}$, $\widehat{\mathrm{K}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{J}})$, and $\widehat{\mathrm{J}} \not\subseteq \widehat{\mathrm{K}}$, but $\mathcal{I} = \langle\, \pi(\widehat{\mathrm{J}}); \pi(\widehat{\mathrm{K}}) - \pi(\widehat{\mathrm{J}}) \,\rangle$ is a stable object model of P.

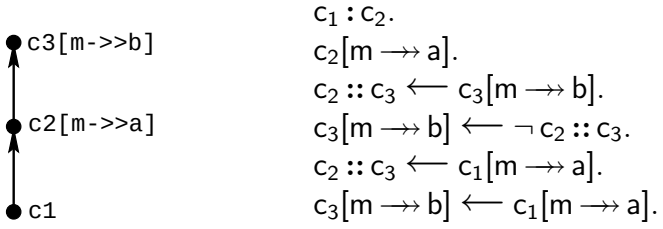


Figure 8: Constructive Fixpoints

Example 8.3.1 Consider the F-logic program P in Figure 8 and the following two extended sets $\widehat{\mathrm{J}}$ and $\widehat{\mathrm{K}}$:

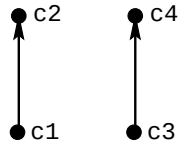
$$\begin{aligned}\widehat{\mathrm{J}} &= \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{local}}, \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{value}}, \mathsf{c}_2 :: \mathsf{c}_3, \mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c}_3}_{\mathrm{local}}\} \cup \\ &\quad \{\mathsf{c}_2[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{c}_1, \mathsf{c}_3[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{c}_1\} \\ \widehat{\mathrm{K}} &= \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{local}}\} \cup \{\mathsf{c}_2[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{c}_1\}\end{aligned}$$

We can verify that $\widehat{\mathrm{J}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{K}})$, $\widehat{\mathrm{K}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{J}})$, and so $\widehat{\mathrm{J}}$ is a fixpoint of Ψ_{P}. Moreover,

$$\begin{aligned}&\pi(\widehat{\mathrm{J}}) = \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{local}}, \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{value}}, \mathsf{c}_2 :: \mathsf{c}_3, \mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c}_3}_{\mathrm{local}}\} \\ &\pi(\widehat{\mathrm{K}}) - \pi(\widehat{\mathrm{J}}) = \emptyset\end{aligned}$$

We can also verify that the interpretation $\mathcal{I} = \langle\, \pi(\widehat{\mathrm{J}}); \pi(\widehat{\mathrm{K}}) - \pi(\widehat{\mathrm{J}}) \,\rangle$ is a stable object model of P. But clearly $\widehat{\mathrm{J}} - \widehat{\mathrm{K}} \neq \emptyset$. Thus $\widehat{\mathrm{J}} \not\subseteq \widehat{\mathrm{K}}$. □

Another interesting question is whether we can *always* construct stable object models of an F-logic program P out of fixpoints of Ψ_{P}. The answer turns out to be *no*. As illustrated by the following example, we may not even be able to construct an object model out of some fixpoints of Ψ_{P}.



$\mathsf{c}_1 : \mathsf{c}_2.$
$\mathsf{c}_3 : \mathsf{c}_4.$
$\mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}] \longleftarrow \neg\, \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}].$
$\mathsf{c}_4[\mathsf{m} \twoheadrightarrow \mathsf{b}] \longleftarrow \neg\, \mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{b}].$
$\mathsf{c}_4[\mathsf{m} \twoheadrightarrow \mathsf{c}] \longleftarrow \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}], \mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{b}].$

Figure 9: Nonconstructive Fixpoints

Example 8.3.2 Consider the F-logic program P in Figure 9 and the following two extended sets $\widehat{\mathrm{J}}$ and $\widehat{\mathrm{K}}$:

$$\begin{aligned}\widehat{\mathrm{J}} &= \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_3 : \mathsf{c}_4, \mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{local}}, \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{value}}\} \cup \{\mathsf{c}_2[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{c}_1\} \\ \widehat{\mathrm{K}} &= \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_3 : \mathsf{c}_4, \mathsf{c}_4[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c}_4}_{\mathrm{local}}, \mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c}_4}_{\mathrm{value}}\} \cup \{\mathsf{c}_4[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{c}_3\}\end{aligned}$$

We can verify that $\widehat{\mathrm{J}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{K}})$, $\widehat{\mathrm{K}} = \Psi_{\mathrm{P}}(\widehat{\mathrm{J}})$, and so $\widehat{\mathrm{J}}$ is a fixpoint of Ψ_{P}. However,

$$\begin{aligned}&\pi(\widehat{\mathrm{J}}) = \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_3 : \mathsf{c}_4, \mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{local}}, \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{value}}\} \\ &\pi(\widehat{\mathrm{K}}) - \pi(\widehat{\mathrm{J}}) = \{\mathsf{c}_4[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c}_4}_{\mathrm{local}}, \mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c}_4}_{\mathrm{value}}\}\end{aligned}$$

It is easy to check that the interpretation $\mathcal{I} = \langle\, \pi(\widehat{\mathrm{J}}); \pi(\widehat{\mathrm{K}}) - \pi(\widehat{\mathrm{J}}) \,\rangle$ is not even an object model of P, because $\mathcal{I}$ does not satisfy the program in Figure 9, namely, the last rule of the program in Figure 9. But if we eliminate the last from the program in Figure 9 and get a new program, then $\mathcal{I}$ would be an object model, but *not* a stable object model, of this new program. □

Chapter 9

Optimistic Object Models

In this chapter we will introduce a particular object model, called *optimistic object model*, which exists for *any* F-logic program. We will show that the optimistic object model is a stable object model and thus satisfies all the V-rules and C-rules of an F-logic program plus the core and optimistic inheritance constraints. Finally, we will introduce a partial order, called *information ordering*, among object models. We will show that the optimistic object model is the *least* stable object model with respect to information ordering.

9.1 Definitions and Properties

Definition 9.1.1 (Optimistic Object Model) The *optimistic object model*, $\mathcal{M}$, of an F-logic program P is defined as follows:

$$\begin{array}{rcl} \mathcal{M} & \stackrel{\text{def}}{=} & \langle\, \mathrm{T}; \mathrm{U} \,\rangle \\ \mathrm{T} & = & \pi(\mathrm{lfp}(\mathbf{F}_\mathrm{P})) \\ \mathrm{U} & = & \pi(\Psi_\mathrm{P}(\mathrm{lfp}(\mathbf{F}_\mathrm{P}))) - \pi(\mathrm{lfp}(\mathbf{F}_\mathrm{P})) \end{array}$$

where π is the projection operator defined earlier. It removes the auxiliary atoms of the forms $\mathsf{c[m]} \overset{v}{\rightsquigarrow} \mathsf{o}$ and $\mathsf{c[m]} \overset{c}{\rightsquigarrow} \mathsf{o}$, which are used for book-keeping inheritance candidacy information during computation.

Definition 9.1.1 gives a procedural definition as well as characterization of optimistic object models. Note that $\mathrm{lfp}(\mathbf{F}_\mathrm{P})$ is *unique* and *always* exists given an F-logic program P . Therefore, the optimistic object model is uniquely defined for any F-logic program.

To show the properties of optimistic object models, we need to introduce the intermediate results of fixpoint computation.

Definition 9.1.2 Let α range over all countable ordinals. The sets $\widehat{\mathrm{T}}_\alpha$, $\widehat{\mathrm{U}}_\alpha$, $\widehat{\mathrm{T}}_\infty$, and $\widehat{\mathrm{U}}_\infty$, which are extended atom sets of an F-logic program P, are defined as follows:

$$\begin{array}{lcllcll}
\widehat{\mathrm{T}}_0 & = & \emptyset & \widehat{\mathrm{U}}_0 & = & \Psi_\mathrm{P}(\widehat{\mathrm{T}}_0) & \text{for limit ordinal } 0 \\
\widehat{\mathrm{T}}_\alpha & = & \Psi_\mathrm{P}(\widehat{\mathrm{U}}_{\alpha-1}) & \widehat{\mathrm{U}}_\alpha & = & \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\alpha) & \text{for successor ordinal } \alpha \\
\widehat{\mathrm{T}}_\alpha & = & \bigcup_{\beta<\alpha} \widehat{\mathrm{T}}_\beta & \widehat{\mathrm{U}}_\alpha & = & \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\alpha) & \text{for limit ordinal } \alpha \neq 0 \\
\widehat{\mathrm{T}}_\infty & = & \bigcup_{\alpha} \widehat{\mathrm{T}}_\alpha & \widehat{\mathrm{U}}_\infty & = & \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\infty) &
\end{array}$$

Given an F-logic program P, the power set of its extended Herbrand base $\widehat{\mathcal{HB}}_\mathrm{P}$ constitutes a complete lattice where the partial order is defined by set inclusion. Therefore, Propositions 4.1.2 and 4.1.3 apply to any monotonic operator defined on the power set of $\widehat{\mathcal{HB}}_\mathrm{P}$.

Lemma 9.1.1 Let α and β range over all countable ordinals:

(1) for all α, β: if $\alpha < \beta$ then $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{T}}_\beta$

(2) $\widehat{\mathrm{T}}_\infty = \mathrm{lfp}(\mathbf{F}_\mathrm{P})$

(3) for all α: $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{T}}_\infty$

(4) $\widehat{\mathrm{U}}_\infty = \mathrm{gfp}(\mathbf{F}_\mathrm{P})$

(5) for all α: $\widehat{\mathrm{U}}_\alpha \supseteq \widehat{\mathrm{U}}_\infty$

(6) for all α, β: if $\alpha < \beta$ then $\widehat{\mathrm{U}}_\alpha \supseteq \widehat{\mathrm{U}}_\beta$

(7) for all α: $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{U}}_\alpha$

(8) for all α, β: $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{U}}_\beta$

Proof.

(1) for all α, β: if $\alpha < \beta$ then $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{T}}_\beta$

By Definition 9.1.2, for a successor ordinal α:

$$\widehat{\mathrm{T}}_\alpha = \Psi_\mathrm{P}(\widehat{\mathrm{U}}_{\alpha-1}) = \Psi_\mathrm{P}(\Psi_\mathrm{P}(\widehat{\mathrm{T}}_{\alpha-1})) = \mathbf{F}_\mathrm{P}(\widehat{\mathrm{T}}_{\alpha-1})$$

Since $\mathbf{F}_\mathrm{P}$ is monotonic by Lemma 7.2.6, the result directly follows by Proposition 4.1.3.

(2) $\widehat{\mathrm{T}}_\infty = \mathrm{lfp}(\mathbf{F}_\mathrm{P})$

By Proposition 4.1.3.

(3) for all α: $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{T}}_\infty$

By Proposition 4.1.3.

(4) $\widehat{\mathrm{U}}_\infty = \mathrm{gfp}(\mathbf{F}_\mathrm{P})$

$\mathbf{F}_\mathrm{P}(\widehat{\mathrm{U}}_\infty) = \Psi_\mathrm{P}(\Psi_\mathrm{P}(\Psi_\mathrm{P}(\widehat{\mathrm{T}}_\infty))) = \Psi_\mathrm{P}(\mathbf{F}_\mathrm{P}(\widehat{\mathrm{T}}_\infty)) = \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\infty) = \widehat{\mathrm{U}}_\infty$. It follows that $\widehat{\mathrm{U}}_\infty$ is a fixpoint of $\mathbf{F}_\mathrm{P}$. Similarly, $\Psi_\mathrm{P}(\mathrm{gfp}(\mathbf{F}_\mathrm{P}))$ is also a fixpoint of $\mathbf{F}_\mathrm{P}$. Thus $\Psi_\mathrm{P}(\mathrm{gfp}(\mathbf{F}_\mathrm{P})) \supseteq \widehat{\mathrm{T}}_\infty$, and so $\Psi_\mathrm{P}(\Psi_\mathrm{P}(\mathrm{gfp}(\mathbf{F}_\mathrm{P}))) \subseteq \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\infty)$, $\mathrm{gfp}(\mathbf{F}_\mathrm{P}) \subseteq \widehat{\mathrm{U}}_\infty$, by the antimonotonicity of Ψ_P. Therefore, $\widehat{\mathrm{U}}_\infty$ is the greatest fixpoint of $\mathbf{F}_\mathrm{P}$.

(5) for all α: $\widehat{\mathrm{U}}_\alpha \supseteq \widehat{\mathrm{U}}_\infty$

By Definition 9.1.2, (3), and the antimonotonicity of Ψ_P.

(6) for all α, β: if $\alpha < \beta$ then $\widehat{\mathrm{U}}_\alpha \supseteq \widehat{\mathrm{U}}_\beta$

By Definition 9.1.2, (1), and the antimonotonicity of Ψ_P.

(7) for all α: $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{U}}_\alpha$

By (3), $\widehat{\mathrm{T}}_\infty \supseteq \widehat{\mathrm{T}}_\alpha$. So $\Psi_\mathrm{P}(\widehat{\mathrm{T}}_\infty) \subseteq \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\alpha)$ by the antimonotonicity of Ψ_P. Thus $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{T}}_\infty \subseteq \widehat{\mathrm{U}}_\infty = \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\infty) \subseteq \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\alpha) = \widehat{\mathrm{U}}_\alpha$.

(8) for all α, β: $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{U}}_\beta$

If $\alpha = \beta$, then $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{U}}_\beta$ by (7). If $\alpha < \beta$, then $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{T}}_\beta \subseteq \widehat{\mathrm{U}}_\beta$, by (1) and (7). If $\alpha > \beta$, then $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{U}}_\alpha \subseteq \widehat{\mathrm{U}}_\beta$, by (6) and (7).

□

Therefore, by Definition 9.1.1, Lemma 9.1.1, and the definition of the projection function π in Section 7.1, we can reformulate the optimistic object model in the following lemma.

Lemma 9.1.2 The optimistic object model, $\mathcal{M}$, of an F-logic program P is defined as follows:

$$\mathcal{M} = \langle\, \pi(\widehat{\mathrm{T}}_\infty); \pi(\widehat{\mathrm{U}}_\infty) - \pi(\widehat{\mathrm{T}}_\infty) \,\rangle = \langle\, \pi(\widehat{\mathrm{T}}_\infty); \pi(\widehat{\mathrm{U}}_\infty - \widehat{\mathrm{T}}_\infty) \,\rangle$$

Lemma 9.1.3 Let α range over all successor ordinals and β range over all countable ordinals:

$$\begin{aligned}
\widehat{\mathrm{T}}_\alpha &= \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{T}}_\alpha) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{T}}_\alpha) \cup \mathbf{IC}^t(\widehat{\mathrm{T}}_\alpha) \cup \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{T}}_\alpha) \cup \mathbf{IC}^i_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{T}}_\alpha) \\
\widehat{\mathrm{U}}_\beta &= \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{T}}_\beta}(\widehat{\mathrm{U}}_\beta) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{T}}_\beta}(\widehat{\mathrm{U}}_\beta) \cup \mathbf{IC}^t(\widehat{\mathrm{U}}_\beta) \cup \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{T}}_\beta}(\widehat{\mathrm{U}}_\beta) \cup \mathbf{IC}^i_{\mathrm{P},\widehat{\mathrm{T}}_\beta}(\widehat{\mathrm{U}}_\beta) \\
\widehat{\mathrm{T}}_\infty &= \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{U}}_\infty}(\widehat{\mathrm{T}}_\infty) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{U}}_\infty}(\widehat{\mathrm{T}}_\infty) \cup \mathbf{IC}^t(\widehat{\mathrm{T}}_\infty) \cup \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{U}}_\infty}(\widehat{\mathrm{T}}_\infty) \cup \mathbf{IC}^i_{\mathrm{P},\widehat{\mathrm{U}}_\infty}(\widehat{\mathrm{T}}_\infty) \\
\widehat{\mathrm{U}}_\infty &= \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{T}}_\infty}(\widehat{\mathrm{U}}_\infty) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{T}}_\infty}(\widehat{\mathrm{U}}_\infty) \cup \mathbf{IC}^t(\widehat{\mathrm{U}}_\infty) \cup \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{T}}_\infty}(\widehat{\mathrm{U}}_\infty) \cup \mathbf{IC}^i_{\mathrm{P},\widehat{\mathrm{T}}_\infty}(\widehat{\mathrm{U}}_\infty)
\end{aligned}$$

Proof.

By Definitions 9.1.2, 7.2.7, 7.2.6, and 7.2.5.
□

Let α be a countable ordinal. Given a pair of extended atom sets $\widehat{\mathrm{T}}_\alpha$ and $\widehat{\mathrm{U}}_\alpha$, we know that $\widehat{\mathrm{T}}_\alpha \subseteq \widehat{\mathrm{U}}_\alpha$ and so $\pi(\widehat{\mathrm{T}}_\alpha) \subseteq \pi(\widehat{\mathrm{U}}_\alpha)$ by Lemma 9.1.1. We can construct an interpretation $\mathcal{I}_\alpha$ as follows: $\mathcal{I}_\alpha = \langle\, \pi(\widehat{\mathrm{T}}_\alpha); \pi(\widehat{\mathrm{U}}_\alpha) - \pi(\widehat{\mathrm{T}}_\alpha) \,\rangle$. Then the set of atoms $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o}$ ($\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o}$) in $\widehat{\mathrm{T}}_\alpha$ constitutes a *subset* of the set of strong value (code) inheritance candidates in $\mathcal{I}_\alpha$, whereas the set of atoms $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o}$ ($\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o}$) in $\widehat{\mathrm{U}}_\alpha$ constitutes a *superset* of the set of strong and weak value (code) inheritance candidates in $\mathcal{I}_\alpha$. In other words, $\widehat{\mathrm{T}}_\alpha$ *underestimates* inheritance information whereas $\widehat{\mathrm{U}}_\alpha$ *overestimates* inheritance information. The following lemma illustrates this book-keeping mechanism of the alternating fixpoint computation.

Lemma 9.1.4 Let $\mathcal{I}_\alpha = \langle\, \pi(\widehat{\mathrm{T}}_\alpha); \pi(\widehat{\mathrm{U}}_\alpha) - \pi(\widehat{\mathrm{T}}_\alpha) \,\rangle$ where α ranges over all countable ordinals:

(1) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\alpha$ then $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$

(2) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\alpha$ then $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$

(3) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ then $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_\alpha$

(4) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ then $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_\alpha$

Proof.

(1) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\alpha$ then $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$

Proof by transfinite induction.

