Answer Set Programming Modulo Theories

by

Michael Bartholomew

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2016 by the
Graduate Supervisory Committee:

Joohyung Lee, Chair
Rida Bazzi
Charles Colbourn
Georgios Fainekos
Vladimir Lifschitz

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

Knowledge representation and reasoning is a prominent subject of study within the field of artificial intelligence that is concerned with the symbolic representation of knowledge in such a way to facilitate automated reasoning about this knowledge. Often in real-world domains, it is necessary to perform defeasible reasoning when representing default behaviors of systems. Answer Set Programming is a widely-used knowledge representation framework that is well-suited for such reasoning tasks and has been successfully applied to practical domains due to efficient computation through grounding–a process that replaces variables with variable-free terms–and propositional solvers similar to SAT solvers. However, some domains provide a challenge for grounding-based methods such as domains requiring reasoning about continuous time or resources.

To address these domains, there have been several proposals to achieve efficiency through loose integrations with efficient declarative solvers such as constraint solvers or satisfiability modulo theories solvers. While these approaches successfully avoid substantial grounding, due to the loose integration, they are not suitable for performing defeasible reasoning on functions. As a result, this expressive reasoning on functions must either be performed using predicates to simulate the functions or in a way that is not elaboration tolerant. Neither compromise is reasonable; the former suffers from the grounding bottleneck when domains are large as is often the case in real-world domains while the latter necessitates encodings to be non-trivially modified for elaborations.

This dissertation presents a novel framework called Answer Set Programming Modulo Theories (ASPMT) that is a tight integration of the stable model semantics and

satisfiability modulo theories. This framework both supports defeasible reasoning about functions and alleviates the grounding bottleneck. Combining the strengths of Answer Set Programming and satisfiability modulo theories enables efficient continuous reasoning while still supporting rich reasoning features such as reasoning about defaults and reasoning in domains with incomplete knowledge. This framework is realized in two prototype implementations called MVSM and ASPMT2SMT, and the latter was recently incorporated into a non-monotonic spatial reasoning system. To define the semantics of this framework, we extend the first-order stable model semantics by Ferraris, Lee and Lifschitz to allow "intensional functions" and provide analyses of the theoretical properties of this new formalism and on the relationships between this and existing approaches.

Dedicated to my loving parents, who instilled in me a life-long sense of curiosity.

ACKNOWLEDGEMENTS

First and foremost, I want to thank my adviser, Joohyung Lee. Joohyung rekindled my nearly-extinguished interest in Computer Science when I was an undergraduate. He patiently mentored me as an undergraduate researcher and when I approached him about my plans for after my undergraduate studies, he invited me to continue research with him as a fully-supported Ph.D. student. Joohyung worked closely with me throughout my time at ASU, encouraging me to work in collaboration with other groups and suggesting courses of interest related to our reseach. I'm also grateful for being given the opportunity to attend and present at conferences in interesting places like Toronto, Rome, and Istanbul. I'm fortunate to have had such a generous, patient, and dedicated adviser. Thanks, Joohyung.

I would also like to thank the members of Joohyung's research lab. Ravi Palla, Yunsong Meng, and Michael Casolary were all seasoned members of Joohyung's lab when I joined the group and provided plenty of useful comments and mentorship. Possibly just as important, they also provided countless welcome distractions whether through celebratory lab dinners or casual conversations in the lab. I also want to thank the members of the lab that joined after I did–Sunjin Kim, Yu Zhang, Chao Zheng, Yi Wang, Joseph Babb, and Mohammad Hekmatnejad. We discussed research, exchanged suggestions for courses, and had our fun outside the lab as well. It was a pleasure working with all of these bright minds. Thank you all.

Next, I would like to thank my close friend and college roommate Devon O'Brien. We were either friendly rivals or project teammates in 10 courses at ASU and he even taught me how to drive. Devon has always been incredibly insightful and made me challenge my own world views. Devon provided much-needed words of encouragement when I was on the brink of abandoning my Ph.D. Then, in one final show of support at the end of this road, Devon flew in from out of town to attend my final Ph.D.

defense. I'm sure I couldn't have finished this journey without your encouragement and support. Thanks, Devon.

Finally, I want to thank my parents. Throughout my life, my parents have been loving and encouraging. They somehow found a way to pique a young child's curiosity and maintain this as I grew up without it ever feeling like they were pushing me. When my mother introduced me to computer programming at age 10, I was completely uninterested and she wisely backed off. Who could have guessed I would later undergo such an extreme reversal to arrive at this point? My parents were more than happy to have me move back in with them for what turned out to be my most productive two years of research. I love you both so much. Thanks, Mom and Dad.

My dissertation work was partially supported by the NSF under Grants IIS-1319794 and IIS-0916116, by the Office of the Director of National Intelligence(ODNI), Intelligence Advanced Research Projects Activities (IARPA), by the South Korea IT R&D program MKE/KIAT 2010-TD-300404-001, and by the ASU Fulton Schools of Engineering Dean's Fellowship.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

Figure | Page

Chapter 1

INTRODUCTION

Knowledge Representation and Reasoning is an area of study within the field of Artificial Intelligence that is concerned with the symbolic representation of knowledge in such a way to facilitate automated reasoning about this knowledge. Many approaches in this area are based on formal logics, often having limitations that make certain kinds of reasoning difficult or preclude efficient computation. In particular, we often want to efficiently reason about expressive functions.