The case of $\alpha = 0$ is trivial. Now suppose α is a successor ordinal. Since $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\alpha$, so $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{T}}_\alpha)$ by Lemma 9.1.3. Thus $\mathsf{o} \neq \mathsf{c}$, $\mathsf{o} : \mathsf{c} \in \widehat{\mathrm{T}}_\alpha$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{T}}_\alpha$, $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, by Definition 7.2.5. But $\mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1}) \supseteq \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_\alpha)$ by Lemma 9.1.1 and the monotonicity of $\mathbf{IB}_{\mathrm{P}}$. Thus $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_\alpha)$ and so $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ by Lemma 7.2.2.

If α is a limit ordinal and $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\alpha = \bigcup_{\beta<\alpha} \widehat{\mathrm{T}}_\beta$, then there exists $\gamma < \alpha$ such that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\gamma$. Therefore $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_\gamma} \mathsf{o}$ by the induction hypothesis. So $\mathsf{o} \neq \mathsf{c}$, $\mathsf{o} : \mathsf{c} \in \widehat{\mathrm{T}}_\gamma \subseteq \widehat{\mathrm{T}}_\alpha$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{T}}_\gamma \subseteq \widehat{\mathrm{T}}_\alpha$, $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_\gamma)$ by Lemma 7.2.2. But $\mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_\gamma) \supseteq \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_\alpha)$ by Lemma 9.1.1 and the monotonicity of $\mathbf{IB}_{\mathrm{P}}$. Thus $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_\alpha)$ and so $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ by Lemma 7.2.2.

(2) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\alpha$ then $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$

Similarly to (1).

(3) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ then $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_\alpha$

Since $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$, it follows that $\mathsf{o} \neq \mathsf{c}$, $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{U}}_\alpha$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{U}}_\alpha$, $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\alpha)$, by Lemma 7.2.2. It follows that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{T}}_\alpha}(\widehat{\mathrm{U}}_\alpha) \subseteq \widehat{\mathrm{U}}_\alpha$, by Definition 7.2.5 and Lemma 9.1.3.

(4) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: if $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{I}_\alpha} \mathsf{o}$ then $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_\alpha$

Similarly to (3).

□

Lemma 9.1.5 Let $\mathcal{M}$ be the optimistic object model of an F-logic program P. Then the following statements are true:

(1) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\infty$

(2) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\infty$

(3) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_\infty$

(4) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_\infty$

Proof.

Recall that $\mathcal{M} = \langle\, \pi(\widehat{\mathrm{T}}_\infty); \pi(\widehat{\mathrm{U}}_\infty) - \pi(\widehat{\mathrm{T}}_\infty) \,\rangle$ by Lemma 9.1.2.

(1) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\infty$

By Lemma 9.1.3 and Definition 7.2.5, $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\infty$, iff $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{U}}_\infty}(\widehat{\mathrm{T}}_\infty)$, iff $\mathsf{o} \neq \mathsf{c}$, $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{T}}_\infty$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{T}}_\infty$, and $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_\infty)$, thus iff $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$, by Lemma 7.2.2.

(2) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\infty$

Similarly to (1).

(3) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_\infty$

By Lemma 9.1.3 and Definition 7.2.5, $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_\infty$, iff $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{T}}_\infty}(\widehat{\mathrm{U}}_\infty)$, iff $\mathsf{o} \neq \mathsf{c}$, $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{U}}_\infty$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{U}}_\infty$, and $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, thus iff $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \overset{wv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{s}$, by Lemma 7.2.2.

(4) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{U}}_\infty$

Similarly to (3).

□

The lemma above says that $\widehat{\mathrm{T}}_\infty$ includes exactly all strong inheritance candidates while $\widehat{\mathrm{U}}_\infty$ includes exactly all strong and weak inheritance candidates in the object

model. This essentially implies that the optimistic object model is indeed a stable object model.

Proposition 9.1.6 The optimistic object model $\mathcal{M}$ of an F-logic program P is a stable object model of P.

Proof.

Let $\mathcal{M} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be the optimistic object model of P. Then $\mathrm{T} = \pi(\widehat{\mathrm{T}}_\infty)$ and $\mathrm{U} = \pi(\widehat{\mathrm{U}}_\infty) - \pi(\widehat{\mathrm{T}}_\infty)$. So by Definition 8.1.1 and Lemma 9.1.5, $\widehat{\mathrm{T}}_\mathcal{M} = \widehat{\mathrm{T}}_\infty$ and $\widehat{\mathrm{U}}_\mathcal{M} = \widehat{\mathrm{U}}_\infty$. Moreover, $\widehat{\mathrm{U}}_\infty = \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\infty)$ and $\widehat{\mathrm{T}}_\infty = \Psi_\mathrm{P}(\widehat{\mathrm{U}}_\infty)$ by Definition 9.1.2 and Lemma 9.1.1. It follows that $\widehat{\mathrm{T}}_\mathcal{M} = \Psi_\mathrm{P}(\widehat{\mathrm{U}}_\mathcal{M})$ and $\widehat{\mathrm{U}}_\mathcal{M} = \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\mathcal{M})$. Therefore, $\mathcal{M}$ is a stable interpretation and thus a stable object model of P.
□

Corollary 9.1.7 The optimistic object model $\mathcal{M}$ of an F-logic program P is an object model of P. Moreover, $\mathcal{M}$ satisfies the optimistic ISA transitivity constraint and the optimistic inheritance constraint.

Proof.

By Proposition 9.1.6 and Theorem 8.2.9.
□

9.2 Information Ordering

By comparing the amount of "definite" information, *i.e.*, truth *and* falsehood, that is contained in different stable object models of an F-logic program P, we can define a partial order, called *information ordering*, among stable object models.

Definition 9.2.1 (Information Ordering) Given two stable object models, $\mathcal{I}_1 = \langle\, \mathrm{P}_1; \mathrm{Q}_1 \,\rangle$ and $\mathcal{I}_2 = \langle\, \mathrm{P}_2; \mathrm{Q}_2 \,\rangle$, of an F-logic program P, let $\mathrm{R}_1 = \mathcal{HB}_\mathrm{P} - (\mathrm{P}_1 \cup \mathrm{Q}_1)$ and $\mathrm{R}_2 = \mathcal{HB}_\mathrm{P} - (\mathrm{P}_2 \cup \mathrm{Q}_2)$. Then $\mathcal{I}_1 \preceq \mathcal{I}_2$ iff $\mathrm{P}_1 \subseteq \mathrm{P}_2$ and $\mathrm{R}_1 \subseteq \mathrm{R}_2$.

Intuitively, a stable object model is "smaller" in the information ordering, if it carries less amount of truth and less amount of falsehood. Therefore, the least stable object model contains the smallest set of true atoms and the smallest set of false atoms among all stable object models.

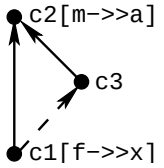
Definition 9.2.2 (Least Stable Object Model) Let $\mathcal{I}$ be a stable object model of an F-logic program P. $\mathcal{I}$ is the *least stable object model* of P, if $\mathcal{I} \preceq \mathcal{J}$ for any stable object model $\mathcal{J}$ of P.

Theorem 9.2.1 The optimistic object model $\mathcal{M}$ of an F-logic program P is the least stable object model of P.

Proof.

Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be any stable object model of P. We need to show that $\mathcal{M} \preceq \mathcal{I}$. Clearly, given two stable object models, $\mathcal{I}_1 = \langle\, \mathrm{P}_1; \mathrm{Q}_1 \,\rangle$ and $\mathcal{I}_2 = \langle\, \mathrm{P}_2; \mathrm{Q}_2 \,\rangle$, of an F-logic program P, $\mathcal{I}_1 \preceq \mathcal{I}_2$ iff $\mathrm{P}_1 \subseteq \mathrm{P}_2$ and $\mathrm{P}_1 \cup \mathrm{Q}_1 \supseteq \mathrm{P}_2 \cup \mathrm{Q}_2$. Recall that $\mathcal{M} = \langle\, \pi(\widehat{\mathrm{T}}_\infty); \pi(\widehat{\mathrm{U}}_\infty) - \pi(\widehat{\mathrm{T}}_\infty) \,\rangle$. Therefore, to show that $\mathcal{M} \preceq \mathcal{I}$, it suffices to show that $\pi(\widehat{\mathrm{T}}_\infty) \subseteq \mathrm{T}$ and $\pi(\widehat{\mathrm{U}}_\infty) \supseteq \mathrm{T} \cup \mathrm{U}$.

Since $\mathcal{I}$ is a stable object model of P, it follows that $\widehat{\mathrm{T}}_\mathcal{I} = \Psi_\mathrm{P}(\widehat{\mathrm{U}}_\mathcal{I})$ and $\widehat{\mathrm{U}}_\mathcal{I} = \Psi_\mathrm{P}(\widehat{\mathrm{T}}_\mathcal{I})$. Therefore, $\widehat{\mathrm{T}}_\mathcal{I} = \Psi_\mathrm{P}(\widehat{\mathrm{U}}_\mathcal{I}) = \Psi_\mathrm{P}(\Psi_\mathrm{P}(\widehat{\mathrm{T}}_\mathcal{I})) = \mathbf{F}_\mathrm{P}(\widehat{\mathrm{T}}_\mathcal{I})$ and so $\widehat{\mathrm{T}}_\mathcal{I}$ is a fixpoint of $\mathbf{F}_\mathrm{P}$. Similarly, $\widehat{\mathrm{U}}_\mathcal{I}$ is also a fixpoint of $\mathbf{F}_\mathrm{P}$. But $\widehat{\mathrm{T}}_\infty = \mathrm{lfp}(\mathbf{F}_\mathrm{P})$ and $\widehat{\mathrm{U}}_\infty = \mathrm{gfp}(\mathbf{F}_\mathrm{P})$, by Lemma 9.1.1. It follows that $\widehat{\mathrm{T}}_\infty \subseteq \widehat{\mathrm{T}}_\mathcal{I}$ and $\widehat{\mathrm{U}}_\infty \supseteq \widehat{\mathrm{U}}_\mathcal{I}$. Thus $\pi(\widehat{\mathrm{T}}_\infty) \subseteq \pi(\widehat{\mathrm{T}}_\mathcal{I})$ and $\pi(\widehat{\mathrm{U}}_\infty) \supseteq \pi(\widehat{\mathrm{U}}_\mathcal{I})$, by Lemma 7.1.1. Moreover, $\pi(\widehat{\mathrm{T}}_\mathcal{I}) = \mathrm{T}$ and $\pi(\widehat{\mathrm{U}}_\mathcal{I}) = \mathrm{T} \cup \mathrm{U}$, by Definition 8.1.1. So $\pi(\widehat{\mathrm{T}}_\infty) \subseteq \mathrm{T}$ and $\pi(\widehat{\mathrm{U}}_\infty) \supseteq \mathrm{T} \cup \mathrm{U}$.
□



code

$\mathsf{c}_1 : \mathsf{c}_2.$
$\mathsf{c}_3 :: \mathsf{c}_2.$
$\mathsf{c}_1 : \mathsf{c}_3 \longleftarrow \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}].$
$\mathsf{c}_1[\mathsf{f} \twoheadrightarrow \mathsf{x}].$
$\mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}].$
$\mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{b}] \longleftarrow \mathsf{c}_3[\mathsf{f} \twoheadrightarrow \mathsf{x}].$

Figure 10: Computation of Optimistic Object Models

Example 9.2.1 We illustrate the computation of optimistic object models using the F-logic program P in Figure 10. First let T and U denote the following sets of atoms:

$$\begin{aligned} \mathrm{T} &= \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_3 :: \mathsf{c}_2, \mathsf{c}_1[\mathsf{f} \twoheadrightarrow \mathsf{x}]^{\mathsf{c}_1}_{\mathrm{local}}, \mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{local}}\} \\ \mathrm{U} &= \{\mathsf{c}_1 : \mathsf{c}_3, \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{value}}, \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c}_3}_{\mathrm{code}}\} \end{aligned}$$

Then the computation process of Ψ_P is as follows:

$$\begin{aligned} \widehat{\mathrm{T}}_0 &= \emptyset \\ \widehat{\mathrm{T}}_1 &= \Psi_\mathrm{P}(\widehat{\mathrm{T}}_0) = \mathrm{T} \cup \mathrm{U} \cup \{\mathsf{c}_2[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{c}_1, \mathsf{c}_3[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{c}_1\} \\ \widehat{\mathrm{T}}_2 &= \Psi_\mathrm{P}(\widehat{\mathrm{T}}_1) = \mathrm{T} \\ \widehat{\mathrm{T}}_3 &= \Psi_\mathrm{P}(\widehat{\mathrm{T}}_2) = \widehat{\mathrm{T}}_1 \\ \widehat{\mathrm{T}}_4 &= \Psi_\mathrm{P}(\widehat{\mathrm{T}}_3) = \widehat{\mathrm{T}}_2 \end{aligned}$$

Therefore, $\mathrm{lfp}(\mathbf{F}_\mathrm{P}) = \widehat{\mathrm{T}}_2$ and $\Psi_\mathrm{P}(\mathrm{lfp}(\mathbf{F}_\mathrm{P})) = \widehat{\mathrm{T}}_1$, and so the optimistic object model of the program in Figure 10 is $\langle\, \mathrm{T}; \mathrm{U} \,\rangle$. □

Chapter 10

Minimal Object Models

So far we have shown two different characterizations of the optimistic object model semantics: (i) the optimistic object model is the least fixpoint of an extended alternating fixpoint computation; and (ii) it is the least three-valued stable object model with respect to information ordering. In this chapter we will introduce a different partial order, called *truth ordering*, for all object models of an F-logic program. We will then present a new characterization of optimistic object models: optimistic object models are *minimal* object models that satisfy the optimistic ISA transitivity constraint and the optimistic inheritance constraint.

10.1 Truth Ordering

Since the introduction of the Closed World Assumption [47], comparing different models of a program based on the amount of "truth" contained in those models has become a common technique. Typically, the true component of a model is minimized and the false component is maximized. However, in F-logic we also deal with inheritance, which complicates the matters somewhat, because the truth value of a fact may depend on inheritance. This can create object models that look similar but actually are incomparable. This issue is illustrated by an example that follows the definition of minimality below. The solution is to minimize not only the set of true atoms of an object model, but also the amount of positive inheritance information implied by the object model.

Definition 10.1.1 (Truth Ordering) Let $\mathcal{I}_1 = \langle \mathrm{P}_1; \mathrm{Q}_1 \rangle$ and $\mathcal{I}_2 = \langle \mathrm{P}_2; \mathrm{Q}_2 \rangle$ be two object models of an F-logic program P. We write $\mathcal{I}_1 \leq \mathcal{I}_2$ *iff*

(i) $\mathrm{P}_1 \subseteq \mathrm{P}_2$; and

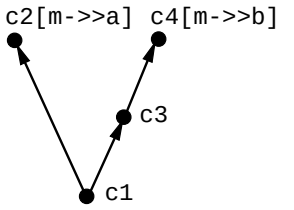
(ii) $\mathrm{P}_1 \cup \mathrm{Q}_1 \subseteq \mathrm{P}_2 \cup \mathrm{Q}_2$; and

(iii) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_1} \mathsf{o}$ implies $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}_2} \mathsf{o}$; and

(iv) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}_1} \mathsf{o}$ implies $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}_2} \mathsf{o}$.

Definition 10.1.2 (Minimal Object Model) An object model $\mathcal{I}$ is *minimal* iff there exists no object model $\mathcal{J}$ such that $\mathcal{J} \leq \mathcal{I}$ and $\mathcal{J} \neq \mathcal{I}$.

The above definitions minimize the number of strong inheritance candidates implied by an object model *in addition to* the usual minimization of truth and maximization of falsehood. This is needed because increasing the number of false facts might inflate the number of strong inheritance candidates, which in turn might unjustifiably inflate the number of facts that are derived by inheritance.



$\mathsf{c}_1 : \mathsf{c}_2.$
$\mathsf{c}_1 : \mathsf{c}_3.$
$\mathsf{c}_3 :: \mathsf{c}_4.$
$\mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}].$
$\mathsf{c}_4[\mathsf{m} \twoheadrightarrow \mathsf{b}].$
$\mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{c}] \longleftarrow \mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}].$

Figure 11: Minimal Object Model

Example 10.1.1 Consider the program in Figure 11 and the following two object models of the program: $\mathcal{I}_1 = \langle \mathrm{P}_1; \mathrm{Q}_1 \rangle$, where

$$\begin{aligned} \mathrm{P}_1 &= \{\mathsf{c}_1 : \mathsf{c}_2, \mathsf{c}_1 : \mathsf{c}_3, \mathsf{c}_3 :: \mathsf{c}_4, \mathsf{c}_2[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{local}}, \mathsf{c}_4[\mathsf{m} \twoheadrightarrow \mathsf{b}]^{\mathsf{c}_4}_{\mathrm{local}}\} \\ \mathrm{Q}_1 &= \emptyset \end{aligned}$$

and $\mathcal{I}_2 = \langle \mathrm{P}_2; \mathrm{Q}_2 \rangle$, where

$$\begin{aligned} \mathrm{P}_2 &= \mathrm{P}_1 \\ \mathrm{Q}_2 &= \{\mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{value}}, \mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{c}]^{\mathsf{c}_3}_{\mathrm{local}}\} \end{aligned}$$

$\mathcal{I}_1$ and $\mathcal{I}_2$ both agree on the atoms that are true. But in $\mathcal{I}_1$ both $\mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}]^{\mathsf{c}_2}_{\mathrm{value}}$ and $\mathsf{c}_3[\mathsf{m} \twoheadrightarrow \mathsf{c}]^{\mathsf{c}_3}_{\mathrm{local}}$ are false, whereas in $\mathcal{I}_2$ they are both undefined. Clearly, $\mathcal{I}_1$ carries more false atoms than $\mathcal{I}_2$ and so with the usual notion of minimality we would say $\mathcal{I}_1 \leq \mathcal{I}_2$. However, $\mathcal{I}_1$ is not as "tight" as it appears, because the additional false atoms in $\mathcal{I}_1$ are not automatically implied by the program under our optimistic object model semantics. Indeed, although $\mathsf{c}_4[\mathsf{m}]$ is a strong value inheritance candidate for c_1 in $\mathcal{I}_1$, it is only a weak value inheritance candidate in $\mathcal{I}_2$. We can see that it is due to this extra positive information about inheritance candidates that $\mathcal{I}_1$ is able to increase the number of false atoms while keeping the true atoms intact. This anomaly is eliminated by the inheritance minimization built into Definition 10.1.1, which renders the two models incomparable, *i.e.*, $\mathcal{I}_1 \not\leq \mathcal{I}_2$. □

10.2 Minimality

Now we will present the main theorem of this chapter. In the proof of this theorem we will often need to compare a normal atom set with the projection of an extended atom set to test for set inclusion. Without complicating the presentation we will usually omit the project function and just write the extended atom set, when its intended usage is clear from the context.