While first-order logic (FOL) is well-known and decidable fragments of FOL are used in knowledge representation formalisms such as Description Logic and Boolean Satisfiability (SAT), it is unsuitable for certain kinds of reasoning. One such reasoning is defeasible reasoning such as reasoning about inertia or the default behavior of a system. This kind of reasoning is important for systems that do not always have complete information about the domain, but should still make decisions based on default assumptions of the world. However, such reasoning cannot be naturally expressed in FOL due to its monotonic nature. For instance, FOL is not well-structured for representing defaults such as that by default, a box will remain at its previous location. Due to this limitation, there has been extensive work in developing and studying non-monotonic formalisms.

One successful nonmonotonic formalism that has been successfully applied to a number of real world domains is the stable model semantics Gelfond and Lifschitz (1988). The Answer Set Programming (ASP) framework is based on this formalism and has successfully been applied to domains such as Automated Product Configuration Tiihonen *et al.* (2003), Space Shuttle Decision Support Balduccini *et al.* (2001), and Phylogenetic Tree Inference Brooks *et al.* (2007) [1] . The success of these applications of ASP is largely due to efficient grounders–tools that replace variables with ground terms–and efficient solvers, which are based on SAT solvers.

However, grounding-based methods suffer when the domain contains many values as is the case in many real-world settings requiring reasoning about continuous time or resources. For example, we consider a simple domain in which a tank of water has some current water level that remains the same by default but can increase at a constant rate when an input valve is open and decrease at a constant rate when on output valve is open. The reasoning task of determining the current water level of the tank requires reasoning about continuous time but any grounding based method will need to discretize this and to achieve reasonable approximations, the discretized domain must be quite large.

To address this grounding bottleneck, several formalisms have been proposed that avoid extensive grounding. This is achieved by loosely integrating Answer Set Programming with other declarative formalisms such as constraint processing Gebser *et al.* (2009a); Balduccini (2009), satisfiability modulo theories (SMT) Janhunen *et al.* (2011), and mixed integer programming Liu *et al.* (2012). While these approaches outperform standard ASP, due to the loose coupling, these approaches treat functions as in First-order logic so that they are unsuitable for defeasible reasoning. Further, except for Balduccini (2009), these approaches all consider integral domains, but are not able to perform continuous reasoning.

Approaches such as Cabalar (2011); Lifschitz (2012); Balduccini (2012) have incorporated so-called "intensional functions" into the stable model semantics in order

[1] This list comes from Lifschitz (2008).

to support defeasible reasoning about the value of functions (nonBoolean fluents). Intensional functions are intuitively functions that are defined in terms of other functions and predicates as opposed to predefined functions. For example, we can express the speed of a car as an intensional function that relies on the applied acceleration and previous speed of the car, whereas the function '+' is usually intended to be defined as arithmetic addition.

However, these frameworks are not focused on efficient computation and do not address the grounding bottleneck. In addition, the semantics described in Cabalar (2011); Balduccini (2012) are defined using a more complex notion of satisfaction than in the original stable model semantics while Lifschitz (2012) exhibits some behavior that is unexpected compared to typical extensions of the original stable model semantics.

We propose a novel framework–Answer Set Programming Modulo Theories (ASPMT)– that addresses some deficiencies in both groups of proposals. ASPMT is a tight integration of ASP and SMT that addresses the grounding bottleneck present in ASP and the restricted reasoning about functions present in SMT, resulting in a framework that is able to perform defeasible reasoning on continuous domains. To give the formal semantics of this framework, we introduce the functional stable model semantics (FSM), defined similarly to the first-order stable model semantics by Ferraris, Lee and Lifschitz Ferraris *et al.* (2011) but supporting the notion of intensional functions so that we attain the ability to support defeasible reasoning on these functions. By directly augmenting the first-order stable model semantics, we are able to extend some of the results established for the first-order stable model semantics. With these two newly introduced formalisms, we see a complete analogy between SAT, SMT, and FOL and the nonmonotonic counterparts ASP, ASPMT, and FSM.

Monotonic	Nonmonotonic
First-order Logic	Functional Stable Model Semantics
Satisfiability Modulo Theories	Answer Set Programming Modulo Theories
Propositional Satisfiability	Answer Set Programming

We provide two prototype implementations of the ASPMT system–MVSM and ASPMT2SMT. System MVSM computes the stable models of ASPMT theories by a reduction to ASP. While this approach still suffers the grounding bottleneck, it provides a more natural representation of functions while also adding basic typing to ASP. System ASPMT2SMT computes the stable models of ASPMT theories by a reduction to SMT. This approach partially addresses the grounding bottleneck and in doing so, is able to perform efficient defeasible reasoning about continuous time and resources. We formally compare the latter system to the state-of-the-art and show that this approach is a promising one.

This document is organized as follows. In Chapter 2, we present background material necessary to understand why the existing approaches are insufficient in performing efficient defeasible reasoning about continuous resources. In Chapter 3, we formally present the stable model semantics and the notion of intensional functions. In Chapter 4, we present the functional stable model (FSM) semantics and provide two alternate reformulations of this semantics. Chapter 5 discusses several properties of FSM that are of both theoretical and practical interest. In Chapter 6, we detail how to eliminate intensional predicates in favor of intensional functions and in Chapter 7, we detail how to eliminate intensional functions in favor of intensional predicates. In Chapter 8, we present the many-sorted generalization of FSM, a practical generalization that allows different functions to have different ranges and domains. Chapter 9

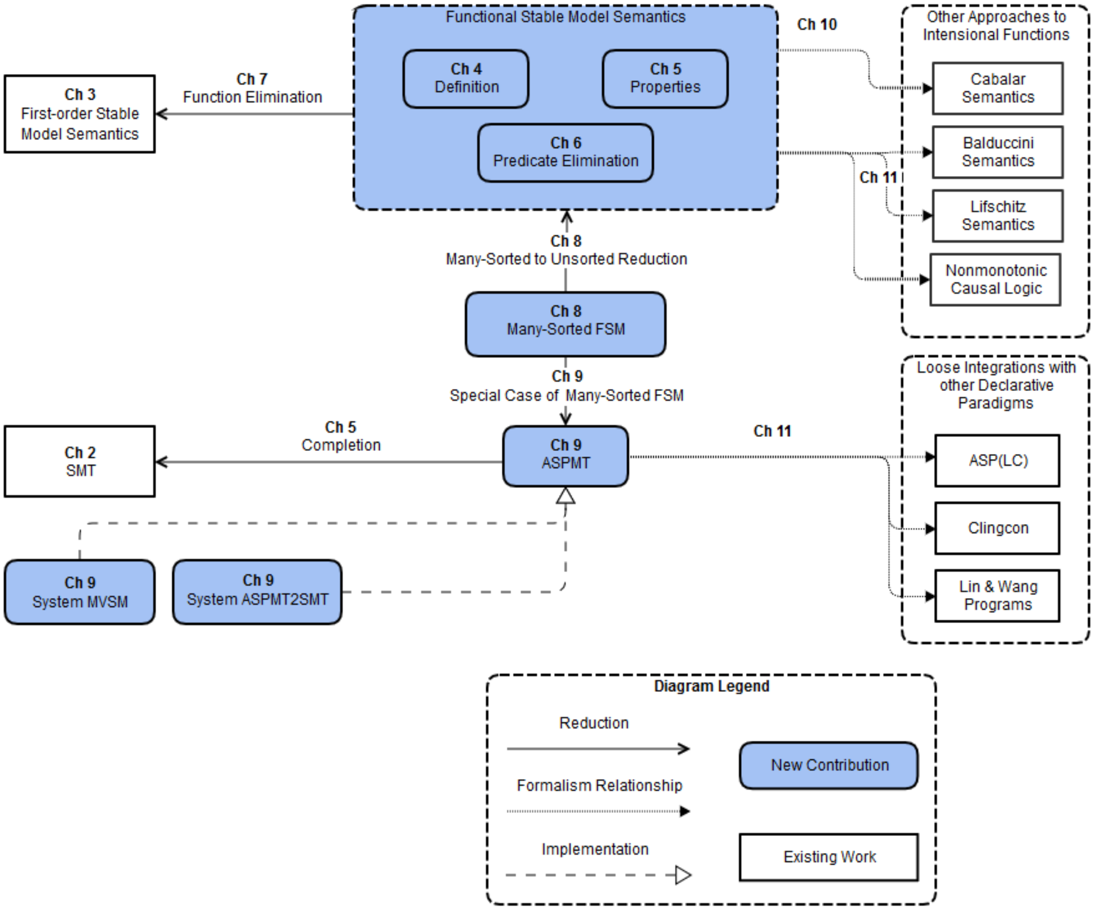

Figure 1.1: Dissertation Outline

describes the ASPMT framework and presents two prototype implementations of this framework. In Chapter 10, we discuss in detail the relationship between FSM and the Cabalar Semantics. Chapter 11 provides a comparison of FSM and ASPMT to several other related approaches. We then conclude in Chapter 12.

Chapter 2

BACKGROUND

2.1 Answer Set Programming

Answer Set Programming is a declarative programming paradigm especially well-suited for solving NP-hard combinatorial search problems. Among the applications that ASP has been successfully used are Automated Product Configuration Tiihonen *et al.* (2003), Space Shuttle Decision Support Balduccini *et al.* (2001), and Phylogenetic Tree Inference Brooks *et al.* (2007). The syntax of traditional ASP programs is similar to that of Prolog programs but the computation is based instead on the idea of grounding and search techniques similar to those used in SAT solvers. However, modern systems have augmented the language with rich features including aggregates, external predicates, and preferences. The efficient implementations of intelligent grounders and developments in SAT solvers have enabled the successful application of ASP to these domains.

The semantics of ASP is the stable model semantics originally defined in Gelfond and Lifschitz (1988) which is presented in terms of a notion called a *reduct* that will be detailed in Section 3.2. This semantics is non-monotonic which makes representation in this framework appealing because encoded domains are elaboration tolerant in sense of McCarthy (1998)–that is, it is convenient to modify a description to accommodate new behavior. It is possible to simply add the new domain specifications without amending existing formulas. This is particularly useful in defeasible reasoning, examples of which include reasoning about inertia and default behaviors of systems.