Theorem 10.2.1 The optimistic object model $\mathcal{M}$ of an F-logic program P is minimal among those object models of P that satisfy the optimistic ISA transitivity constraint and the optimistic inheritance constraint.

Proof. By contradiction.

Recall that $\mathcal{M} = \langle\, \pi(\widehat{\mathrm{T}}_\infty); \pi(\widehat{\mathrm{U}}_\infty) - \pi(\widehat{\mathrm{T}}_\infty) \,\rangle$. Let $\mathcal{I} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be any object model of P that satisfies the optimistic ISA transitivity constraint and the optimistic inheritance constraint. We want to show that if $\mathcal{I} \leq \mathcal{M}$ then $\mathrm{T} = \widehat{\mathrm{T}}_\infty$ and $\mathrm{T} \cup \mathrm{U} = \widehat{\mathrm{U}}_\infty$.

Let us assume that $\mathcal{I} \leq \mathcal{M}$. So by Definition 10.1.1 it follows that: (i) $\mathrm{T} \subseteq \widehat{\mathrm{T}}_\infty$; (ii) $\mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_\infty$; (iii) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ implies $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$; and (iv) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ implies $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$.

Let $\mathcal{J} = \langle\, \mathrm{T}; \emptyset \,\rangle$ and $\mathcal{K} = \langle\, \mathrm{T} \cup \mathrm{U}; \emptyset \,\rangle$.

Suppose on the contrary $\mathrm{T} \subset \widehat{\mathrm{T}}_\infty$. Since $\widehat{\mathrm{T}}_\infty = \bigcup_\gamma \widehat{\mathrm{T}}_\gamma$ by Definition 9.1.2 and $\{\widehat{\mathrm{T}}_\gamma\}$ is an increasing sequence by Lemma 9.1.1, let α be the first ordinal such that $\mathrm{T} \subset \widehat{\mathrm{T}}_\alpha$ and $\mathrm{T} \supseteq \widehat{\mathrm{T}}_\gamma$ for all $\gamma < \alpha$. Clearly, α must be a successor ordinal. Thus $\widehat{\mathrm{T}}_\alpha = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}})$, by Definitions 9.1.2 and 7.2.7. Since $\mathbf{T}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}$ is monotonic by Lemma 7.2.3, it follows that the ordinal powers of $\mathbf{T}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}$ is an increasing sequence by Proposition 4.1.3. Denote $\widehat{\mathrm{J}}_\gamma = \mathbf{T}^\gamma_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}$ for all ordinal γ. Let β be the first ordinal such that $\mathrm{T} \subset \widehat{\mathrm{J}}_\beta$ and $\mathrm{T} \supseteq \widehat{\mathrm{J}}_\gamma$ for all $\gamma < \beta$. Clearly, β must be a successor ordinal.

Let A be any atom in $\mathcal{HB}_\mathrm{P}$ such that $\mathsf{A} \notin \mathrm{T}$ and $\mathsf{A} \in \widehat{\mathrm{J}}_\beta$. By Definitions 7.2.6 and 7.2.5,

$$\begin{aligned}\widehat{\mathrm{J}}_\beta \;=\; & \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{J}}_{\beta-1}) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{J}}_{\beta-1}) \cup \\ & \mathbf{IC}^t(\widehat{\mathrm{J}}_{\beta-1}) \cup \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{J}}_{\beta-1}) \cup \mathbf{IC}^i_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{J}}_{\beta-1})\end{aligned}$$

There are four cases to consider:

(1) $\mathsf{A} \in \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{J}}_{\beta-1})$

By Definition 7.2.2, there must exist a V-rule, $\mathsf{H} \longleftarrow \mathsf{L}_1, \ldots, \mathsf{L}_\mathsf{n}$, in $\mathit{ground}(\mathrm{P})$, such that H matches A, and for all $\mathsf{L}_\mathsf{i}, 1 \leq \mathsf{i} \leq \mathsf{n}$: (i) if L_i is a positive literal, then $\mathit{val}^{\mathbf{b}}_{\widehat{\mathrm{J}}_{\beta-1}}(\mathsf{L}_\mathsf{i}) = \mathbf{t}$; and (ii) if L_i is a negative literal, then $\mathit{val}^{\mathbf{b}}_{\widehat{\mathrm{U}}_{\alpha-1}}(\mathsf{L}_\mathsf{i}) = \mathbf{t}$.

We will show that for all $\mathsf{L_i}, 1 \leq \mathsf{i} \leq \mathsf{n}$, $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L_i}) = \mathbf{t}$. If $\mathsf{L_i}$ is a positive literal, since $\widehat{\mathrm{J}}_{\beta-1} \subseteq \mathrm{T}$ and $val^{\mathbf{b}}_{\widehat{\mathrm{J}}_{\beta-1}}(\mathsf{L_i}) = \mathbf{t}$, then it follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L_i}) = \mathbf{t}$, by Lemma 5.3.2. Thus $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L_i}) = \mathbf{t}$ by Lemma 5.3.1. Note that $\widehat{\mathrm{U}}_{\infty} \subseteq \widehat{\mathrm{U}}_{\alpha-1}$ by Lemma 9.1.1. It follows that $\mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\infty} \subseteq \widehat{\mathrm{U}}_{\alpha-1}$. Therefore, if $\mathsf{L_i}$ is a negative literal, since $val^{\mathbf{b}}_{\widehat{\mathrm{U}}_{\alpha-1}}(\mathsf{L_i}) = \mathbf{t}$, then it follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{L_i}) = \mathbf{t}$, by Lemma 5.3.2. Thus $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L_i}) = \mathbf{t}$ by Lemma 5.3.1.

Because $\mathcal{I}$ satisfies P, it follows that $\mathcal{I}(\mathsf{A}) = \mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) = \mathbf{t}$. Thus $\mathsf{A} \in \mathrm{T}$, a contradiction.

(2) $\mathsf{A} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{J}}_{\beta-1})$

Then $\mathsf{A} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}$. Thus by Definition 7.2.4, $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}_{\beta-1}$, $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, and there is a C-rule, $\mathsf{R} \equiv \mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{B}$, in $ground(\mathrm{P})$ such that for every literal $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$: (i) if L is a positive literal then $val^{\mathbf{b}}_{\widehat{\mathrm{J}}_{\beta-1}}(\mathsf{L}) = \mathbf{t}$; and (ii) if L is a negative literal then $val^{\mathbf{b}}_{\widehat{\mathrm{U}}_{\alpha-1}}(\mathsf{L}) = \mathbf{t}$.

Because $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}_{\beta-1}$, there must exist a successor ordinal $\rho \leq \beta - 1 < \beta$, such that $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}_{\rho}$. It follows that $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{J}}_{\rho-1})$. Therefore, $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{J}}_{\rho-1}$, and $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, by Definition 7.2.5. Since $\widehat{\mathrm{J}}_{\rho-1} \subseteq \mathrm{T}$ and $\mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\infty} \subseteq \widehat{\mathrm{U}}_{\alpha-1}$, it follows that $\mathsf{o}:\mathsf{c} \in \mathrm{T}$ and $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T} \cup \mathrm{U})$. Thus $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ by Lemma 7.2.2. Because $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, so $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T} \cup \mathrm{U})$ by the monotonicity of $\mathbf{IB}_{\mathrm{P}}$. It follows that $\mathsf{o}[\mathsf{m}]$ is neither a strong nor a weak local context in $\mathcal{I}$, by Definitions 7.2.3 and 6.1.1.

Next we will show that there is no x such that $\mathsf{x} \neq \mathsf{c}$ and $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$. Suppose on the contrary there is $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$. Then $\mathsf{x} \neq \mathsf{o}$, $\mathsf{o}:\mathsf{x} \in \mathrm{T} \cup \mathrm{U}$, $\mathsf{x}[\mathsf{m} \twoheadrightarrow \mathsf{y}]^{\mathsf{x}}_{\mathrm{local}} \in \mathrm{T} \cup \mathrm{U}$ for some value y or there is a C-rule in $ground(\mathrm{P})$ which specifies the instance method m for the class c, and $ov(\mathsf{x},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$, by Lemma 7.2.2. Since $\mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\infty} \subseteq \widehat{\mathrm{U}}_{\alpha-1}$ and $\mathrm{T} \supseteq \widehat{\mathrm{T}}_{\alpha-1}$, it follows that $\mathsf{o}:\mathsf{x} \in \widehat{\mathrm{U}}_{\alpha-1}$, $\mathsf{x}[\mathsf{m} \twoheadrightarrow \mathsf{y}]^{\mathsf{x}}_{\mathrm{local}} \in \widehat{\mathrm{U}}_{\alpha-1}$ for some value y or there is a C-rule in $ground(\mathrm{P})$ which specifies the instance method m for the class c, and $ov(\mathsf{x},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\alpha-1})$. Thus $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{T}}_{\alpha-1}}(\widehat{\mathrm{U}}_{\alpha-1}) \subseteq \widehat{\mathrm{U}}_{\alpha-1}$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{T}}_{\alpha-1}}(\widehat{\mathrm{U}}_{\alpha-1}) \subseteq \widehat{\mathrm{U}}_{\alpha-1}$, by Definition 7.2.5 and Lemma 9.1.3. Therefore, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, by Definition 7.2.3, which contradicts the fact that $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$.

So far we have shown that $\mathsf{o}[\mathsf{m}]$ is neither a strong nor a weak local context in $\mathcal{I}$, $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, and there is no x such that $\mathsf{x} \neq \mathsf{c}$ and $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$. Therefore, o strongly inherits R in $\mathcal{I}$, by Definition 6.3.2. So $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{t}$.

We already know that for every literal $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$: (i) if L is positive then $val^{\mathbf{b}}_{\widehat{\mathrm{J}}_{\beta-1}}(\mathsf{L}) = \mathbf{t}$; and (ii) if L is negative then $val^{\mathbf{b}}_{\widehat{\mathrm{U}}_{\alpha-1}}(\mathsf{L}) = \mathbf{t}$. Now we will show that for all $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$, $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$. If L is a positive literal, since $\widehat{\mathrm{J}}_{\beta-1} \subseteq \mathrm{T}$ and $val^{\mathbf{b}}_{\widehat{\mathrm{J}}_{\beta-1}}(\mathsf{L}) = \mathbf{t}$, then it follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}) = \mathbf{t}$, by Lemma 5.3.2. Thus $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$ by Lemma 5.3.1. Note that $\widehat{\mathrm{U}}_{\infty} \subseteq \widehat{\mathrm{U}}_{\alpha-1}$ by Lemma 9.1.1. It follows that $\mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\infty} \subseteq \widehat{\mathrm{U}}_{\alpha-1}$. Therefore, if L is a negative literal, since $val^{\mathbf{b}}_{\widehat{\mathrm{U}}_{\alpha-1}}(\mathsf{L}) = \mathbf{t}$, then it follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{L}) = \mathbf{t}$, by Lemma 5.3.2. Thus $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$ by Lemma 5.3.1.

Therefore, $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) = \mathbf{t}$ for every literal $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$. It follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{t}$. Moreover, $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) = \mathbf{t}$. Because $\mathcal{I}$ is an object model of P, so $\mathcal{I}$ should satisfy $\mathsf{R}||\mathsf{o}$. It follows that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) = \mathbf{t}$ by Definition 6.3.4. Thus $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathrm{T}$, a contradiction.

(3) $\mathsf{A} \in \mathbf{IC}^{t}(\widehat{\mathrm{J}}_{\beta-1})$

If $\mathsf{A} = \mathsf{o}:\mathsf{c}$, then there exists x, such that $\mathsf{o}:\mathsf{x} \in \widehat{\mathrm{J}}_{\beta-1}$ and $\mathsf{x}::\mathsf{c} \in \widehat{\mathrm{J}}_{\beta-1}$, by Definition 7.2.5. Since $\widehat{\mathrm{J}}_{\beta-1} \subseteq \mathrm{T}$, it follows that $\mathsf{o}:\mathsf{x} \in \mathrm{T}$ and $\mathsf{x}::\mathsf{c} \in \mathrm{T}$. So $\mathcal{I}(\mathsf{o}:\mathsf{x}) = \mathbf{t}$ and $\mathcal{I}(\mathsf{x}::\mathsf{c}) = \mathbf{t}$. Because $\mathcal{I}$ is an object model of P and so satisfies the positive ISA transitivity constraint, therefore $\mathcal{I}(\mathsf{o}:\mathsf{c}) = \mathbf{t}$ by Definition 6.2.1. It follows that $\mathsf{o}:\mathsf{c} \in \mathrm{T}$, a contradiction. Similarly, if $\mathsf{A} = \mathsf{s}::\mathsf{c}$, then we can also show that $\mathsf{s}::\mathsf{c} \in \mathrm{T}$, which is a contradiction.

(4) $\mathsf{A} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{J}}_{\beta-1})$

Then $\mathsf{A} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}$. Thus $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}_{\beta-1}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{J}}_{\beta-1}$, $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, and $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, by Definition 7.2.5. Because $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}_{\beta-1}$, there must exist a successor ordinal $\rho \leq \beta - 1 < \beta$, such that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}_{\rho}$. It follows that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}}(\widehat{\mathrm{J}}_{\rho-1})$. Therefore, $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{J}}_{\rho-1}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{J}}_{\rho-1}$ for some value z, and $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, by Definition 7.2.5. Since $\widehat{\mathrm{J}}_{\rho-1} \subseteq \widehat{\mathrm{J}}_{\beta-1} \subseteq \mathrm{T}$ and $\mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\infty} \subseteq \widehat{\mathrm{U}}_{\alpha-1}$, it follows that $\mathsf{o}:\mathsf{c} \in \mathrm{T}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \mathrm{T}$, and $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T} \cup \mathrm{U})$. Thus $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ by Lemma 7.2.2. Because $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, so $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T} \cup \mathrm{U})$ by the monotonicity of $\mathbf{IB}_{\mathrm{P}}$. It follows that $\mathsf{o}[\mathsf{m}]$ is neither a strong nor a weak local context in $\mathcal{I}$, by Definitions 7.2.3 and 6.1.1.

Next we will show that there is no x such that $\mathsf{x} \neq \mathsf{c}$ and $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$. Suppose on the contrary there is $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$. Then $\mathsf{x} \neq \mathsf{o}$, $\mathsf{o}:\mathsf{x} \in \mathrm{T} \cup \mathrm{U}$, $\mathsf{x}[\mathsf{m} \twoheadrightarrow \mathsf{y}]^{\mathsf{x}}_{\mathrm{local}} \in \mathrm{T} \cup \mathrm{U}$ for some value y or there is a C-rule in $ground(\mathrm{P})$ which specifies the instance method m for the class c, and $ov(\mathsf{x},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$, by Lemma 7.2.2. Since $\mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\infty} \subseteq \widehat{\mathrm{U}}_{\alpha-1}$ and $\mathrm{T} \supseteq \widehat{\mathrm{T}}_{\alpha-1}$, it follows that $\mathsf{o}:\mathsf{x} \in \widehat{\mathrm{U}}_{\alpha-1}$, $\mathsf{x}[\mathsf{m} \twoheadrightarrow \mathsf{y}]^{\mathsf{x}}_{\mathrm{local}} \in \widehat{\mathrm{U}}_{\alpha-1}$ for some value y or there is

a C-rule in *ground*(P) which specifies the instance method m for the class c, and $ov(\mathsf{x},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_{\alpha-1})$. Thus $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{T}}_{\alpha-1}}(\widehat{\mathrm{U}}_{\alpha-1}) \subseteq \widehat{\mathrm{U}}_{\alpha-1}$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{T}}_{\alpha-1}}(\widehat{\mathrm{U}}_{\alpha-1}) \subseteq \widehat{\mathrm{U}}_{\alpha-1}$, by Definition 7.2.5 and Lemma 9.1.3. Therefore, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$, by Definition 7.2.3, which contradicts the fact that $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1})$.

So far we have shown that $\mathsf{o}[\mathsf{m}]$ is neither a strong nor a weak local context in $\mathcal{I}$, $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}}) = \mathbf{t}$, and there is no x such that $\mathsf{x} \neq \mathsf{c}$ and $\mathsf{x}[\mathsf{m}] \rightsquigarrow_{\mathcal{I}} \mathsf{o}$. Because $\mathcal{I}$ is an object model of P and so satisfies the unique source inheritance constraint, therefore $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \mathrm{T}$ by Definition 6.2.3, a contradiction.

Therefore, if $\mathrm{T} \subset \widehat{\mathrm{T}}_{\infty}$, then we can derive a contradiction in all three possible cases. So $\mathrm{T} = \widehat{\mathrm{T}}_{\infty}$.

It remains to show that $\mathrm{T} \cup \mathrm{U} = \widehat{\mathrm{U}}_{\infty}$. We know that $\mathrm{T} \cup \mathrm{U} \subseteq \widehat{\mathrm{U}}_{\infty}$, because $\mathcal{I} \leq \mathcal{M}$. Therefore if we can show that $\mathrm{T} \cup \mathrm{U} \supseteq \widehat{\mathrm{U}}_{\infty}$, then $\mathrm{T} \cup \mathrm{U} = \widehat{\mathrm{U}}_{\infty}$. By Definitions 9.1.2 and 7.2.7, $\widehat{\mathrm{U}}_{\infty} = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}})$. Since $\mathbf{T}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}}$ is monotonic, the ordinal powers of $\mathbf{T}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}}$ is an increasing sequence by Proposition 4.1.3. Denote $\widehat{\mathrm{K}}_{\gamma} = \mathbf{T}^{\gamma}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}}$ for all ordinal γ. We will prove by transfinite induction that $\mathrm{T} \cup \mathrm{U} \supseteq \widehat{\mathrm{K}}_{\alpha}$ for all ordinal α, thus complete the proof.

The case for a limit ordinal α is trivial. If $\alpha = 0$, then $\widehat{\mathrm{K}}_0 = \emptyset \subseteq \mathrm{T} \cup \mathrm{U}$. If $\alpha \neq 0$, then $\widehat{\mathrm{K}}_{\alpha} = \bigcup_{\beta<\alpha} \widehat{\mathrm{K}}_{\beta}$. By the induction hypothesis we know that $\mathrm{T} \cup \mathrm{U} \supseteq \widehat{\mathrm{K}}_{\beta}$ for all $\beta < \alpha$. So $\mathrm{T} \cup \mathrm{U} \supseteq \widehat{\mathrm{K}}_{\alpha}$.

Let α be a successor ordinal and A be any atom in $\mathcal{HB}_{\mathrm{P}}$ such that $\mathsf{A} \in \widehat{\mathrm{K}}_{\alpha}$. We will show that $\mathsf{A} \in \mathrm{T} \cup \mathrm{U}$. By Definitions 7.2.6 and 7.2.5,

$$\begin{aligned}\widehat{\mathrm{K}}_{\alpha} \;=\; & \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}}(\widehat{\mathrm{K}}_{\alpha-1}) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}}(\widehat{\mathrm{K}}_{\alpha-1}) \cup \\ & \mathbf{IC}^{t}(\widehat{\mathrm{K}}_{\alpha-1}) \cup \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}}(\widehat{\mathrm{K}}_{\alpha-1}) \cup \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}}(\widehat{\mathrm{K}}_{\alpha-1})\end{aligned}$$

There are four cases to consider:

(1) $\mathsf{A} \in \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}}(\widehat{\mathrm{K}}_{\alpha-1})$

By Definition 7.2.2, there must exist a V-rule, $\mathsf{H} \longleftarrow \mathsf{L}_1,\ldots,\mathsf{L}_{\mathsf{n}}$, in *ground*(P), such that H matches A, and for all $\mathsf{L}_{\mathsf{i}}, 1 \leq \mathsf{i} \leq \mathsf{n}$: (i) if L_{i} is a positive literal, then $val^{\mathbf{b}}_{\widehat{\mathrm{K}}_{\alpha-1}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$; and (ii) if L_{i} is a negative literal, then $val^{\mathbf{b}}_{\widehat{\mathrm{T}}_{\infty}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$.

We will show that for all $\mathsf{L}_{\mathsf{i}}, 1 \leq \mathsf{i} \leq \mathsf{n}, \mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_{\mathsf{i}}) \geq \mathbf{u}$. If L_{i} is a positive literal, since $\widehat{\mathrm{K}}_{\alpha-1} \subseteq \mathrm{T} \cup \mathrm{U}$ by the induction hypothesis and $val^{\mathbf{b}}_{\widehat{\mathrm{K}}_{\alpha-1}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$, then it follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$, by Lemma 5.3.2. Thus $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_{\mathsf{i}}) \geq \mathbf{u}$ by Lemma 5.3.1. We have proved that $\mathrm{T} = \widehat{\mathrm{T}}_{\infty}$. Therefore, if L_{i} is a negative literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}_{\mathsf{i}}) = \mathcal{V}^{\mathbf{b}}_{\widehat{\mathrm{T}}_{\infty}}(\mathsf{L}_{\mathsf{i}}) = \mathbf{t}$. Thus $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_{\mathsf{i}}) \geq \mathbf{u}$ by Lemma 5.3.1.

Because $\mathcal{I}$ satisfies P, it follows that $\mathcal{I}(\mathsf{A}) = \mathcal{V}^{\mathbf{h}}_{\mathcal{I}}(\mathsf{H}) \geq \mathbf{u}$. Thus $\mathsf{A} \in \mathrm{T} \cup \mathrm{U}$.

(2) $\mathsf{A} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{T}}_\infty}(\widehat{\mathrm{K}}_{\alpha-1})$

Then $\mathsf{A} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}$. Thus by Definition 7.2.4, $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}_{\alpha-1}$, $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, and there is a C-rule, $\mathsf{R} \equiv \mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{B}$, in $ground(\mathrm{P})$ such that for every literal $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$: (i) if L is a positive literal then $val^{\mathbf{b}}_{\widehat{\mathrm{K}}_{\alpha-1}}(\mathsf{L}) = \mathbf{t}$; and (ii) if L is a negative literal then $val^{\mathbf{b}}_{\widehat{\mathrm{T}}_\infty}(\mathsf{L}) = \mathbf{t}$.

Because $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}_{\alpha-1}$, there must exist a successor ordinal $\rho \leq \alpha - 1 < \alpha$, such that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}_{\rho}$. It follows that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{T}}_\infty}(\widehat{\mathrm{K}}_{\rho-1})$. Therefore, $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o}\mathsf{:}\mathsf{c} \in \widehat{\mathrm{K}}_{\rho-1}$, and $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, by Definition 7.2.5. Since $\widehat{\mathrm{K}}_{\rho-1} \subseteq \mathrm{T} \cup \mathrm{U}$ by the induction hypothesis and we have proved that $\mathrm{T} = \widehat{\mathrm{T}}_\infty$, it follows that $\mathsf{o}\mathsf{:}\mathsf{c} \in \mathrm{T} \cup \mathrm{U}$ and $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$. Therefore $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, by Lemma 7.2.2. Because $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, it follows that $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$, and so $\mathsf{o}[\mathsf{m}]$ is not a strong local context, by Definitions 7.2.3 and 6.1.1. Because $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{K}}_{\alpha-1} \subseteq \mathrm{T} \cup \mathrm{U}$, it follows that $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}}) \geq \mathbf{u}$.

Next we will show that there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$. Suppose on the contrary there exists $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$. Then $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$, because $\mathcal{I} \leq \mathcal{M}$. It follows that $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\infty$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\infty$, by Lemma 9.1.5. Therefore, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, by Definition 7.2.3, which contradicts the fact that $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$.

So far we have shown that $\mathsf{c}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, $\mathsf{o}[\mathsf{m}]$ is not a strong local context, and there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$. Therefore, o must either strongly or weakly inherit R in $\mathcal{I}$, by Definitions 6.3.2 and 6.3.2. So $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) \geq \mathbf{u}$.

We already know that for every literal $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$: (i) if L is a positive literal then $val^{\mathbf{b}}_{\widehat{\mathrm{K}}_{\alpha-1}}(\mathsf{L}) = \mathbf{t}$; and (ii) if L is a negative literal then $val^{\mathbf{b}}_{\widehat{\mathrm{T}}_\infty}(\mathsf{L}) = \mathbf{t}$. We will show that for all $\mathsf{L}_\mathsf{i}, 1 \leq \mathsf{i} \leq \mathsf{n}, \mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_\mathsf{i}) \geq \mathbf{u}$. If L_i is a positive literal, since $\widehat{\mathrm{K}}_{\alpha-1} \subseteq \mathrm{T} \cup \mathrm{U}$ by the induction hypothesis and $val^{\mathbf{b}}_{\widehat{\mathrm{K}}_{\alpha-1}}(\mathsf{L}_\mathsf{i}) = \mathbf{t}$, then it follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{K}}(\mathsf{L}_\mathsf{i}) = \mathbf{t}$, by Lemma 5.3.2. Thus $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_\mathsf{i}) \geq \mathbf{u}$ by Lemma 5.3.1. We have proved that $\mathrm{T} = \widehat{\mathrm{T}}_\infty$. Therefore, if L_i is a negative literal, then $\mathcal{V}^{\mathbf{b}}_{\mathcal{J}}(\mathsf{L}_\mathsf{i}) = \mathcal{V}^{\mathbf{b}}_{\widehat{\mathrm{T}}_\infty}(\mathsf{L}_\mathsf{i}) = \mathbf{t}$. Thus $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}_\mathsf{i}) \geq \mathbf{u}$ by Lemma 5.3.1.

Therefore, $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{L}) \geq \mathbf{u}$ for every literal $\mathsf{L} \in \mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}$. It follows that $\mathcal{V}^{\mathbf{b}}_{\mathcal{I}}(\mathsf{B}_{\mathsf{c}\backslash\mathsf{o}}) \geq \mathbf{u}$. Moreover, $imode_{\mathcal{I}}(\mathsf{R}||\mathsf{o}) \geq \mathbf{u}$. Because $\mathcal{I}$ is an object model of P, so $\mathcal{I}$ should satisfy $\mathsf{R}||\mathsf{o}$. It follows that $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}) \geq \mathbf{u}$ by Definition 6.3.4. Thus $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathrm{T} \cup \mathrm{U}$.

(3) $\mathsf{A} \in \mathbf{IC}^t(\widehat{\mathrm{K}}_{\alpha-1})$

If $\mathsf{A} = \mathsf{o}:\mathsf{c}$, then there exists x such that $\mathsf{o}:\mathsf{x} \in \widehat{\mathrm{K}}_{\alpha-1}$ and $\mathsf{x}::\mathsf{c} \in \widehat{\mathrm{K}}_{\alpha-1}$, by Definition 7.2.5. Since $\widehat{\mathrm{K}}_{\alpha-1} \subseteq \mathrm{T} \cup \mathrm{U}$ by the induction hypothesis, it follows that $\mathsf{o}:\mathsf{x} \in \mathrm{T} \cup \mathrm{U}$ and $\mathsf{x}::\mathsf{c} \in \mathrm{T} \cup \mathrm{U}$. So $\mathcal{I}(\mathsf{o}:\mathsf{x}) \geq \mathbf{u}$ and $\mathcal{I}(\mathsf{x}::\mathsf{c}) \geq \mathbf{u}$. Because $\mathcal{I}$ satisfies the optimistic ISA transitivity constraint, therefore $\mathcal{I}(\mathsf{o}:\mathsf{c}) \geq \mathbf{u}$ by Definitions 6.5.1 and 6.2.1. It follows that $\mathsf{o}:\mathsf{c} \in \mathrm{T} \cup \mathrm{U}$. Similarly, if $\mathsf{A} = \mathsf{s}::\mathsf{c}$, then we can also show that $\mathsf{s}::\mathsf{c} \in \mathrm{T} \cup \mathrm{U}$.

(4) $\mathsf{A} \in \mathbf{IC}^i_{\mathrm{P},\widehat{\mathrm{T}}_\infty}(\widehat{\mathrm{K}}_{\alpha-1})$

Then $\mathsf{A} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}$. Thus $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}_{\alpha-1}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{K}}_{\alpha-1}$, $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, and $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, by Definition 7.2.5. Because $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}_{\alpha-1}$, there must exist a successor ordinal $\rho \leq \alpha - 1 < \alpha$, such that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}_{\rho}$. It follows that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{T}}_\infty}(\widehat{\mathrm{K}}_{\rho-1})$. Therefore, $\mathsf{c} \neq \mathsf{o}$, $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{K}}_{\rho-1}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{z}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{K}}_{\rho-1}$ for some value z, and $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, by Definition 7.2.5. Since $\widehat{\mathrm{K}}_{\rho-1} \subseteq \widehat{\mathrm{K}}_{\alpha-1} \subseteq \mathrm{T} \cup \mathrm{U}$ by the induction hypothesis and $\mathrm{T} = \widehat{\mathrm{T}}_\infty$, it follows that $\mathsf{o}:\mathsf{c} \in \mathrm{T} \cup \mathrm{U}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \mathrm{T} \cup \mathrm{U}$, and $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$. Thus $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ by Lemma 7.2.2. Because $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, it follows that $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\mathrm{T})$, and so $\mathsf{o}[\mathsf{m}]$ is not a strong local context, by Definitions 7.2.3 and 6.1.1. Because $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \mathrm{T} \cup \mathrm{U}$, it follows that $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}}) \geq \mathbf{u}$.

Next we will show that there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$. Suppose on the contrary there exists $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$. Then $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{M}} \mathsf{o}$, because $\mathcal{I} \leq \mathcal{M}$. It follows that $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\infty$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{T}}_\infty$, by Lemma 9.1.5. Therefore, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$, by Definition 7.2.3, which contradicts the fact that $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{T}}_\infty)$.

So far we have shown that $\mathsf{o}[\mathsf{m}]$ is not a strong local context, $\mathsf{c}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{c}[\mathsf{m}] \overset{wv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$, $\mathcal{I}(\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}}) \geq \mathbf{u}$, and there is no $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{sv}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$ or $\mathsf{x}[\mathsf{m}] \overset{sc}{\rightsquigarrow}_{\mathcal{I}} \mathsf{o}$. Because $\mathcal{I}$ satisfies the optimistic inheritance constraint, therefore $\mathcal{I}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}) \geq \mathbf{u}$, by Definition 6.5.2. So $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \mathrm{T} \cup \mathrm{U}$.

□

Chapter 11

Implementation

It turns out that the optimistic object model of an F-logic program P can be computed as the well-founded model of a certain general logic program with negation which is obtained from P by rewriting.

Given an F-logic program P, we first rewrite P into a general logic program P^{wf}. Then we show that the well-founded model of P^{wf} is isomorphic to the optimistic object model of P.

11.1 Rewriting

First we will show how to rewrite the V-rules and C-rules of an F-logic program P. We can define a rewriting function that applies to all V-rules and C-rules. Note that because literals in rule heads and rule bodies have different meanings, they are rewritten differently. Moreover, literals in the heads of V-rules and in the heads of C-rules are also rewritten differently.

Definition 11.1.1 Given an F-logic program P, let L be a literal in P. The functions ρ^h and ρ^b for rewriting head and body literals in P, respectively, are defined as follows:

$$\rho^h(\mathsf{L}) = \begin{cases} isa(\mathsf{o},\mathsf{c}), & \text{if } \mathsf{L} = \mathsf{o}:\mathsf{c} \\ sub(\mathsf{s},\mathsf{c}), & \text{if } \mathsf{L} = \mathsf{s}::\mathsf{c} \\ locmvd(\mathsf{s},\mathsf{m},\mathsf{v}), & \text{if } \mathsf{L} = \mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \end{cases}$$

$$\rho^b(\mathsf{L}) = \begin{cases} isa(\mathsf{o},\mathsf{c}), & \text{if } \mathsf{L} = \mathsf{o}:\mathsf{c} \\ sub(\mathsf{s},\mathsf{c}), & \text{if } \mathsf{L} = \mathsf{s}::\mathsf{c} \\ mvd(\mathsf{o},\mathsf{m},\mathsf{v}), & \text{if } \mathsf{L} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \\ \neg\,(\rho^b(\mathrm{G})), & \text{if } \mathsf{L} = \neg\,\mathsf{G} \end{cases}$$

The rewriting function ρ on V-rules in P is defined as follows:

$$\rho(\mathsf{H} \longleftarrow \mathsf{L}_1,\ldots,\mathsf{L}_n) = \rho^h(\mathsf{H}) \longleftarrow \rho^b(\mathsf{L}_1),\ldots,\rho^b(\mathsf{L}_n)$$

And the rewriting function ρ on C-rules in P is defined as follows:

$$\rho(\mathsf{code} \quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{L}_1, \ldots, \mathsf{L}_\mathsf{n}) = codmvd(\mathsf{O}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \longleftarrow \rho^b(\mathsf{B}_1), \ldots, \rho^b(\mathsf{B}_\mathsf{n})$$

where O is a new variable that does not appear in the C-rule and $\mathsf{B}_\mathsf{i} = (\mathsf{L}_\mathsf{i})_{\mathsf{c}\backslash\mathsf{O}}$ for all $1 \leq i \leq n$, *i.e.*, B_i is obtained from L_i by substituting the new variable O for all occurrences of c.

$$
\begin{array}{rcl}
mvd(\mathsf{O},\mathsf{M},\mathsf{V}) & \longleftarrow & locmvd(\mathsf{O},\mathsf{M},\mathsf{V}). \\
mvd(\mathsf{O},\mathsf{M},\mathsf{V}) & \longleftarrow & valinhmvd(\mathsf{O},\mathsf{M},\mathsf{V},\mathsf{C}). \\
mvd(\mathsf{O},\mathsf{M},\mathsf{V}) & \longleftarrow & codinhmvd(\mathsf{O},\mathsf{M},\mathsf{V},\mathsf{C}). \\
sub(\mathsf{S},\mathsf{C}) & \longleftarrow & sub(\mathsf{S},\mathsf{X}),\ sub(\mathsf{X},\mathsf{C}). \\
isa(\mathsf{O},\mathsf{C}) & \longleftarrow & isa(\mathsf{O},\mathsf{S}),\ sub(\mathsf{S},\mathsf{C}). \\
valinhmvd(\mathsf{O},\mathsf{M},\mathsf{V},\mathsf{C}) & \longleftarrow & valcandidate(\mathsf{C},\mathsf{M},\mathsf{O}),\ locmvd(\mathsf{C},\mathsf{M},\mathsf{V}), \\
 & & \neg\, local(\mathsf{O},\mathsf{M}),\ \neg\, multiple(\mathsf{C},\mathsf{M},\mathsf{O}). \\
codinhmvd(\mathsf{O},\mathsf{M},\mathsf{V},\mathsf{C}) & \longleftarrow & codcandidate(\mathsf{C},\mathsf{M},\mathsf{O}),\ codmvd(\mathsf{O},\mathsf{M},\mathsf{V},\mathsf{C}), \\
 & & \neg\, local(\mathsf{O},\mathsf{M}),\ \neg\, multiple(\mathsf{C},\mathsf{M},\mathsf{O}). \\
valcandidate(\mathsf{C},\mathsf{M},\mathsf{O}) & \longleftarrow & isa(\mathsf{O},\mathsf{C}),\ locmvd(\mathsf{C},\mathsf{M},\mathsf{V}), \\
 & & \mathsf{C} \neq \mathsf{O},\ \neg\, override(\mathsf{C},\mathsf{M},\mathsf{O}). \\
codcandidate(\mathsf{C},\mathsf{M},\mathsf{O}) & \longleftarrow & isa(\mathsf{O},\mathsf{C}),\ coddef(\mathsf{C},\mathsf{M}), \\
 & & \mathsf{C} \neq \mathsf{O},\ \neg\, override(\mathsf{C},\mathsf{M},\mathsf{O}). \\
local(\mathsf{O},\mathsf{M}) & \longleftarrow & locmvd(\mathsf{O},\mathsf{M},\mathsf{V}). \\
multiple(\mathsf{C},\mathsf{M},\mathsf{O}) & \longleftarrow & valcandidate(\mathsf{X},\mathsf{M},\mathsf{O}),\ \mathsf{X} \neq \mathsf{C}. \\
multiple(\mathsf{C},\mathsf{M},\mathsf{O}) & \longleftarrow & codcandidate(\mathsf{X},\mathsf{M},\mathsf{O}),\ \mathsf{X} \neq \mathsf{C}. \\
override(\mathsf{C},\mathsf{M},\mathsf{O}) & \longleftarrow & sub(\mathsf{X},\mathsf{C}),\ isa(\mathsf{O},\mathsf{X}),\ locmvd(\mathsf{X},\mathsf{M},\mathsf{V}), \\
 & & \mathsf{X} \neq \mathsf{C},\ \mathsf{X} \neq \mathsf{O}. \\
override(\mathsf{C},\mathsf{M},\mathsf{O}) & \longleftarrow & sub(\mathsf{X},\mathsf{C}),\ isa(\mathsf{O},\mathsf{X}),\ coddef(\mathsf{X},\mathsf{M}), \\
 & & \mathsf{X} \neq \mathsf{C},\ \mathsf{X} \neq \mathsf{O}.
\end{array}
$$

Figure 12: Trailer Rules for Well-Founded Rewriting

Definition 11.1.2 (Well-Founded Rewriting) The well-founded rewriting of an F-logic program P, denoted P^{wf}, is a general logic program constructed by the following steps:

(1) For every V-rule R in P, add its *rewriting* $\rho(\mathsf{R})$ into P^{wf}.