Example 1 *Consider the following running example from a Texas Action Group discussion, posted by Vladimir Lifschitz* [1] *.*

> *A car is on a road of length l. If the accelerator is activated, the car will speed up with constant acceleration a until the accelerator is released or the car reaches its maximum speed ms, whichever comes first. If the brake is activated, the car will slow down with acceleration $-a$ until the brake is released or the car stops, whichever comes first. Otherwise, the speed of the car remains constant. Give a formal representation of this domain, and write a program that uses your representation to generate a plan satisfying the following conditions: at duration 0, the car is at rest at one end of the road; at duration t, it should be at rest at the other end.*

We can represent the property that by default, the speed of the car will stay the same as in the previous timestep with the ASP rule [2]

$$\{speed(1, Y)\} \leftarrow speed(0, Y). \tag{2.1}$$

which intuitively reads "If the speed is Y at timestep 0, then by default, the speed is Y at time 1". The exceptions to this default behavior can then simply be added to the description

$$speed(1, Y) \leftarrow accel(0) \wedge speed(0, X) \wedge duration(0, D) \wedge (Y = X + a \times D).$$
$$speed(1, Y) \leftarrow decel(0) \wedge speed(0, X) \wedge duration(0, D) \wedge (Y = X - a \times D).$$

[1] http://www.cs.utexas.edu/users/vl/tag/continuous_problem

[2] Rather than presenting the native syntax of the various formalisms, many descriptions will be given in this syntax similar to predicate logic but where free variables are capitalized to distinguish these from lower-case constants. Further, unless otherwise stated, formulas with free variables are to be understood as the universal closure of the formula.

which intuitively reads "If the speed is X at timestep 0 and the agent accelerates (or decelerates) for a duration of D, then the speed at timestep 1 is $Y = X + a \times D$ (or $Y = X - a \times D$)".

Any number of elaborations to the default behavior can be added to the domain description in this manner. This is unlike in classical logic where either the original rules must be revised to explicitly exclude situations according to these new elaborations or must include auxiliary abnormality constants that these new elaborations trigger.

However, due to the grounding-based computation of ASP, this encoding cannot be efficiently processed by standard solvers when the domain becomes too large. Further, when the domain becomes infinite, such as in the case of reasoning about continuous time, distance, or speed, ASP systems cannot compute solutions at all.

In addition, fluents in this domain such as *speed* and *duration* are functional in nature but are represented using relations and therefore must have the uniqueness and existence of these relations explicitly expressed in the encoding. These challenges have led to research into alternative formalisms that partially address these issues.

2.2 Constraint Answer Set Programming

While Answer Set Programming addressed the problem of performing defeasible reasoning on predicates, two issues that still challenge ASP are that defeasible reasoning cannot be performed on functions and efficient computation is not achievable when domains grow too large due to the grounding-based computation of ASP.

By loosely integrating constraint processing with ASP, Constraint Answer Set

Program (CASP) has been able to partially address the latter issue. At the same time, CASP allows for more general functions; while in ASP, functions are taken to be Herbrand (so that they must be mapped to themselves in any interpretation so that $f(1) = 1$ can never be true), in CASP, this is relaxed by processing functions using constraint solving rather than ASP. CASP has been implemented in solvers including ACSOLVER Mellarkod *et al.* (2008), CLINGCON Gebser *et al.* (2009b), EZCSP Balduccini (2009), IDP Mariën *et al.* (2008), and MINGO Liu *et al.* (2012).

However, CASP is unsuitable for performing defeasible reasoning on functions. CASP employs loose integration of ASP and CSP solvers. Consequently, functions are treated as in classical logic and so defeasible reasoning reasoning must either be performed by representing functions as predicates or by representing the defeasible reasoning as it is represented in classical logic–by modifying the existing rules for new elaborations or defining abnormality atoms that can be triggered by new elaborations.

For example, representing the rule (2.1) from the car example can be expressed using functions but would have to be modified with explicit exceptions like

$$\mathit{speed}(1) = Y \;\leftarrow\; \mathit{speed}(0) = Y \wedge \neg \mathit{accel}(0) \wedge \neg \mathit{decel}(0)$$

or require auxiliary abnormality atoms as in

$$\mathit{speed}(1) = Y \;\leftarrow\; \mathit{speed}(0) = Y \wedge \neg \mathit{abnormal}(0)$$

and then elaborations would trigger the abnormality with the rules

$$\begin{aligned} \mathit{abnormal}(0) &\leftarrow \mathit{accel}(0) \\ \mathit{abnormal}(0) &\leftarrow \mathit{decel}(0). \end{aligned}$$

Representing functions as predicates encounters the same issue as ASP that large domains will encounter a grounding bottleneck, while representing the defeasible reasoning as in classical logic either requires auxiliary constants or is not tolerant to elaborations.

2.3 Satisfiability Modulo Theories

The Boolean Satisfiability Problem, or SAT, is a well-studied logical formalism and while the restriction to propositional constants is inconvenient, SAT solver technology has been successfully applied to Answer Set Programming through the process of grounding into atoms that amount to propositional constants. However, SAT lacks the ability to represent information about functions. To enable more expressive reasoning, boolean satisfiability modulo theories, or SMT, considers the satisfiability of a formula subject to some background theory. Common background theories include the theory of arithmetic over reals or integers but background theories are very general and so complex concepts such as bit vectors, lists, and arrays can be represented in SMT.

For example, we can consider a simple formula which SAT solvers cannot handle but SMT solvers equipped with the background theories of integer arithmetic can:

$$a \ \vee \ b \ \vee \ 2 * f \geq g$$

An SMT solver will find many models among which are $\{b, f = 3, g = 8\}$ and $\{f = 4, g = 4\}$.

Efficient SMT solvers such as iSAT (https://projects.avacs.org/projects/isat/) and Z3 (http://z3.codeplex.com/releases) have built-in background theories including linear arithmetic, non-linear arithmetic, bit vectors, and quantifiers, among others; they have been successfully applied to challenging domains such as software verification, planning, model checking, and automated test generation. In addition, annual

SMT competitions promote further improvements to SMT solvers. However, functions here are still viewed under classical logic so SMT, too, is unsuitable for performing defeasible reasoning on functions.