(2) For every C-rule R in P, which specifies an instance method m for a class c, add its *rewriting* $\rho(\mathsf{R})$ into P^{wf}. Moreover, add a fact $codedef(\mathsf{c},\mathsf{m})$ into P^{wf}.

(3) Include the *trailer* shown in Figure 12 to P^{wf} (note that uppercase letters denote variables in these trailer rules).

While rewriting an F-logic program into a general logic program, we need to output facts of the form $codedef(\mathsf{c},\mathsf{m})$ to remember that there is a C-rule specifying the instance method m for the class c. Such atoms are used to derive overriding and code inheritance candidacy information. Clearly, given an F-logic program P, the amount of time needed to generate P^{wf} is linear in the size of P. Note that the trailer rules are fixed for an given F-logic program. Therefore, the size of P^{wf} is also linear in the size of the original F-logic program P.

11.2 Isomorphism

Given the well-founded rewriting, P^{wf}, of an F-logic program P, the Herbrand base of P^{wf}, denoted $\mathcal{HB}_{\mathrm{P}^{wf}}$, consists of atoms of the following forms: $isa/2$, $sub/2$, $locmvd/3$, $valinhmvd/4$, $codmvd/4$, $codedef/2$, $codinhmvd/4$, $mvd/3$, $valcandidate/3$, $codcandidate/3$, $local/2$, $multiple/3$, and $override/3$.

Definition 11.2.1 (Isomorphism) Let P^{wf} be the well-founded rewriting of an F-logic program P, $\mathcal{HB}_{\mathrm{P}^{wf}}$ be the Herbrand base of P^{wf}, $\widehat{\mathcal{HB}}_{\mathrm{P}}$ be the extended Herbrand base of P, I^{wf} be a subset of $\mathcal{HB}_{\mathrm{P}^{wf}}$, and $\widehat{\mathrm{I}}$ be a subset of $\widehat{\mathcal{HB}}_{\mathrm{P}}$, we say that I^{wf} is *isomorphic* to $\widehat{\mathrm{I}}$, if all of the following conditions are true:

(1) for all o,c: $isa(\mathsf{o},\mathsf{c}) \in \mathrm{I}^{wf}$ *iff* $\mathsf{o}:\mathsf{c} \in \mathrm{I}$

(2) for all s,c: $sub(\mathsf{s},\mathsf{c}) \in \mathrm{I}^{wf}$ *iff* $\mathsf{s}::\mathsf{c} \in \mathrm{I}$

(3) for all $\mathsf{s},\mathsf{m},\mathsf{v}$: $locmvd(\mathsf{s},\mathsf{m},\mathsf{v}) \in \mathrm{I}^{wf}$ *iff* $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{s}}_{\mathrm{local}} \in \mathrm{I}$

(4) for all $\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}$: $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{I}^{wf}$ *iff* $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \mathrm{I}$

(5) for all $\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}$: $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{I}^{wf}$ *iff* $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathrm{I}$

(6) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: $valcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{I}^{wf}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{I}}$

(7) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: $codcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{I}^{wf}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{I}}$

(8) for all o,m: $local(\mathsf{o},\mathsf{m}) \in \mathrm{I}^{wf}$ *iff* $lc(\mathsf{o},\mathsf{m}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$

(9) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: $multiple(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{I}^{wf}$ *iff* $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$

(10) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: $override(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{I}^{wf}$ *iff* $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$

Let $\mathcal{M}^{wf} = \langle \mathrm{T}^{wf}; \mathrm{U}^{wf} \rangle$ be the well-founded model of P^{wf}, and $\mathcal{M} = \langle \pi(\widehat{\mathrm{T}}_\infty); \pi(\widehat{\mathrm{U}}_\infty - \widehat{\mathrm{T}}_\infty) \rangle$ be the optimistic object model of P. We say that $\mathcal{M}^{wf}$ is *isomorphic* to $\mathcal{M}$, if T^{wf} and U^{wf} are isomorphic to $\widehat{\mathrm{T}}_\infty$ and $\widehat{\mathrm{U}}_\infty - \widehat{\mathrm{T}}_\infty$, respectively.

Note that the definition of isomorphism above includes atoms which do not finally appear in an interpretation of an F-logic program P. However, once we can show that the well-founded model of P^{wf} is isomorphic to the the optimistic object model $\mathcal{M}$ of P, we can establish an one-to-one mapping between $isa(\mathsf{o},\mathsf{c}) \in \mathcal{M}^{wf}$ and $\mathsf{o}:\mathsf{c} \in \mathcal{M}$, between $sub(\mathsf{s},\mathsf{c}) \in \mathcal{M}^{wf}$ and $\mathsf{s}::\mathsf{c} \in \mathcal{M}$, between $locmvd(\mathsf{s},\mathsf{m},\mathsf{v}) \in \mathcal{M}^{wf}$ and $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{s}}_{\mathrm{local}} \in \mathcal{M}$, between $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathcal{M}^{wf}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \mathcal{M}$, and between $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathcal{M}^{wf}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \mathcal{M}$, taking into account the truth values of atoms. Thus the optimistic object model of P can be effectively computed by the well-founded model of P^{wf}.

Definition 11.2.2 Let P^{wf} be the well-founded rewriting of an F-logic program P and I^{wf} be a subset of $\mathcal{HB}_{\mathrm{P}^{wf}}$. We say that I^{wf} is in *normal form*, if for all $\mathsf{o},\mathsf{m},\mathsf{v}$: $mvd(\mathsf{o},\mathsf{m},\mathsf{v}) \in \mathrm{I}^{wf}$ *iff* $locmvd(\mathsf{o},\mathsf{m},\mathsf{v}) \in \mathrm{I}^{wf}$, or there is c such that $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{I}^{wf}$ or $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{I}^{wf}$.

Lemma 11.2.1 Let P^{wf} be the well-founded rewriting of an F-logic program P and I^{wf} be a subset of $\mathcal{HB}_{\mathrm{P}^{wf}}$. Then $\mathrm{lfp}(\mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}})$ is in normal form.

Proof.

By Definitions 11.2.2, 11.1.2, 4.2.6, and 4.2.5.
□

Lemma 11.2.2 Let P^{wf} be the well-founded rewriting of an F-logic program P, $\widehat{\mathrm{I}}$ be a subset of $\widehat{\mathcal{HB}}_{\mathrm{P}}$, I^{wf} be a subset of $\mathcal{HB}_{\mathrm{P}^{wf}}$ which is isomorphic to $\widehat{\mathrm{I}}$ and is in normal form, and G be a ground positive literal. Then $val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\neg\,\mathsf{G}) = \mathbf{t}$ iff $\rho^b(\mathsf{G}) \notin \mathrm{I}^{wf}$.

Proof.

There are three cases to consider:

(1) $\mathsf{G} = \mathsf{o}:\mathsf{c}$

Then $\rho^b(\mathsf{G}) = isa(\mathsf{o},\mathsf{c})$. It follows that $val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\neg\,\mathsf{s}:\mathsf{c}) = \mathbf{t}$, iff $\mathsf{s}:\mathsf{c} \notin \widehat{\mathrm{I}}$, by Definition 5.3.1, iff $isa(\mathsf{o},\mathsf{c}) \notin \mathrm{I}^{wf}$, because I^{wf} is isomorphic to $\widehat{\mathrm{I}}$.

(2) $\mathsf{G} = \mathsf{s}::\mathsf{c}$

Then $\rho^b(\mathsf{G}) = sub(\mathsf{s},\mathsf{c})$. It follows that $val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\neg\,\mathsf{s}::\mathsf{c}) = \mathbf{t}$, iff $\mathsf{s}::\mathsf{c} \notin \widehat{\mathrm{I}}$, by Definition 5.3.1, iff $sub(\mathsf{s},\mathsf{c}) \notin \mathrm{I}^{wf}$, because I^{wf} is isomorphic to $\widehat{\mathrm{I}}$.

(3) $\mathsf{G} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$

Then $\rho^b(\mathsf{G}) = mvd(\mathsf{o},\mathsf{m},\mathsf{v})$. It follows that $val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\neg\, \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$, iff $val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{f}$, iff $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{local}} \notin \widehat{\mathrm{I}}$, $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \notin \widehat{\mathrm{I}}$ and $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \notin \widehat{\mathrm{I}}$ for all c, iff $locmvd(\mathsf{o},\mathsf{m},\mathsf{v}) \notin \mathrm{I}^{wf}$, $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \notin \mathrm{I}^{wf}$ and $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \notin \mathrm{I}^{wf}$ for all c, because I^{wf} is isomorphic to $\widehat{\mathrm{I}}$, thus iff $mvd(\mathsf{o},\mathsf{m},\mathsf{v}) \notin \mathrm{I}^{wf}$, since I^{wf} is in normal form.

□

Proposition 11.2.3 Let P^{wf} be the well-founded rewriting of an F-logic program P, I^{wf} be a subset of $\mathcal{HB}_{\mathrm{P}^{wf}}$ which is in normal form, and $\widehat{\mathrm{I}}$ be a subset of $\widehat{\mathcal{HB}}_{\mathrm{P}}$. If I^{wf} is isomorphic to $\widehat{\mathrm{I}}$, then $\mathrm{lfp}(\mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}})$ is isomorphic to $\mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}})$.

Proof.

Let $\mathrm{J}^{wf} = \mathrm{lfp}(\mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}})$ and $\widehat{\mathrm{J}} = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}})$. First we will show that all of the following conditions are true:

(1) for all o,c: $isa(\mathsf{o},\mathsf{c}) \in \mathrm{J}^{wf}$ *iff* $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{J}}$

(2) for all s,c: $sub(\mathsf{s},\mathsf{c}) \in \mathrm{J}^{wf}$ *iff* $\mathsf{s}::\mathsf{c} \in \widehat{\mathrm{J}}$

(3) for all $\mathsf{s},\mathsf{m},\mathsf{v}$: $locmvd(\mathsf{s},\mathsf{m},\mathsf{v}) \in \mathrm{J}^{wf}$ *iff* $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{s}}_{\mathrm{local}} \in \widehat{\mathrm{J}}$

(4) for all $\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}$: $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{J}^{wf}$ *iff* $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \widehat{\mathrm{J}}$

(5) for all $\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}$: $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{J}^{wf}$ *iff* $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \widehat{\mathrm{J}}$

(6) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: $valcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{J}^{wf}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$

(7) for all $\mathsf{c},\mathsf{m},\mathsf{o}$: $codcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{J}^{wf}$ *iff* $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$

I. $\Rightarrow$

First let us define:

$$
\begin{array}{llllll}
\mathrm{S}^{wf}_0 = \emptyset & & \widehat{\mathrm{S}}_0 = \emptyset & & \text{for limit ordinal } 0 \\
\mathrm{S}^{wf}_\alpha = \mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}}(\mathrm{S}^{wf}_{\alpha-1}) & & \widehat{\mathrm{S}}_\alpha = \mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) & & \text{for successor ordinal } \alpha \\
\mathrm{S}^{wf}_\alpha = \bigcup\limits_{\beta<\alpha} \mathrm{S}^{wf}_\beta & & \widehat{\mathrm{S}}_\alpha = \bigcup\limits_{\beta<\alpha} \widehat{\mathrm{S}}_\beta & & \text{for limit ordinal } \alpha \neq 0 \\
\mathrm{S}^{wf}_\infty = \bigcup\limits_{\alpha} \mathrm{S}^{wf}_\alpha & & \widehat{\mathrm{S}}_\infty = \bigcup\limits_{\alpha} \widehat{\mathrm{S}}_\alpha & &
\end{array}
$$

Then $\mathrm{S}^{wf}_\infty = \mathrm{lfp}(\mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}})$ and $\widehat{\mathrm{S}}_\infty = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}})$. We will prove by transfinite induction that for any ordinal α and for all $\mathsf{o},\mathsf{s},\mathsf{c},\mathsf{m},\mathsf{v}$, the following conditions are true:

(1) if $isa(\mathsf{o},\mathsf{c}) \in \mathrm{S}^{wf}_\alpha$ then $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{S}}_\alpha$

(2) if $sub(\mathsf{s},\mathsf{c}) \in \mathrm{S}^{wf}_\alpha$ then $\mathsf{s}::\mathsf{c} \in \widehat{\mathrm{S}}_\alpha$

(3) if $locmvd(\mathsf{s},\mathsf{m},\mathsf{v}) \in \mathrm{S}^{wf}_{\alpha}$ then $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{s}}_{\mathrm{local}} \in \widehat{\mathrm{S}}_{\alpha}$

(4) if $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha}$ then $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \widehat{\mathrm{S}}_{\alpha}$

(5) if $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha}$ then $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \widehat{\mathrm{S}}_{\alpha}$

(6) if $valcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{S}^{wf}_{\alpha}$ then $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{S}}_{\alpha}$

(7) if $codcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{S}^{wf}_{\alpha}$ then $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{S}}_{\alpha}$

The case for a limit ordinal α is trivial. Now let α be a successor ordinal. So $\mathrm{S}^{wf}_{\alpha} = \mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}}(\mathrm{S}^{wf}_{\alpha-1})$.

First we show that for any ground positive literal L, if $\rho^b(\mathsf{L}) \in \mathrm{S}^{wf}_{\alpha-1}$, then $val^{\mathbf{b}}_{\widehat{\mathrm{S}}_{\alpha-1}}(\mathsf{L}) = \mathbf{t}$: (i) If $\mathsf{L} = \mathsf{o}:\mathsf{c}$, then $\rho^b(\mathsf{L}) = isa(\mathsf{o},\mathsf{c})$. It follows that $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{S}}_{\alpha-1}$ by the induction hypothesis. Thus $val^{\mathbf{b}}_{\widehat{\mathrm{S}}_{\alpha-1}}(\mathsf{o}:\mathsf{c}) = \mathbf{t}$; (ii) Similarly, we can show that if $\rho^b(\mathsf{L}) = sub(\mathsf{s},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha-1}$, then $val^{\mathbf{b}}_{\widehat{\mathrm{S}}_{\alpha-1}}(\mathsf{s}::\mathsf{c}) = \mathbf{t}$; (iii) If $\mathsf{L} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$, then $\rho^b(\mathsf{L}) = mvd(\mathsf{o},\mathsf{m},\mathsf{v})$. Because $\mathrm{S}^{wf}_{\gamma} \subseteq \mathrm{S}^{wf}_{\alpha-1}$ for all $\gamma \leq \alpha - 1$ by Proposition 4.1.3, there must exist a successor ordinal $\rho \leq \alpha - 1$ such that $mvd(\mathsf{s},\mathsf{m},\mathsf{v}) \in \mathrm{S}^{wf}_{\rho} = \mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}}(\mathrm{S}^{wf}_{\rho-1})$. It follows that $locmvd(\mathsf{o},\mathsf{m},\mathsf{v}) \in \mathrm{S}^{wf}_{\rho-1}$, or there is c such that $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{S}^{wf}_{\rho-1}$ or $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{S}^{wf}_{\rho-1}$, according to the trailer rules in Definition 11.1.2. Thus $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\circ}_{\mathrm{local}} \in \widehat{\mathrm{S}}_{\rho-1}$, or there is c such that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \widehat{\mathrm{S}}_{\rho-1}$ or $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}} \in \widehat{\mathrm{S}}_{\rho-1}$, by the induction hypothesis. Moreover, $\widehat{\mathrm{S}}_{\rho-1} \subseteq \widehat{\mathrm{S}}_{\alpha-1}$ by Proposition 4.1.3. It follows that $val^{\mathbf{b}}_{\widehat{\mathrm{S}}_{\alpha-1}}(\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$.

Now consider the following cases:

(1) $isa(\mathsf{o},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha}$ and $isa(\mathsf{o},\mathsf{c})$ is derived via a rule $\mathsf{R}^{wf} \in ground(\mathrm{P}^{wf})$ which is rewritten from a V-rule $\mathsf{R} \in ground(\mathrm{P})$.

Then $\mathsf{R}^{wf} \equiv isa(\mathsf{o},\mathsf{c}) \longleftarrow \rho^b(\mathsf{C}_1),\ldots,\rho^b(\mathsf{C}_\mathsf{m}),\neg\,\rho^b(\mathsf{G}_1),\ldots,\neg\,\rho^b(\mathsf{G}_\mathsf{n})$, such that R^{wf} is the rewriting of $\mathsf{R} \equiv \mathsf{o}:\mathsf{c} \longleftarrow \mathsf{C}_1,\ldots,\mathsf{C}_\mathsf{m},\neg\,\mathsf{G}_1,\ldots,\neg\,\mathsf{G}_\mathsf{n}$, where C_i $(1 \leq \mathsf{i} \leq \mathsf{m})$ and G_j $(1 \leq \mathsf{j} \leq \mathsf{n})$ are positive literals. By Definition 4.2.5, $\rho^b(\mathsf{C}_\mathsf{i}) \in \mathrm{S}^{wf}_{\alpha-1}$ for all $1 \leq \mathsf{i} \leq \mathsf{m}$, and $\rho^b(\mathsf{G}_\mathsf{j}) \notin \mathrm{I}^{wf}$ for all $1 \leq \mathsf{j} \leq \mathsf{n}$. Following the above claim, $val^{\mathbf{b}}_{\widehat{\mathrm{S}}_{\alpha-1}}(\mathsf{C}_\mathsf{i}) = \mathbf{t}$ for all $1 \leq \mathsf{i} \leq \mathsf{m}$. Moreover, I^{wf} is isomorphic to $\widehat{\mathrm{I}}$ and is in normal form, therefore $val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\neg\,\mathsf{G}_\mathsf{j}) = \mathbf{t}$ for all $1 \leq \mathsf{j} \leq \mathsf{n}$, by Lemma 11.2.2. So $\mathsf{o}:\mathsf{c} \in \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) \subseteq \mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) = \widehat{\mathrm{S}}_{\alpha}$, by Definitions 7.2.2 and 7.2.6.

(2) $isa(\mathsf{o},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha}$ and $isa(\mathsf{o},\mathsf{c})$ is derived via a trailer rule R^{wf} in $ground(\mathrm{P}^{wf})$.

Then there exists s such that $\mathsf{R}^{wf} = isa(\mathsf{o},\mathsf{c}) \longleftarrow isa(\mathsf{o},\mathsf{s}),sub(\mathsf{s},\mathsf{c})$. It follows that $isa(\mathsf{o},\mathsf{s}) \in \mathrm{S}^{wf}_{\alpha-1}$ and $sub(\mathsf{s},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha-1}$. Thus $\mathsf{o}:\mathsf{s} \in \widehat{\mathrm{S}}_{\alpha-1}$ and $\mathsf{s}::\mathsf{c} \in \widehat{\mathrm{S}}_{\alpha-1}$ by the induction hypothesis. So $\mathsf{o}:\mathsf{c} \in \mathbf{IC}^t(\widehat{\mathrm{S}}_{\alpha-1}) \subseteq \mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) = \widehat{\mathrm{S}}_{\alpha}$, by Definitions 7.2.5 and 7.2.6.