2.4 Intensional Functions

To address the problem of performing defeasible reasoning on functions, several formalisms have been introduced that extend the stable model semantics to include the notion of *intensional functions* Cabalar (2011); Lifschitz (2012); Balduccini (2012). For example, the rule (2.1) can now be expressed using functions as

$$\mathit{speed}(1) = Y \vee \neg(\mathit{speed}(1) = Y) \;\leftarrow\; \mathit{speed}(0) = Y.$$

While this is a tautology in classical logic, under these extensions we can use rules like this to express default behavior that can be superseded in the presence of other knowledge.

However, these approaches focused on the modeling aspect and so the grounding bottleneck still poses a challenge to these formalisms. Additionally, the semantics described in Cabalar (2011); Balduccini (2012) are defined using a more complex notion of satisfaction than in the original stable model semantics while Lifschitz (2012) exhibits some unintuitive behavior that was not present in the original stable model semantics. These are explained in further detail in chapter 3

Chapter 3

TECHNICAL PRELIMINARIES

3.1 Reduct Characterization of the Stable Model Semantics

For a ground formula (i.e., a first-order formula with no variables), we define the answer sets in terms of a reduct. This definition is similar to the one given in Ferraris (2005) for propositional formulas that generalizes the original definition in Gelfond and Lifschitz (1988).

For two interpretations I, J of the same signature and a list $\boldsymbol{c}$ of distinct predicate constants, which we refer to as the "intensional predicates", we write $J <^{\mathbf{c}} I$ if

- J and I have the same universe and agree on all constants not in $\boldsymbol{c}$,
- $p^J \subseteq p^I$ for all predicates p in $\boldsymbol{c}$, and
- J and I do not agree on $\boldsymbol{c}$.

Example 2 *Consider four interpretations I, J, K, L with universe $\{1, 2\}$ and signature $\sigma = \{p, q\}$ where p and q are unary predicates. Let $\mathbf{c}$ be $\{p\}$ (so that q is non-intensional). When*

$$p^I = \{1\}, q^I = \{1\}$$
$$p^J = \emptyset, q^J = \{1\}$$
$$p^K = \{2\}, q^K = \{1\}$$
$$p^L = \{1\}, q^L = \{2\}$$

we can see that $J <^{\mathbf{c}} I$ holds since $p^J \subseteq p^I$ (the former has an empty extent while the latter has extent $\{1\}$) and J and I do not agree on $\mathbf{c}$ since J and I disagree on p. On *the other hand, $K <^{\mathbf{c}} I$ does not hold since $p^K \not\subseteq p^I$ (the former has extent $\{2\}$ while the latter has extent $\{1\}$). Finally, $L <^{\mathbf{c}} I$ does not hold since L and I do not agree on the constant q which is not in $\boldsymbol{c}$.*

The *reduct* F^I of a formula F relative to an interpretation I is the formula obtained from F by replacing every maximal subformula that is not satisfied by I with $\bot$ (falsity).

Definition 1 *For any interpretation I of σ, I is called an* answer set *of F if*

- *I satisfies F, and*
- *every interpretation J of σ such that $J <^{\mathbf{c}} I$ does not satisfy F^I.*

Example 3 *For example, consider a box initially at $l1$ which stays at its current location by default. This can be represented as (where L is a variable ranging over locations $l1$ and $l2$)*

$$at(box, 0, l1) \wedge Choice(at(box, 1, L)) \leftarrow at(box, 0, L)).$$

where $Choice(at(box, 1, L))$ is an abbreviation for $((at(box, 1, L) \vee \neg at(box, 1, L))$. Although this is a tautology in classical logic, in the answer set semantics, this can be used to describe default behaviors. This will be explained in more detail in Section 5.2. We consider the ground version F of this:

$$\begin{array}{c}
at(box, 0, l1)\wedge \\
((at(box, 1, l1) \quad \vee \quad \neg at(box, 1, l1)) \quad \leftarrow \quad at(box, 0, l1))\wedge \\
((at(box, 1, l2) \quad \vee \quad \neg at(box, 1, l2)) \quad \leftarrow \quad at(box, 0, l2))
\end{array}$$

We consider the interpretation I_1 such that $at(box, 0, l1)^{I_1} = \boldsymbol{t}$ and $at(box, 1, l1)^{I_1} = \boldsymbol{t}$ (and all other atoms are interpreted as $\boldsymbol{f}$). Clearly I_1 satisfies F so we then form the reduct F^{I_1}:

$$\begin{array}{c}
at(box, 0, l1)\wedge \\
((at(box, 1, l1) \quad \vee \quad \bot) \quad \leftarrow \quad at(box, 0, l1))\wedge \\
((\bot \quad \vee \quad \neg\bot) \quad \leftarrow \quad \bot)
\end{array}$$

If we take $\mathbf{c}$ *to be* $\{at\}$*, we can see that there is no interpretation* $J <^{\mathbf{c}} I_1$ *that satisfies* F^{I_1}*. Thus,* I_1 *is an answer set.*

On the other hand, if we consider I_2 *such that* $at(box, 0, l1)^{I_2} = \boldsymbol{t}$ *and* $at(box, 1, l2)^{I_2} = \boldsymbol{t}$ *(and all other atoms are interpreted as* $\boldsymbol{f}$*), we can see that* I_2 *satisfies* F*. We then form the reduct* F^{I_2}*:*

$$
\begin{array}{ccccc}
 & & at(box, 0, l1)\wedge & & \\
((\bot & \vee & \neg\bot) & \leftarrow & at(box, 0, l1))\wedge \\
((at(box, 1, l2) & \vee & \bot) & \leftarrow & \bot)
\end{array}
$$

and then we can see that the interpretation J *such that* $at(box, 0, l1)^J = \boldsymbol{t}$ *(and all other atoms are interpreted as* $\boldsymbol{f}$*) is an interpretation such that* $J <^{\mathbf{c}} I_2$ *and* J *satisfies* F^{I_2}*. Thus,* I_2 *is not an answer set.*

3.2 First Order Stable Model Semantics

We review the stable model semantics as defined in Ferraris *et al.* (2011) which presents an extension to the stable model semantics to first-order logic. This extension is defined in terms of second-order logic, where we have quantifiers over function and predicate variables in addition to the quantifiers over object variables as in first-order logic.

Formulas are built the same as in first-order logic. A *signature* consists of *function constants* and *predicate constants*. Function constants of arity 0 are called *object constants*. We assume the following set of primitive propositional connectives and quantifiers:

$$\bot \text{ (falsity)},\ \wedge,\ \vee,\ \rightarrow,\ \forall,\ \exists\ .$$

We understand $\neg F$ as an abbreviation of $F \rightarrow \bot$; symbol $\top$ stands for $\bot \rightarrow \bot$, and $F \leftrightarrow G$ stands for $(F \rightarrow G) \wedge (G \rightarrow F)$.

For predicate symbols (constants or variables) u and c, we define $u \leq c$ as $\forall \boldsymbol{x}(u(\boldsymbol{x}) \rightarrow c(\boldsymbol{x}))$. For two lists of predicate symbols $\boldsymbol{u}$ and $\boldsymbol{c}$, we define $\boldsymbol{u} \leq \boldsymbol{c}$ as the conjunction of $u \leq c$ for each $u \in \boldsymbol{u}$ and the corresponding $c \in \boldsymbol{c}$. We then define $\boldsymbol{u} < \boldsymbol{c}$ as $\boldsymbol{u} \leq \boldsymbol{c} \wedge \neg(\boldsymbol{c} \leq \boldsymbol{u})$.

Let $\boldsymbol{c}$ be a list of distinct predicate constants and let $\widehat{\boldsymbol{c}}$ be a list of distinct predicate variables corresponding to $\boldsymbol{c}$ [1] . We call members of $\boldsymbol{c}$ *intensional* predicates. We define $\mathrm{SM}[F;\ \boldsymbol{c}]$ as

$$F \wedge \neg \exists \widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F^*(\widehat{\boldsymbol{c}})),$$

where $F^*(\widehat{\boldsymbol{c}})$ is defined as follows:

- When F is an atomic formula, F^* is F', where F' is obtained from F by replacing every intensional predicate in it with the corresponding predicate variables;
- $(F \wedge G)^* = F^* \wedge G^*$; $\quad (F \vee G)^* = F^* \vee G^*$;
- $(F \rightarrow G)^* = (F^* \rightarrow G^*) \wedge (F \rightarrow G)$;
- $(\forall x F)^* = \forall x F^*$; $\quad (\exists x F)^* = \exists x F^*$.

Example 3 continued *For example, when F is*

$$at(box, 0, l1) \wedge ((at(box, 1, L) \vee \neg at(box, 1, L)) \leftarrow at(box, 0, L)),$$

[1] That is to say, $\boldsymbol{d}$ and $\widehat{\boldsymbol{c}}$ have the same length and the corresponding members have the same arity.

as before, then $F^*(\widehat{at})$ *is* [2]

$$\widehat{at}(box, 0, l1) \wedge ($$
$$(\ (at(box, 1, L) \vee \neg at(box, 1, L)) \leftarrow at(box, 0, L)\ )\wedge$$
$$(\ (\widehat{at}(box, 1, L) \vee (\neg at(box, 1, L) \wedge \neg\widehat{at}(box, 1, L))) \leftarrow \widehat{at}(box, 0, L)\ )\ ).$$

When F is a sentence, the models of SM$[F;\ \boldsymbol{c}]$ are called the $\boldsymbol{c}$-*stable* models of F. They are the models of F that are "stable" on $\boldsymbol{c}$. We often drop the list of constants when $\boldsymbol{c}$ is the entire signature.

Example 3 continued *Consider* I_1 *and* I_2 *from before.* $I_1 \models SM[F;\ at]$ *while* $I_2 \not\models SM[F;\ at]$. *Thus,* I_1 *is a stable model of* F *but* I_2 *is not.*

3.3 Constraint Answer Set Programming

A *constraint satisfaction problem* (CSP) is a tuple (V, D, C), where V is a set of *constraint variables* with the respective *domains* D, and C is a set of *constraints* that specify legal assignments of values in the domains to the constraint variables.

A *constraint answer set program* Π with a constraint satisfaction problem (V, D, C) is a set of rules of the form

$$a \leftarrow B, N, Cn, \tag{3.1}$$

where a is a propositional atom or $\bot$, B is a set of positive propositional literals, N is a set of negative propositional literals, and Cn is a set of constraints from C, possibly preceded by not.

[2] Recall $\neg at(box, 1, L)$ is an abbreviation for $at(box, 1, L) \to \bot$ so that $(\neg at(box, 1, L))^*(\widehat{at})$ is $(at(box, 1, L) \to \bot) \wedge (\widehat{at}(box, 1, L) \to \bot)$, which we then abbreviate as $\neg at(box, 1, L) \wedge \neg\widehat{at}(box, 1, L)$

For any signature σ that consists of object constants and propositional constants, we identify an interpretation I of σ as the tuple $\langle I^f, X\rangle$, where I^f is the restriction of I on the object constants in σ, and X is a set of propositional constants in σ that are true under I.