(3) $sub(\mathsf{s},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha}$ and $sub(\mathsf{s},\mathsf{c})$ is derived via a rule $\mathsf{R}^{wf} \in ground(\mathrm{P}^{wf})$ which is rewritten from a V-rule $\mathsf{R} \in ground(\mathrm{P})$.

Similarly to (1), we can show that $\mathsf{s} :: \mathsf{c} \in \widehat{\mathrm{S}}_{\alpha}$.

(4) $sub(\mathsf{s},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha}$ and $sub(\mathsf{s},\mathsf{c})$ is derived via a trailer rule R^{wf} in $ground(\mathrm{P}^{wf})$.

Similarly to (2), we can show that $\mathsf{s} :: \mathsf{c} \in \widehat{\mathrm{S}}_{\alpha}$.

(5) $locmvd(\mathsf{s},\mathsf{m},\mathsf{v}) \in \mathrm{S}^{wf}_{\alpha}$

Then $locmvd(\mathsf{s},\mathsf{m},\mathsf{v})$ must be derived via a rule $\mathsf{R}^{wf} \in ground(\mathrm{P}^{wf})$ which is rewritten from a V-rule $\mathsf{R} \in ground(\mathrm{P})$. Similarly to (1), we can also show that $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{s}}_{\mathrm{local}} \in \widehat{\mathrm{S}}_{\alpha}$.

(6) $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha}$

By Definition 11.1.2, $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c})$ must be derived via a trailer rule in $ground(\mathrm{P}^{wf})$. It follows that $valcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{S}^{wf}_{\alpha-1}$, $locmvd(\mathsf{c},\mathsf{m},\mathsf{v}) \in \mathrm{S}^{wf}_{\alpha-1}$, $local(\mathsf{o},\mathsf{m}) \notin \mathrm{I}^{wf}$, and $multiple(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathrm{I}^{wf}$, by Definition 11.1.2. So $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{S}}_{\alpha-1}$ and $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{S}}_{\alpha-1}$, by the induction hypothesis. Moreover, $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$ and $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$, since I^{wf} is isomorphic to $\widehat{\mathrm{I}}$. It follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{value}} \in \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) \subseteq \mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) = \widehat{\mathrm{S}}_{\alpha}$, by Definitions 7.2.5 and 7.2.6.

(7) $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha}$

By Definition 11.1.2, $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c})$ must be derived via a trailer rule in $ground(\mathrm{P}^{wf})$. It follows that $codcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{S}^{wf}_{\alpha-1}$, $codmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha-1}$, $local(\mathsf{o},\mathsf{m}) \notin \mathrm{I}^{wf}$, and $multiple(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathrm{I}^{wf}$, by Definition 11.1.2. So $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{S}}_{\alpha-1}$ by the induction hypothesis. Moreover, $lc(\mathsf{o},\mathsf{m}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$ and $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$, since I^{wf} is isomorphic to $\widehat{\mathrm{I}}$.

Note that by Definition 11.1.2, $codmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c})$ must be derived via a rule $\mathsf{R}^{wf} \equiv codmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \longleftarrow \rho^{b}(\mathsf{C}_1),\ldots,\rho^{b}(\mathsf{C}_{\mathsf{m}}),\neg\,\rho^{b}(\mathsf{G}_1),\ldots,\neg\,\rho^{b}(\mathsf{G}_{\mathsf{n}})$, in $ground(\mathrm{P}^{wf})$, which is rewritten from the following C-rule in $ground(\mathrm{P})$, $\mathsf{R} \equiv \mathsf{code}\quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{B}_1,\ldots,\mathsf{B}_{\mathsf{m}},\neg\,\mathsf{F}_1,\ldots,\neg\,\mathsf{F}_{\mathsf{n}}$, where B_{i} and F_{j} are positive literals, $\mathsf{C}_{\mathsf{i}} = (\mathsf{B}_{\mathsf{i}})_{\mathsf{c}\backslash\mathsf{o}}$ and $\mathsf{G}_{\mathsf{j}} = (\mathsf{F}_{\mathsf{j}})_{\mathsf{c}\backslash\mathsf{o}}$, for all $1 \le \mathsf{i} \le \mathsf{m}$ and $1 \le \mathsf{j} \le \mathsf{n}$. Similarly to (1), we can show that $val^{\mathbf{b}}_{\widehat{\mathrm{S}}_{\alpha-1}}((\mathsf{B}_{\mathsf{i}})_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{t}$ for all $1 \le \mathsf{i} \le \mathsf{m}$ and $val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\neg\,(\mathsf{F}_{\mathsf{j}})_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{t}$ for all $1 \le \mathsf{j} \le \mathsf{n}$. It follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{value}} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) \subseteq \mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) = \widehat{\mathrm{S}}_{\alpha}$, by Definitions 7.2.4 and 7.2.6.

(8) $valcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{S}^{wf}_{\alpha}$

By Definition 11.1.2, $valcandidate(\mathsf{c},\mathsf{m},\mathsf{o})$ must be derived via a trailer rule in $ground(\mathrm{P}^{wf})$. It follows that $isa(\mathsf{o},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha-1}$, $locmvd(\mathsf{c},\mathsf{m},\mathsf{v}) \in \mathrm{S}^{wf}_{\alpha-1}$, $\mathsf{c} \neq \mathsf{o}$, and $override(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathrm{I}^{wf}$, by Definition 11.1.2. So $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{S}}_{\alpha-1}$ and $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{S}}_{\alpha-1}$, by the induction hypothesis. Moreover, $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$, since I^{wf} is isomorphic to $\widehat{\mathrm{I}}$. It follows that $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) \subseteq \mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) = \widehat{\mathrm{S}}_{\alpha}$, by Definitions 7.2.5 and 7.2.6.

(9) $codcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{S}^{wf}_{\alpha}$

By Definition 11.1.2, $codcandidate(\mathsf{c},\mathsf{m},\mathsf{o})$ must be derived via a trailer rule in $ground(\mathrm{P}^{wf})$. It follows that $isa(\mathsf{o},\mathsf{c}) \in \mathrm{S}^{wf}_{\alpha-1}$, $coddef(\mathsf{c},\mathsf{m}) \in \mathrm{S}^{wf}_{\alpha-1}$, $\mathsf{c} \neq \mathsf{o}$, and $override(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathrm{I}^{wf}$, by Definition 11.1.2. So $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{S}}_{\alpha-1}$ by the induction hypothesis, and $ov(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$, because I^{wf} is isomorphic to $\widehat{\mathrm{I}}$. Moreover, since $coddef(\mathsf{c},\mathsf{m}) \in \mathrm{S}^{wf}_{\alpha-1}$, therefore there is a C-rule in P which specifies the instance method m for the class c, according to Definition 11.1.2. It follows that $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) \subseteq \mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{S}}_{\alpha-1}) = \widehat{\mathrm{S}}_{\alpha}$, by Definitions 7.2.5 and 7.2.6.

II. $\Leftarrow$

First construct an extended atom set $\widehat{\mathrm{K}}$ from J^{wf} as follows: generate one $\mathsf{o}:\mathsf{c}$ in $\widehat{\mathrm{K}}$ for every $isa(\mathsf{o},\mathsf{c})$ in J^{wf}, one $\mathsf{s}::\mathsf{c}$ in $\widehat{\mathrm{K}}$ for every $sub(\mathsf{s},\mathsf{c})$ in J^{wf}, one $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{s}}_{\mathrm{local}}$ in $\widehat{\mathrm{K}}$ for every $locmvd(\mathsf{s},\mathsf{m},\mathsf{v})$ in J^{wf}, one $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}$ in $\widehat{\mathrm{K}}$ for every $valinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c})$ in J^{wf}, one $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{code}}$ in $\widehat{\mathrm{K}}$ for every $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c})$ in J^{wf}, one $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o}$ in $\widehat{\mathrm{K}}$ for every $valcandidate(\mathsf{c},\mathsf{m},\mathsf{o})$ in J^{wf}, and one $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o}$ in $\widehat{\mathrm{K}}$ for every $codcandidate(\mathsf{c},\mathsf{m},\mathsf{o})$ in J^{wf}. Clearly, to prove that the conditions are true, it suffices to show that $\widehat{\mathrm{K}} \supseteq \widehat{\mathrm{J}}$.

We will show that $\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}}) \subseteq \widehat{\mathrm{K}}$. Thus $\widehat{\mathrm{K}} \supseteq \widehat{\mathrm{J}}$ by Proposition 4.1.2, because $\widehat{\mathrm{J}} = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}})$. Recall that by Definitions 7.2.6 and 7.2.5,

$$\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}}) = \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}}) \cup \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}}) \cup \mathbf{IC}^{t}(\widehat{\mathrm{K}}) \cup \mathbf{IC}^{c}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}}) \cup \mathbf{IC}^{i}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}})$$

Let A be any atom in $\mathbf{T}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}})$. Consider the following cases:

(1) $\mathsf{A} \in \mathbf{VC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}})$

Then there is a V-rule $\mathsf{R} \equiv \mathsf{H} \longleftarrow \mathsf{C}_1,\ldots,\mathsf{C}_{\mathsf{m}},\neg\,\mathsf{G}_1,\ldots,\neg\,\mathsf{G}_{\mathsf{n}}$ in $ground(\mathrm{P})$, such that H matches A, C_{i} $(1 \leq \mathsf{i} \leq \mathsf{m})$ and G_{j} $(1 \leq \mathsf{j} \leq \mathsf{n})$ are positive literals, $val^{\mathbf{b}}_{\widehat{\mathrm{K}}}(\mathsf{C}_{\mathsf{i}}) = \mathbf{t}$ for all $1 \leq \mathsf{i} \leq \mathsf{m}$ and $val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\neg\,\mathsf{G}_{\mathsf{j}}) = \mathbf{t}$ for all $1 \leq \mathsf{j} \leq \mathsf{n}$. Consider the rewriting R^{wf} of R, $\rho^{h}(\mathsf{H}) \longleftarrow \rho^{b}(\mathsf{C}_1),\ldots,\rho^{b}(\mathsf{C}_{\mathsf{m}}),\neg\,\rho^{b}(\mathsf{G}_1),\ldots,\neg\,\rho^{b}(\mathsf{G}_{\mathsf{n}})$. First we will show that $\rho^{b}(\mathsf{C}_{\mathsf{i}}) \in \mathrm{J}^{wf}$ for all $1 \leq \mathsf{i} \leq \mathsf{m}$: (i) If $\mathsf{C}_{\mathsf{i}} = \mathsf{o}:\mathsf{c}$, then

$\rho^b(\mathsf{C_i}) = isa(\mathsf{o},\mathsf{c})$. Since $val^{\mathbf{b}}_{\widehat{\mathrm{K}}}(\mathsf{o}:\mathsf{c}) = \mathbf{t}$, it follows that $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{K}}$ by Definition 5.3.1. Therefore, $isa(\mathsf{o},\mathsf{c}) \in \mathrm{J}^{wf}$, by the construction of $\widehat{\mathrm{K}}$; (ii) Similarly, we can show that $\mathsf{C_i} = \mathsf{s}::\mathsf{c}$, then $\rho^b(\mathsf{C_i}) = sub(\mathsf{s},\mathsf{c}) \in \mathrm{J}^{wf}$; (iii) If $\mathsf{C_i} = \mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$, then $\rho^b(\mathsf{C_i}) = mvd(\mathsf{s},\mathsf{m},\mathsf{v})$. Since $val^{\mathbf{b}}_{\widehat{\mathrm{K}}}(\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]) = \mathbf{t}$, it follows that $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{s}}_{\mathrm{local}} \in \widehat{\mathrm{K}}$, or there exists c such that $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{value}} \in \widehat{\mathrm{K}}$ or $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}} \in \widehat{\mathrm{K}}$. So $locmvd(\mathsf{s},\mathsf{m},\mathsf{v}) \in \mathrm{J}^{wf}$, or there exists c such that $valinhmvd(\mathsf{s},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{J}^{wf}$ or $codinhmvd(\mathsf{s},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathrm{J}^{wf}$, by the construction of $\widehat{\mathrm{K}}$. Because $\mathrm{J}^{wf} = \mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}}(\mathrm{J}^{wf})$, therefore $mvd(\mathsf{s},\mathsf{m},\mathsf{v}) \in \mathrm{J}^{wf}$, according to the trailer rules in Definition 11.1.2.

By Lemma 11.2.2, $\rho^b(\mathsf{G_j}) \notin \mathrm{I}^{wf}$ for all $1 \leq \mathsf{j} \leq \mathsf{n}$. So $\rho^h(\mathsf{H}) \in \mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}}(\mathrm{J}^{wf}) = \mathrm{J}^{wf}$, by Definition 4.2.5. It follows that: (i) If $\mathsf{A} = \mathsf{o}:\mathsf{c}$, then $\mathsf{H} = \mathsf{o}:\mathsf{c}$. So $\rho^h(\mathsf{H}) = isa(\mathsf{o},\mathsf{c}) \in \mathrm{J}^{wf}$, thus $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{K}}$; (ii) Similarly, if $\mathsf{A} = \mathsf{s}::\mathsf{c}$, then $\mathsf{s}::\mathsf{c} \in \widehat{\mathrm{K}}$; (iii) If $\mathsf{A} = \mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{s}}_{\mathrm{local}}$, then $\mathsf{H} = \mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$. So $\rho^h(\mathsf{H}) = locmvd(\mathsf{s},\mathsf{m},\mathsf{v}) \in \mathrm{J}^{wf}$, thus $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{s}}_{\mathrm{local}} \in \widehat{\mathrm{K}}$.

(2) $\mathsf{A} \in \mathbf{CC}_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}})$

Then $\mathsf{A} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}}$. It follows that $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}$, $lc(\mathsf{o},\mathsf{c}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$, $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$, and there is a C-rule R in $ground(\mathrm{P})$, $\mathsf{R} \equiv \mathsf{code}\quad \mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}] \longleftarrow \mathsf{C_1},\ldots,\mathsf{C_m},\neg\,\mathsf{G_1},\ldots,\neg\,\mathsf{G_n}$, such that $\mathsf{C_i}$ $(1 \leq \mathsf{i} \leq \mathsf{m})$ and $\mathsf{G_j}$ $(1 \leq \mathsf{j} \leq \mathsf{n})$ are positive literals, $val^{\mathbf{b}}_{\widehat{\mathrm{K}}}((\mathsf{C_i})_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{t}$ for all $1 \leq \mathsf{i} \leq \mathsf{m}$ and $val^{\mathbf{b}}_{\widehat{\mathrm{I}}}(\neg\,(\mathsf{G_j})_{\mathsf{c}\backslash\mathsf{o}}) = \mathbf{t}$ for all $1 \leq \mathsf{j} \leq \mathsf{n}$. Consider the rewriting R^{wf} of R, $\mathsf{R}^{wf} \equiv codmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \longleftarrow \rho^b(\mathsf{B_1}),\ldots,\rho^b(\mathsf{B_m}),\neg\,\rho^b(\mathsf{F_1}),\ldots,\neg\,\rho^b(\mathsf{F_n})$, where $\mathsf{B_i} = (\mathsf{C_i})_{\mathsf{c}\backslash\mathsf{o}}$ for all $1 \leq \mathsf{i} \leq \mathsf{m}$ and $\mathsf{F_j} = (\mathsf{G_j})_{\mathsf{c}\backslash\mathsf{o}}$ for all $1 \leq \mathsf{j} \leq \mathsf{n}$. Similarly to (1), we can also show that $\rho^b(\mathsf{B_i}) \in \mathrm{J}^{wf}$ for all $1 \leq \mathsf{i} \leq \mathsf{m}$. By Lemma 11.2.2, $\rho^b(\mathsf{F_j}) \notin \mathrm{I}^{wf}$ for all $1 \leq \mathsf{j} \leq \mathsf{n}$. So $codmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}}(\mathrm{J}^{wf}) = \mathrm{J}^{wf}$, by Definition 4.2.5.

Because $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}$, therefore $codcandidate(\mathsf{c},\mathsf{m},\mathsf{o}) \in \mathrm{J}^{wf}$, by the construction of $\widehat{\mathrm{K}}$. Note that $lc(\mathsf{o},\mathsf{c}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$ and $mc(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{I}})$. Since I^{wf} is isomorphic to $\widehat{\mathrm{I}}$, it follows that $local(\mathsf{o},\mathsf{c}) \notin \mathrm{I}^{wf}$ and $multiple(\mathsf{c},\mathsf{m},\mathsf{o}) \notin \mathrm{I}^{wf}$. So $codinhmvd(\mathsf{o},\mathsf{m},\mathsf{v},\mathsf{c}) \in \mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}}(\mathrm{J}^{wf}) = \mathrm{J}^{wf}$, according to the trailer rules of P^{wf} and Definition 4.2.5. It follows that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\,\mathsf{c}}_{\mathrm{code}} \in \widehat{\mathrm{K}}$.

(3) $\mathsf{A} \in \mathbf{IC}^t(\widehat{\mathrm{K}})$

If $\mathsf{A} = \mathsf{o}:\mathsf{c}$, then there exists x such that $\mathsf{o}:\mathsf{x} \in \widehat{\mathrm{K}}$ and $\mathsf{x}::\mathsf{c} \in \widehat{\mathrm{K}}$. So $isa(\mathsf{o},\mathsf{x}) \in \mathrm{J}^{wf}$ and $sub(\mathsf{x},\mathsf{c}) \in \mathrm{J}^{wf}$, by the construction of $\widehat{\mathrm{K}}$. It follows that $isa(\mathsf{o},\mathsf{c}) \in \mathbf{C}_{\mathrm{P}^{wf},\mathrm{I}^{wf}}(\mathrm{J}^{wf}) = \mathrm{J}^{wf}$, by Definition 4.2.5 and the trailer rules of P^{wf}. Thus $\mathsf{o}:\mathsf{c} \in \widehat{\mathrm{K}}$. Similarly, if $\mathsf{A} = \mathsf{s}::\mathsf{c}$, we can also show that $\mathsf{s}::\mathsf{c} \in \widehat{\mathrm{K}}$.