Given a constraint answer set program Π with (V, D, C), and an interpretation $I = \langle I^f, X\rangle$, we define the *constraint reduct of* Π *relative to* X *and* I^f (denoted by $\Pi^X_{I^f}$) as the set of rules $a \leftarrow B$ for each rule (11.1) is in Π such that $I^f \models Cn$, and $X \models N$. We say that a set X of propositional atoms is a *constraint answer set* of Π relative to I^f if X is a minimal model of $\Pi^X_{I^f}$.

Consider the water level example mentioned in the introduction. By default, the water level will stay the same but if the input valve is open, the water level will increase by 1 unit per time unit and if the output valve is open, the water level will decrease by 2 units per time unit (and if both are open, the water level will decrease by 1 unit per time unit).

Notice that object constants cannot appear in the heads of rules. In addition, the notion of answer set is defined using minimality only w.r.t. propositional atoms. Due to these restrictions, representing this domain using functions must be done either in a way that is not elaboration tolerant or using auxiliary abnormality atoms. The former is illustrated below:

$$\bot \leftarrow \neg\ \mathit{inputOpen}, \neg\ \mathit{outputOpen}, \neg(\mathit{amount}_1 = \mathit{amount}_0)$$
$$\bot \leftarrow\ \mathit{inputOpen}, \neg\ \mathit{outputOpen}, \neg(\mathit{amount}_1 + 1 = \mathit{amount}_0)$$
$$\bot \leftarrow \neg\ \mathit{inputOpen},\ \mathit{outputOpen}, \neg(\mathit{amount}_1 + 2 = \mathit{amount}_0)$$
$$\bot \leftarrow\ \mathit{inputOpen},\ \mathit{outputOpen}, \neg(\mathit{amount}_1 - 1 = \mathit{amount}_0)$$

Now if we wanted to elaborate on this domain by introducing a second input valve, this would require modifying all four of these rules and adding four more rules to handle all eight configurations of the valves.

3.4 Satisfiability Modulo Theories

Formally, an SMT instance is a formula in many-sorted first-order logic, where some designated function and predicate constants are constrained by some fixed background interpretation. SMT is the problem of determining whether such a formula has a model that expands the background interpretation Barrett *et al.* (2009).

Let σ^{bg} be the (many-sorted) signature of the background theory bg. An interpretation of σ^{bg} is called a *background interpretation* if it satisfies the background theory. For instance, in the theory of reals, we assume that σ^{bg} contains the set $\mathcal{R}$ of symbols for all real numbers, the set of arithmetic functions over real numbers, and the set $\{<, >, \leq, \geq\}$ of binary predicates over real numbers. Background interpretations interpret these symbols in the standard way.

Let σ be a signature that is disjoint from σ^{bg}. We say that an interpretation I of σ *satisfies* F w.r.t. the background theory bg, denoted by $I \models_{bg} F$, if there is a background interpretation J of σ^{bg} that has the same universe as I, and $I \cup J$ satisfies F. For any SMT sentence F with background theory σ^{bg}, interpretation I is a model of F (w.r.t. background theory σ^{bg}) if $I \models_{bg} F$.

Example 4 *Consider the formula F*

$$\forall x(f = x \rightarrow \exists y(y * y = x))$$

and consider the interpretations I_1 and I_2 such that $f^{I_1} = 1$ and $f^{I_2} = 2$ and where the universe is the set of real numbers. Now consider the background theory real that *is defined as standard arithmetic over reals. We have* $I_1 \models_{real} F$ *and* $I_2 \models_{real} F$ *(since* $\sqrt{2} * \sqrt{2} = 2$*).*

Now, take interpretations J_1 *and* J_2 *such that* $f^{J_1} = 1$ *and* $f^{J_2} = 2$ *where the universe is the set of integers. Now consider the background theory integer that is defined as standard arithmetic over reals. We have* $I_1 \models_{integer} F$ *but* $I_2 \not\models_{integer} F$*.*

It should be stressed that these background theories can be quite general; aside from integers, rationals, and reals, SMT can have background theories over bit-vectors, lists, and arrays to name a few. This generality has led to the use of SMT solvers in software engineering applications such as static program analysis Moy *et al.* (2009), fuzz testing Bounimova *et al.* (2013), and program verification Ge *et al.* (2007).

3.5 Lifschitz Semantics of Intensional Functions

We consider rules of the form

$$H \leftarrow B, \tag{3.2}$$

where H and B are formulas that do not contain $\rightarrow$. We identify a rule with the universal closure of the implication $B \rightarrow H$. An IF-program is a finite list of those rules.

An occurrence of a symbol in a formula F is *negated* if it belongs to a subformula of F that begins with negation, and is *non-negated* otherwise. Let F be a formula, let $\boldsymbol{f}$ be a list of distinct function constants, and let $\widehat{\boldsymbol{f}}$ be a list of distinct function variables corresponding to $\boldsymbol{f}$.