(4) $\mathsf{A} \in \mathbf{IC}^c_{\mathrm{P},\widehat{\mathrm{I}}}(\widehat{\mathrm{K}})$

If $\mathsf{A} = \mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o}$, then $\mathsf{o} : \mathsf{c} \in \widehat{\mathrm{K}}$, $\mathsf{c} \neq \mathsf{o}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{K}}$, and $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \widehat{\mathrm{I}}$, by Definition 7.2.5. Because $\widehat{\mathrm{K}}$ is constructed from J^{wf} and I^{wf} is isomorphic to $\widehat{\mathrm{I}}$, it follows that $isa(\mathsf{o}, \mathsf{c}) \in \mathrm{J}^{wf}$, $locmvd(\mathsf{c}, \mathsf{m}, \mathsf{v}) \in \mathrm{J}^{wf}$, and $override(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathrm{I}^{wf}$. So $valcandidate(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathbf{C}_{\mathrm{P}^{wf}, \mathrm{I}^{wf}}(\mathrm{J}^{wf}) = \mathrm{J}^{wf}$, by Definition 4.2.5 and the trailer rules of P^{wf}. Thus $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}$. Similarly, if $\mathsf{A} = \mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o}$, we can also show that $\mathsf{c}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}$.

(5) $\mathsf{A} \in \mathbf{IC}^{i}_{\mathrm{P}, \widehat{\mathrm{I}}}(\widehat{\mathrm{K}})$

Then $\mathsf{A} = \mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}}$, and $\mathsf{c}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{K}}$, $\mathsf{c}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{local}} \in \widehat{\mathrm{K}}$, $lc(\mathsf{o}, \mathsf{m}) \notin \widehat{\mathrm{I}}$, $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \widehat{\mathrm{I}}$, by Definition 7.2.5. Because $\widehat{\mathrm{K}}$ is constructed from J^{wf}, and I^{wf} is isomorphic to $\widehat{\mathrm{I}}$, it follows that $valcandidate(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathrm{J}^{wf}$, $locmvd(\mathsf{c}, \mathsf{m}, \mathsf{v}) \in \mathrm{J}^{wf}$, $local(\mathsf{o}, \mathsf{m}) \notin \mathrm{I}^{wf}$, $multiple(\mathsf{c}, \mathsf{m}, \mathsf{o}) \notin \mathrm{I}^{wf}$. So by Definition 4.2.5 and the trailer rules of P^{wf}, $valinhmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathbf{C}_{\mathrm{P}^{wf}, \mathrm{I}^{wf}}(\mathrm{J}^{wf}) = \mathrm{J}^{wf}$. Thus $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{c}}_{\mathrm{value}} \in \widehat{\mathrm{K}}$.

Finally, to finish the proof that J^{wf} is isomorphic to $\widehat{\mathrm{J}}$, we still need to show that the following conditions are true:

(1) for all o, m: $local(\mathsf{o}, \mathsf{m}) \in \mathrm{J}^{wf}$ *iff* $lc(\mathsf{o}, \mathsf{m}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$

(2) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $multiple(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathrm{J}^{wf}$ *iff* $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$

(3) for all $\mathsf{c}, \mathsf{m}, \mathsf{o}$: $override(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathrm{J}^{wf}$ *iff* $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$

Note that $local/2$, $multiple/3$, and $override/3$ can only be derived via the trailer rules as defined in Definition 11.1.2. Moreover, $\mathrm{J}^{wf} = \mathbf{C}_{\mathrm{P}^{wf}, \mathrm{I}^{wf}}(\mathrm{J}^{wf})$. It follows that:

(1) $local(\mathsf{o}, \mathsf{m}) \in \mathrm{J}^{wf}$, iff there exists v such that $locmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}) \in \mathrm{J}^{wf}$, iff there exits v such that $\mathsf{o}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{o}}_{\mathrm{local}} \in \widehat{\mathrm{J}}$, iff $lc(\mathsf{o}, \mathsf{m}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$.

(2) $multiple(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathrm{J}^{wf}$, iff there exists $\mathsf{x} \neq \mathsf{c}$ such that $valcandidate(\mathsf{x}, \mathsf{m}, \mathsf{o}) \in \mathrm{J}^{wf}$ or $codcandidate(\mathsf{x}, \mathsf{m}, \mathsf{o}) \in \mathrm{J}^{wf}$, iff there exists $\mathsf{x} \neq \mathsf{c}$ such that $\mathsf{x}[\mathsf{m}] \overset{v}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$ or $\mathsf{x}[\mathsf{m}] \overset{c}{\rightsquigarrow} \mathsf{o} \in \widehat{\mathrm{J}}$, iff $mc(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$.

(3) $override(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathrm{J}^{wf}$, iff there is x, such that $\mathsf{x} \neq \mathsf{c}$, $\mathsf{x} \neq \mathsf{o}$, $sub(\mathsf{x}, \mathsf{c}) \in \mathrm{J}^{wf}$, $isa(\mathsf{o}, \mathsf{x}) \in \mathrm{J}^{wf}$, and there is v such that $locmvd(\mathsf{x}, \mathsf{m}, \mathsf{v}) \in \mathrm{J}^{wf}$ or $coddef(\mathsf{x}, \mathsf{m}) \in \mathrm{J}^{wf}$, iff there is x such that $\mathsf{x} \neq \mathsf{c}$, $\mathsf{x} \neq \mathsf{o}$, $\mathsf{x} :: \mathsf{c} \in \widehat{\mathrm{J}}$, $\mathsf{o} : \mathsf{x} \in \widehat{\mathrm{J}}$, and there is v such that $\mathsf{x}[\mathsf{m} \twoheadrightarrow \mathsf{v}]^{\mathsf{x}}_{\mathrm{local}} \in \widehat{\mathrm{J}}$ or there is a C-rule in P which specifies the instance method m for the class c, iff $ov(\mathsf{c}, \mathsf{m}, \mathsf{o}) \in \mathbf{IB}_{\mathrm{P}}(\widehat{\mathrm{J}})$.

□

Before we finally show the main theorem of this section, we need to introduce notations to represent the intermediate results during the computation of the well-founded model of a given program.

Definition 11.2.3 Given the well-founded rewriting P^{wf} of an F-logic program P, define:

$$\begin{array}{llllll}
\mathrm{T}_0^{wf} & = & \emptyset & \mathrm{U}_0^{wf} & = & \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{T}_0^{wf}) \quad \text{for limit ordinal } 0 \\
\mathrm{T}_\alpha^{wf} & = & \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{U}_{\alpha-1}^{wf}) & \mathrm{U}_\alpha^{wf} & = & \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{T}_\alpha^{wf}) \quad \text{for successor ordinal } \alpha \\
\mathrm{T}_\alpha^{wf} & = & \bigcup_{\beta<\alpha} \mathrm{T}_\beta^{wf} & \mathrm{U}_\alpha^{wf} & = & \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{T}_\alpha^{wf}) \quad \text{for limit ordinal } \alpha \neq 0 \\
\mathrm{T}_\infty^{wf} & = & \bigcup_{\alpha} \mathrm{T}_\alpha^{wf} & \mathrm{U}_\infty^{wf} & = & \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{T}_\infty^{wf})
\end{array}$$

Lemma 11.2.4 Let α range over all ordinals, then T_α^{wf}, T_∞^{wf}, U_α^{wf}, and U_∞^{wf} are all in normal form.

Proof.

First we show by transfinite induction that T_α^{wf} is in normal form for any ordinal α. The case is trivial for limit ordinal 0. If α is a successor ordinal, then $\mathrm{T}_\alpha^{wf} = \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{U}_{\alpha-1}^{wf}) = \mathrm{lfp}(\mathbf{C}_{\mathrm{P}^{wf},\mathrm{U}_{\alpha-1}^{wf}})$. It follows that T_α^{wf} is in normal form, by Lemma 11.2.1. Now suppose $\alpha \neq 0$ is a limit ordinal. Then $\mathrm{T}_\alpha^{wf} = \bigcup_{\beta<\alpha} \mathrm{T}_\beta^{wf}$. By Definition 11.2.2, we need to show that for all $\mathsf{o}, \mathsf{m}, \mathsf{v}$: $mvd(\mathsf{o}, \mathsf{m}, \mathsf{v}) \in \mathrm{T}_\alpha^{wf}$ *iff* $locmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}) \in \mathrm{T}_\alpha^{wf}$, or there is c such that $valinhmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathrm{T}_\alpha^{wf}$ or $codinhmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathrm{T}_\alpha^{wf}$.

(1) $\Rightarrow$

If $mvd(\mathsf{o}, \mathsf{m}, \mathsf{v}) \in \mathrm{T}_\alpha^{wf}$, then there exists $\beta < \alpha$ such that $mvd(\mathsf{o}, \mathsf{m}, \mathsf{v}) \in \mathrm{T}_\beta^{wf}$. T_β^{wf} is in normal form by the induction hypothesis. Thus $locmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}) \in \mathrm{T}_\beta^{wf} \subseteq \mathrm{T}_\alpha^{wf}$, or there is c such that $valinhmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathrm{T}_\beta^{wf} \subseteq \mathrm{T}_\alpha^{wf}$ or $codinhmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathrm{T}_\beta^{wf} \subseteq \mathrm{T}_\alpha^{wf}$.

(2) $\Leftarrow$

If $locmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}) \in \mathrm{T}_\alpha^{wf}$, then there exists $\beta < \alpha$ such that $locmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}) \in \mathrm{T}_\beta^{wf}$. It follows that $mvd(\mathsf{o}, \mathsf{m}, \mathsf{v}) \in \mathrm{T}_\beta^{wf} \subseteq \mathrm{T}_\alpha^{wf}$, since T_β^{wf} is in normal form by the induction hypothesis. On the other hand, if there exists exists c such that $valinhmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathrm{T}_\alpha^{wf}$ or $valinhmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathrm{T}_\alpha^{wf}$, then there exists $\gamma < \alpha$ such that $valinhmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathrm{T}_\gamma^{wf}$ or $codinhmvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathrm{T}_\gamma^{wf}$. It follows that $mvd(\mathsf{o}, \mathsf{m}, \mathsf{v}, \mathsf{c}) \in \mathrm{T}_\gamma^{wf} \subseteq \mathrm{T}_\alpha^{wf}$, since T_γ^{wf} is in normal form by the induction hypothesis.

Similarly, we can also prove that T_∞^{wf} is in normal form. Moreover, for any ordinal α, $\mathrm{U}_\alpha^{wf} = \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{T}_\alpha^{wf}) = \mathrm{lfp}(\mathbf{C}_{\mathrm{P}^{wf},\mathrm{T}_\alpha^{wf}})$. It follows that U_α^{wf} is in normal form, by Lemma 11.2.1. Similarly, we can also show U_∞^{wf} is in normal form.

□

Theorem 11.2.5 Given the well-founded rewriting P^{wf} of an F-logic program P, the well-founded model of P^{wf} is isomorphic to the optimistic object model of P.

Proof.

Let $\mathcal{M}^{wf} = \langle\, \mathrm{T}^{wf}; \mathrm{U}^{wf} \,\rangle$ be the well-founded model of P^{wf}. Then by Proposition 4.2.4, $\mathrm{T}^{wf} = \mathrm{T}^{wf}_{\infty}$ and $\mathrm{U}^{wf} = \mathrm{U}^{wf}_{\infty} - \mathrm{T}^{wf}_{\infty}$. Let $\mathcal{M} = \langle\, \mathrm{T}; \mathrm{U} \,\rangle$ be the optimistic object model of P. Then by Lemma 9.1.2, $\mathrm{T} = \pi(\widehat{\mathrm{T}}_{\infty})$ and $\mathrm{U} = \pi(\widehat{\mathrm{U}}_{\infty} - \widehat{\mathrm{T}}_{\infty})$. Therefore, by Definition 11.2.1, to show that $\mathcal{M}^{wf}$ is isomorphic to $\mathcal{M}$, it suffices to show that T^{wf}_{∞} is isomorphic to $\widehat{\mathrm{T}}_{\infty}$ and U^{wf}_{∞} is isomorphic to $\widehat{\mathrm{U}}_{\infty}$.

First note that T^{wf}_{α} and U^{wf}_{α} are in normal form for any ordinal α, by Lemma 11.2.4. Now we will prove by transfinite induction that T^{wf}_{α} is isomorphic to $\widehat{\mathrm{T}}_{\alpha}$ and U^{wf}_{α} is isomorphic to $\widehat{\mathrm{U}}_{\alpha}$, for any ordinal α.

(1) If $\alpha = 0$, then T^{wf}_{0} is isomorphic to $\widehat{\mathrm{T}}_{0}$, since $\mathrm{T}^{wf}_{0} = \emptyset$ and $\widehat{\mathrm{T}}_{0} = \emptyset$. $\mathrm{U}^{wf}_{0} = \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{T}^{wf}_{0}) = \mathrm{lfp}(\mathbf{C}_{\mathrm{P}^{wf},\mathrm{T}^{wf}_{0}})$, by Definitions 11.2.3 and 4.2.6, and $\widehat{\mathrm{U}}_{0} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{0}) = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{T}}_{0}})$, by Definitions 9.1.2 and 7.2.7. It follows that U^{wf}_{0} is isomorphic to $\widehat{\mathrm{U}}_{0}$, by Proposition 11.2.3.

(2) If α is a successor ordinal, then $\mathrm{T}^{wf}_{\alpha} = \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{U}^{wf}_{\alpha-1}) = \mathrm{lfp}(\mathbf{C}_{\mathrm{P}^{wf},\mathrm{U}^{wf}_{\alpha-1}})$, by Definitions 11.2.3 and 4.2.6, and $\widehat{\mathrm{T}}_{\alpha} = \Psi_{\mathrm{P}}(\widehat{\mathrm{U}}_{\alpha-1}) = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{U}}_{\alpha-1}})$, by Definitions 9.1.2 and 7.2.7. Moreover, $\mathrm{U}^{wf}_{\alpha-1}$ is isomorphic to $\widehat{\mathrm{U}}_{\alpha-1}$ by the induction hypothesis. It follows that U^{wf}_{α} is isomorphic to $\widehat{\mathrm{U}}_{\alpha}$, by Proposition 11.2.3. Similarly to (1), we can also show that U^{wf}_{α} is isomorphic to $\widehat{\mathrm{U}}_{\alpha}$.

(3) If $\alpha \neq 0$ is a limit ordinal, then $\mathrm{T}^{wf}_{\alpha} = \bigcup_{\beta<\alpha} \mathrm{T}^{wf}_{\beta}$ and $\widehat{\mathrm{T}}_{\alpha} = \bigcup_{\beta<\alpha} \widehat{\mathrm{T}}_{\beta}$. Clearly, T^{wf}_{α} is isomorphic to $\widehat{\mathrm{T}}_{\alpha}$, because T^{wf}_{β} is isomorphic to $\widehat{\mathrm{T}}_{\beta}$ for all $\beta < \alpha$, by the induction hypothesis. Similarly to (1), we can also show that U^{wf}_{α} is isomorphic to $\widehat{\mathrm{U}}_{\alpha}$.

Note that $\mathrm{T}^{wf}_{\infty} = \bigcup_{\alpha} \mathrm{T}^{wf}_{\alpha}$ and $\widehat{\mathrm{T}}_{\infty} = \bigcup_{\alpha} \widehat{\mathrm{T}}_{\alpha}$. Therefore, it follows that T^{wf}_{∞} is isomorphic to $\widehat{\mathrm{T}}_{\infty}$, because T^{wf}_{α} is isomorphic to $\widehat{\mathrm{T}}_{\alpha}$, for any ordinal α. Moreover, $\mathrm{U}^{wf}_{\infty} = \mathbf{S}_{\mathrm{P}^{wf}}(\mathrm{T}^{wf}_{\infty}) = \mathrm{lfp}(\mathbf{C}_{\mathrm{P}^{wf},\mathrm{T}^{wf}_{\infty}})$, by Definitions 11.2.3 and 4.2.6, and $\widehat{\mathrm{U}}_{\infty} = \Psi_{\mathrm{P}}(\widehat{\mathrm{T}}_{\infty}) = \mathrm{lfp}(\mathbf{T}_{\mathrm{P},\widehat{\mathrm{T}}_{\infty}})$, by Definitions 9.1.2 and 7.2.7. Thus U^{wf}_{∞} is isomorphic to $\widehat{\mathrm{U}}_{\infty}$, by Proposition 11.2.3.

□

11.3 Data Complexity

In general, the optimistic object model of an F-logic program is not necessarily recursively enumerable. However, for function-free F-logic programs, the Herbrand universe is finite and thus the optimistic object model can be effectively constructed. In this section we discuss data complexity for such programs.

In this dissertation we consider only a subset of F-logic, which contains three kinds of atoms: $\mathsf{o} : \mathsf{c}$, $\mathsf{s} :: \mathsf{c}$, and $\mathsf{s}[\mathsf{m} \twoheadrightarrow \mathsf{v}]$. Any ground atomic query must have one of these three forms, where $\mathsf{o}, \mathsf{c}, \mathsf{s}, \mathsf{m}, \mathsf{v}$ are constants.

As for Datalog programs, we can divide any F-logic program, P, into two disjoint parts: an *intentional* database (IDB), $\mathrm{P_R}$, which consists of all rules in P and no facts, and an *extensional* database (EDB), $\mathrm{P_F}$, which contains only the facts in P. We can think of $\mathrm{P_R}$ as a function that maps any EDB, $\mathrm{P_F}$, to the optimistic object model of the combined F-logic program $\mathrm{P_R} \cup \mathrm{P_F}$. Following [57], we have the following definition of data complexity.

Definition 11.3.1 (Data Complexity) Given an IDB $\mathrm{P_R}$ and an EDB $\mathrm{P_F}$, the *data complexity* of $\mathrm{P_R}$ is defined as the computational complexity of deciding the truth value of any ground atomic query in the optimistic object model of $\mathrm{P_R} \cup \mathrm{P_F}$, as a function of the size of $\mathrm{P_F}$.

Given an F-logic program $\mathrm{P} = \mathrm{P_R} \cup \mathrm{P_F}$ and its well-founded rewriting P^{wf}, let $\mathrm{P}_{\mathrm{R}}^{wf}$ be the IDB of P^{wf}, and $\mathrm{P}_{\mathrm{F}}^{wf}$ be the EDB of P^{wf}. By Definitions 11.1.1 and 11.1.2, $\mathrm{P}_{\mathrm{R}}^{wf}$ consists of the trailer rules shown in Figure 12 plus the rewritings of all rules in $\mathrm{P_R}$. The EDB $\mathrm{P}_{\mathrm{F}}^{wf}$ consists of the rewritings of all facts in $\mathrm{P_F}$. Because the rewriting of an F-logic rule is linear and the size of the trailer is a constant, the size of $\mathrm{P}_{\mathrm{R}}^{wf}$ is linear in the size of $\mathrm{P_R}$ and the size of $\mathrm{P}_{\mathrm{F}}^{wf}$ is also linear in the size of $\mathrm{P_F}$.

By Theorem 11.2.5, the well-founded model of P^{wf} is isomorphic to the optimistic object model of P. Therefore, the data complexity of the optimistic object model semantics reduces to the data complexity of the well-founded semantics.

Because the rewriting does not introduce new function symbols, the rewriting of a function-free F-logic program is a function-free Datalog program. Since data complexity of the well-founded semantics for function-free programs is polynomial time [18], we have the following corollary.

Corollary 11.3.1 The data complexity of the optimistic object model semantics for function-free F-logic programs is polynomial time.

Chapter 12

Conclusion and Future Work

We have developed a comprehensive model theory for nonmonotonic multiple value and code inheritance for general, unrestricted object-oriented knowledge bases. Our new inheritance semantics supports implicit inference by inheritance as well as explicit deduction via rules. Inference by inheritance supports a multitude of features, such as overriding, nonmonotonic multiple value and code inheritance, meta programming, and dynamic class hierarchies — the important features that are fundamental to advanced object-oriented knowledge management.

12.1 Contributions

Here we summarize our contributions in this work:

(1) In the setting of three-valued models, we formalize the notions of locality, context, and inheritance candidacy, and formally define the inheritance postulates that capture the common intuition behind overriding and conflict resolution in nonmonotonic multiple value and code inheritance. These postulates specify the minimum requirements for an object model of a program.

(2) We specify an extended alternating fixpoint procedure which can be used to compute object models for F-logic programs.

(3) We define stable object models which satisfy a certain computational property of the alternating fixpoint procedure that we introduce. Moreover, we formally prove that stable object models satisfy all the inheritance postulates.

(4) We define a unique object model, called optimistic object model, for any given F-logic program. We prove three different characterizations of the optimistic object model semantics: the optimistic object model is the least fixpoint of

the extended alternating fixpoint computation; it is the least stable object models with respect to information ordering; and it is a minimal object model with respect to truth ordering.

(5) We propose a linear-time rewriting algorithm which translates F-logic programs to a certain kind of general logic programs, and formally prove the isomorphism between the well-founded model of the rewritten program and the optimistic object model of the original F-logic program.

(6) Our new inheritance semantics has been implemented in the Flora-2 system [59], which incorporates F-logic, HiLog, and Transaction Logic into a single, coherent logic language.

To the best of our knowledge, the optimistic object model semantics is currently the only model-theoretic semantics for nonmonotonic multiple value and code inheritance that applies to general, unrestricted object-oriented knowledge bases.

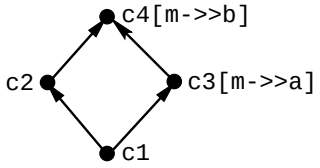
12.2 Future Work

Our formalization of value and code inheritance defines the concepts of locality, context, inheritance candidacy, and introduces the inheritance postulates. These notions have implications beyond the semantics and implementation of inheritance. In particular, this sets the foundation for a framework in which various inheritance policies can be defined *programmatically*.

The overriding semantics in our new model theory can be termed as most-specific-definition-based overriding [25], which is commonly used in AI systems. However, we do not claim that this is the only useful inheritance semantics in object-oriented knowledge bases. It has been argued that no single inheritance policy can suit all needs. Indeed, some advanced applications require a variety of overriding and inheritance semantics. For instance, path-based overriding [25] is widely used in the research on discretionary access control; inflating inheritance [28] and null inheritance [28] are used for multilevel security in databases [27]. The difference between most-specific-definition-based and path-based overriding can be seen in the following example.

Example 12.2.1 Consider the program in Figure 13 which contains facts only. According to the optimistic object model semantics, $\mathsf{c}_4[\mathsf{m}]$ is not a value inheritance candidate for c_1, because it is overridden by a more *specific* inheritance context, $\mathsf{c}_3[\mathsf{m}]$, of c_1. So we can derive $\mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{a}]$ but not $\mathsf{c}_1[\mathsf{m} \twoheadrightarrow \mathsf{b}]$. We can see that most-specific-definition-based overriding only takes into account class memberships and subclass relationships. On the other hand, path-based overriding factors in topologies of class hierarchies. In the program, the class c_1 has two *distinct* inheritance paths to c_4:

$c_1 \rightarrow c_2 \rightarrow c_4$ and $c_1 \rightarrow c_3 \rightarrow c_4$. Although $c_4[m]$ is overridden by $c_3[m]$ along the path $c_1 \rightarrow c_3 \rightarrow c_4$, it is not overridden along the path $c_1 \rightarrow c_2 \rightarrow c_4$. Therefore, according to the path-based semantics, we could derive $c_1[m \twoheadrightarrow b]$ and $c_1[m \twoheadrightarrow a]$ by inheritance along the two paths, $c_1 \rightarrow c_2 \rightarrow c_4$ and $c_1 \rightarrow c_3 \rightarrow c_4$, respectively. □



$c_1 : c_2$.
$c_1 : c_3$.
$c_2 :: c_4$.
$c_3 :: c_4$.
$c_3[m \twoheadrightarrow a]$.
$c_4[m \twoheadrightarrow b]$.

Figure 13: Most-Specific-Definition-Based and Path-Based Overriding

Example 12.2.2 We use the program in Figure 13 to illustrate the idea of programmable inheritance. As we have explained in Example 12.2.1, under the most-specific-definition-based overriding semantics, this program yields $c_1[m \twoheadrightarrow a]$ but not $c_1[m \twoheadrightarrow b]$. In contrast, we can derive both using path-based overriding.

How can one specify path-based inheritance for the class c_1 programmatically? Generally, we can think of path-based inheritance as inheritance directly via immediate superclasses of a given object. For instance, in Figure 13, c_2 and c_3 are both immediate superclasses of c_1, whereas c_4 is not.

The first step is to introduce new syntax[1] to represent immediate class memberships. To this end, we introduce new atoms of the form, $o \,||\, c$, to denote that c is an immediate superclass of o (conversely, o is an immediate member of c). Second, we need to distinguish rules intended for customizing inheritance semantics from rules for specifying class and object methods. The system will use the inheritance rules only when the need for customized inheritance arises. For this purpose, we introduce a special keyword, inh, to mark the rules for programmable inheritance.

With these new language constructs, we can program path-based inheritance for the class c_1 as follows.

$$\mathsf{inh} \quad c_1[m \twoheadrightarrow V] \longleftarrow c_1 \,||\, X,\ X[m \twoheadrightarrow V].$$

Given the program in Figure 13 and provided that the system recognizes this rule to compute inheritance for c_1, we can derive both $c_1[m \twoheadrightarrow a]$ and $c_1[m \twoheadrightarrow b]$ (assuming that c_2 also inherits $m \twoheadrightarrow b$ from c_4). □

[1]The original F-logic and the current Flora-2 query language do not have syntax constructs to represent immediate class memberships.

Some papers in the literature discuss ideas on how to customize inheritance semantics. However, most of them propose only *ad hoc* syntax and do not attempt to develop a general framework for expressing different inheritance semantics.

The work of Dobbie and Topor [14] restricts the syntax of queries to designate the source of inheritance, so that at compile time multiple inheritance conflicts can be detected by checking certain syntactic conditions. This is similar to the approach taken in C^{++}. However, the special syntax only aims at resolving multiple inheritance conflicts, but not at defining different inheritance semantics.

Jamil and Lakshmanan [29] introduce meta syntax in their query language to express *withdrawal* of inheritance, which enables declaring inheritance to be prohibited from a superclass or by a subclass. Although the scenario considered in [29] is more general than [14], their work is still primarily concerned with resolving multiple inheritance conflicts.

In [28], Jamil develops a proof theory for the so called *parametric inheritance*. In his framework, rules are marked with two parameters representing different inheritance types (such as override and inflate) and different inheritance modes (such as value, code, and null), respectively, so that users can parameterize — but not program — propagation of inheritance. However, parametric inheritance is still an *ad hoc*, complicated, and insufficiently general mechanism for specifying a wide variety of inheritance semantics. For example, it does not support withdrawal of inheritance as proposed in [29]. Because the way inheritance is controlled is tightly coupled with the core query language syntax, the proof theory can not account for nonmonotonic multiple inheritance. Moreover, it is not clear how to change the inheritance semantics for a class without even affecting its superclasses.

Jajodia et al. [25] propose a logic-based language that offers flexibility in specifying various access control policies in database security. It utilizes different inheritance semantics as convenient ways of propagating authorizations among subjects and objects that are organized in class hierarchies. However, their language is crafted specifically for access control. It is too general on one hand and insufficient on the other. In particular, it does not support many important features of object-oriented languages, such as value and code inheritance. More importantly, no formal semantics for inheritance is attached to their proposed framework.

As we have seen, although the importance of the issue of programmability in customizing inheritance semantics has been recognized, it has not been very well studied in the literature. It is not enough to just extend inheritance semantics by enumerating a limited number of scenarios like [28] or [25], since the semantics suitable for different applications can be very diverse. Therefore, it is desirable to develop a general mechanism with which users can program the desired effects of inheritance.

Bibliography

[1] S. Abiteboul, G. Lausen, H. Uphoff, and E. Waller. Methods and rules. In *ACM SIGMOD Conference on Management of Data*, pages 32–41, 1993.

[2] F. Afrati, I. Karali, and T. Mitakos. Inheritance in object oriented Datalog: A modular logic programming approach. Technical report, National Technical University of Athens, 1997.

[3] E. Bertino, S. Jajodia, and P. Samarati. A flexible authorization mechanism for relational data management systems. *ACM Transactions on Information Systems*, 17(2):101–140, April 1999.

[4] A. J. Bonner and M. Kifer. Transaction logic programming. In *International Conference on Logic Programming*, pages 257–282, 1993.

[5] A. J. Bonner and M. Kifer. A logic for programming database transactions. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, chapter 5, pages 117–166. Kluwer Academic Publishers, March 1998.

[6] M. Bugliesi and H. M. Jamil. A logic for encapsulation in object oriented languages. In *Proceedings of the 6th International Symposium on Programming Language Implementation and Logic Programming*, pages 215–229, 1994.

[7] M. Bugliesi and H. M. Jamil. A stable model semantics for behavioral inheritance in deductive object oriented languages. In *International Conference on Database Theory*, pages 222–237, 1995.

[8] W. Chen, M. Kifer, and D. S. Warren. HiLog: A foundation for higher-order logic programming. *Journal of Logic Programming*, 15(3):187–230, February 1993.

[9] W. Chen and D. S. Warren. Tabled evaluation with delaying for general logic programs. *Journal of ACM*, 43(1):20–74, 1996.

[10] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Securing XML documents. In *International Conference on Extending Database Technology*, pages 121–135, 2000.

[11] H. Davulcu, G. Yang, M. Kifer, and I.V. Ramakrishnan. Design and implementation of the physical layer in webbases: The XRover experience. In *Sixth International Conference on Rules and Objects in Databases (DOOD'2000)*, London, United Kingdom, July 2000.

[12] S. Decker, D. Brickley, J. Saarela, and J. Angele. A query and inference service for RDF. In *QL'98 – The Query Languages Workshop*, December 1998.

[13] S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In R. Meersman et al., editor, *Database Semantics, Semantic Issues in Multimedia Systems*, pages 351–369. Kluwer Academic Publisher, Boston, 1999.

[14] G. Dobbie and R. Topor. Resolving ambiguities caused by multiple inheritance. In *International Conference on Deductive and Object-Oriented Databases*, pages 265–280, 1995.

[15] A. Farquhar, R. Fikes, W. Pratt, and J. Rice. Collaborative ontology construction for information integration. Technical Report KSL-95-63, Knowledge Systems Laboratory, Stanford University, August 1995.

[16] D. Fensel, S. Decker, M. Erdmann, and R. Studer. Ontobroker: Or how to enable intelligent access to the WWW. In *Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada, 1998.

[17] A. Van Gelder. The alternating fixpoint of logic programs with negation. In *ACM Symposium on Principles of Database Systems*, pages 1–10, 1989.

[18] A. Van Gelder, K. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of ACM*, 38(3):620–650, July 1991.

[19] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R. A. Kowalski and K. Bowen, editors, *Proceedings of the Fifth International Conference on Logic Programming*, pages 1070–1080. The MIT Press, 1988.

[20] C. H. Goh, S. Bressan, S. E. Madnick, and M. D. Siegel. Context interchange: Representing and reasoning about data semantics in heterogeneous systems. Technical report, MIT, School of Management, 1996.

[21] C. H. Goh, S. Bressan, S. E. Madnick, and M. D. Siegel. Context mediation: New features and formalisms for the intelligent integration of information. *ACM Transactions on Information Systems*, 1999.

[22] R. V. Guha, O. Lassila, E. Miler, and D. Brickley. Enabling inferencing. In *QL'98 – The Query Languages Workshop*, December 1998.

[23] A. Gupta, B. Ludäscher, and M. E. Martone. Knowledge-based integration of neuroscience data sources. In *12th International Conference on Scientific and Statistical Database Management (SSDBM)*, Berlin, Germany, July 2000. IEEE.

[24] G.-J. Houben. HERA: Automatically generating hypermedia front-ends for ad hoc data from heterogeneous and legacy information systems. In *Engineering Federated Information Systems*, pages 81–88. Aka and IOS Press, 2000.

[25] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. *ACM Transactions on Database Systems*, 26(2):214–260, June 2001.

[26] H. M. Jamil. Implementing abstract objects with inheritance in Datalog^{neg}. In *International Conference on Very Large Data Bases*, pages 56–65, 1997.

[27] H. M. Jamil. Belief reasoing in MLS deductive databases. In *ACM SIGMOD Conference on Management of Data*, pages 109–120, 1999.

[28] H. M. Jamil. A logic-based language for parametric inheritance. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, *KR2000: Principles of Knowledge Representation and Reasoning*, pages 611–622, San Francisco, 2000. Morgan Kaufmann.

[29] H. M. Jamil and L. V. S. Lakshmanan. A declarative semantics for behavioral inheritance and conflict resolution. In *International Logic Programming Symposium*, pages 130–144, 1995.

[30] M. Kifer. Deductive and object-oriented data languages: A quest for integration. In *International Conference on Deductive and Object-Oriented Databases*, volume 1013 of *Lecture Notes in Computer Science*, pages 187–212, Singapore, December 1995. Springer-Verlag. Keynote address at the 3rd International Conference on Deductive and Object-Oriented Databases.

[31] M. Kifer and G. Lausen. F-Logic: A higher-order language for reasoning about objects, inheritance and schema. In *ACM SIGMOD Conference on Management of Data*, pages 134–146, New York, 1989. ACM.

[32] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of ACM*, 42:741–843, July 1995.

[33] W. Kim and F. H. Lochovsky, editors. *Object-Oriented Concepts, Databases, and Applications*. ACM Press and Addison-Wesley, 1989.

[34] E. Laenens, D. Saccà, and D. Vermeir. Extending logic programming. In *ACM SIGMOD Conference on Management of Data*, pages 184–193, 1990.

[35] E. Laenens and D. Vermeir. A fixpoint semantics for ordered logic. *Journal of Logic and Computation*, 1(2):159–185, 1990.

[36] L. V. S. Lakshmanan and K. Thirunarayan. Declarative frameworks for inheritance. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 357–388. Kluwer Academic Publishers, 1998.

[37] J. W. Lloyd. *Foundations of Logic Programming*. Springer Verlag, 1984.

[38] Y. Lou and Z. M. Ozsoyoglu. LLO: An object-oriented deductive language with methods and method inheritance. In *ACM SIGMOD Conference on Management of Data*, pages 198–207, 1991.

[39] B. Ludäscher. The FLIP system (F-logic to XSB-Prolog compiler). `http://www.informatik.uni-freiburg.de/ ludaesch/flip/`, 1994.

[40] B. Ludäscher, R. Himmeröder, G. Lausen, W. May, and C. Schlepphorst. Managing semistructured data with FLORID: A deductive object-oriented perspective. *Information Systems*, 23(8):589–613, 1998.

[41] A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2), March/April 2001.

[42] W. May and P. Kandzia. Nonmonotonic inheritance in object-oriented deductive database languages. *Journal of Logic and Computation*, 11(4), 2001.

[43] W. May, B. Ludäscher, and G. Lausen. Well-founded semantics for deductive object-oriented database languages. In *International Conference on Deductive and Object-Oriented Databases*, pages 320–336. Springer Verlag LNCS, 1997.

[44] T. C. Przymusinski. Every logic program has a natural stratification and an iterated least fixed point model. In *ACM Symposium on Principles of Database Systems*, pages 11–21, 1989.

[45] T. C. Przymusinski. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae*, 13(4):445–464, 1990.

[46] F. Rabitti, E. Bertino, W. Kim, and D. Woelk. A model of authorization for next-generation database systems. *ACM Transactions on Database Systems*, 16(1):88–131, March 1991.

[47] R. Reiter. On closed world databases. In H. Gallaire and J. Minker, editors, *Logic and Databases*, pages 55–76. Plenum Press, New York, 1978.

[48] R. Reiter. A logic for default reasoning. 13(1–2):81–132, 1980.

[49] K. Sagonas, T. Swift, and D. S. Warren. XSB as an efficient deductive database engine. In *ACM SIGMOD Conference on Management of Data*, pages 442–453, New York, May 1994. ACM.

[50] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.

[51] M. Sintek and S. Decker. TRIPLE – An RDF query, inference, and transformation language. In *Deductive Databases and Knowledge Management*, October 2001.

[52] D. L. Spooner. The impact of inheritance on security in object-oriented database systems. In *Database Security II: Status and Prospects*, pages 141–150, 1988.

[53] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure. AI for the Web — Ontology-based community web portals. In *9-th International World Wide Web Conference (WWW9)*, Amsterdam, The Netherlands, May 2000.

[54] Y. Sure, S. Staab, and J. Angele. OntoEdit: Guiding ontology development by methodology and inferencing. In *First International Conference on Ontologies, Databases, and Applications of Semantics*, Irvine, California, October 2002.

[55] D. S. Touretzky. *The Mathematics of Inheritance*. Morgan-Kaufmann, Los Altos, CA, 1986.

[56] J. D. Ullman. A comparison between deductive and object-oriented database systems. In *International Conference on Deductive and Object-Oriented Databases*, pages 263–277. 1991.

[57] M. Vardi. The complexity of relational query languages. In *ACM Symposium on Theory of Computing*, pages 137–145, 1982.

[58] G. Yang and M. Kifer. FLORA: Implementing an efficient dood system using a tabling logic engine. In *Sixth International Conference on Rules and Objects in Databases (DOOD'2000)*, London, United Kingdom, July 2000.

[59] G. Yang and M. Kifer. Flora-2: User's manual. Technical report, Computer Science Department, SUNY at Stony Brook, June 2002. `http://flora.sourceforge.net/`.

[60] G. Yang and M. Kifer. Well-founded optimism: Inheritance in frame-based knowledge bases. In *First International Conference on Ontologies, Databases, and Applications of Semantics*, Irvine, California, October 2002.