By $F^{\diamond}(\widehat{\boldsymbol{f}})$ we denote the formula obtained from F by replacing each non-negated occurrence of a member of $\boldsymbol{f}$ with the corresponding function variable in $\widehat{\boldsymbol{f}}$. By

IF$[F; \boldsymbol{f}]$ we denote the second-order sentence

$$F \wedge \neg\exists\widehat{\boldsymbol{f}}(\widehat{\boldsymbol{f}} \neq \boldsymbol{f} \wedge F^{\diamond}(\widehat{\boldsymbol{f}})).$$

According to Lifschitz (2012), the $\boldsymbol{f}$*-stable models* of an IF-program Π are defined as the models of IF$[F; \boldsymbol{f}]$, where F is the FOL-representation of Π.

An unexpected property of this extension is that stable models may map functions to constants not occurring anywhere in the formula. This is unexpected in the light of the *rationality principle* Gelfond and Kahl (2014) of the stable model semantics which states "Believe nothing you are not forced to believe". For example, consider the formula $c = 1 \rightarrow \bot$. The interpretation I of signature $\{c\}$ such that $|I| = \{1, 2\}$ and $c^I = 2$ is a stable model under the Lifschitz semantics despite 2 not occurring in the formula.

3.6 Cabalar Semantics

3.6.1 Partial Interpretations

Before formally reviewing the semantics for intensional functions from Cabalar (2011), we first define the notions of partial interpretations and partial satisfaction.

We define the notion of a *partial* interpretation as follows. Given a first-order signature σ comprised of function and predicate constants, a *partial* interpretation I of σ consists of

- a non-empty set $|I|$, called the *universe* of I;
- for every function constant f of arity n, a function f^I from $(|I| \cup \{u\})^n$ to $|I| \cup \{u\}$, where u is not in $|I|$ ("u" stands for *undefined*);

- for every predicate constant p of arity n, a function p^I from $(|I| \cup \{u\})^n$ to $\{1, 0\}$.

For each term $f(t_1, \ldots, t_n)$, we define

$$f(t_1, \ldots, t_n)^I = \begin{cases} u & \text{if } t_i^I = u \text{ for some } i \in \{1, \ldots, n\}; \\ f^I(t_1^I, \ldots, t_n^I) & \text{otherwise.} \end{cases}$$

The satisfaction relation $\models_p$ between a partial interpretation I and a first-order formula F is the same as the one for first-order logic except for the following base cases:

- For each atomic formula $p(t_1, \ldots, t_n)$,

$$p(t_1, \ldots, t_n)^I = \begin{cases} 0 & \text{if } t_i^I = u \text{ for some } i \in \{1, \ldots, n\}; \\ p^I(t_1^I, \ldots, t_n^I) & \text{otherwise.} \end{cases}$$

- For each atomic formula $t_1 = t_2$,

$$(t_1 = t_2)^I = \begin{cases} 1 & \text{if } t_1^I \neq u,\ t_2^I \neq u, \text{ and } t_1^I = t_2^I; \\ 0 & \text{otherwise.} \end{cases}$$

We say that $I \models_p F$ if $F^I = 1$.

Observe that under a partial interpretation, $t = t$ is not necessarily true: $I \not\models_p t = t$ iff $t^I = u$. On the other hand, $\neg(t_1 = t_2)$, also denoted by $t_1 \neq t_2$, is true under I even when both t_1^I and t_2^I are mapped to the same u.

3.6.2 Cabalar Semantics Definition

The Cabalar semantics was originally defined in Cabalar (2011) in terms of a modification to equilibrium logic.

Given any two partial interpretations J and I of the same signature σ, and a set of constants $\boldsymbol{c}$, we write $J \preceq^{\mathbf{c}} I$ if

- J and I have the same universe and agree on all constants not in $\boldsymbol{c}$;
- $p^J \subseteq p^I$ for all predicate constants in $\boldsymbol{c}$; and
- $f^J(\boldsymbol{\xi}) = u$ or $f^J(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$ for all function constants in $\boldsymbol{c}$ and all lists $\boldsymbol{\xi}$ of elements in the universe.

We write $J \prec^{\mathbf{c}} I$ if $J \preceq^{\mathbf{c}} I$ but not $I \preceq^{\mathbf{c}} J$. Note that $J \prec^{\mathbf{c}} I$ is defined similar to $J <^{\mathbf{c}} I$ (Section 3.1) except for the treatment of functions. Note that the third condition means essentially undefined functions are "smaller" than defined functions.

Example 5 *Consider four partial interpretations I, J, K, L with universe $\{1, 2\}$ and signature $\sigma = \{p, q, f\}$ where p and q are unary predicates and f is a unary function. Let $\mathbf{c}$ be $\{p, f\}$. When*

$$p^I = \{1\}, q^I = \{1\}, f^I = 1$$
$$p^J = \emptyset, q^J = \{1\}, f^J = u$$
$$p^K = \{1\}, q^K = \{1\}, f^K = 2$$
$$p^L = \{1\}, q^L = \{2\}, f^L = 1$$

we can see that $J \preceq^{\mathbf{c}} I$ holds since $p^K \subseteq p^I$ (the former has an empty extent while the latter has extent $\{1\}$) and $f^J = u$. However, $I \preceq^{\mathbf{c}} J$ does not hold since $f^I \neq f^J$ and $f^I \neq u$. Thus $J \prec^{\mathbf{c}} I$. Similarly, $K \preceq^{\mathbf{c}} I$ does not hold since $f^J \neq f^I$ and $f^J \neq u$. On the other hand, $L \preceq^{\mathbf{c}} I$ does not hold since L and I do not agree on the constant q which is not in $\mathbf{c}$.

A *PHT-interpretation* ("Partial HT-interpretation") $\mathcal{I}$ of signature σ is a tuple $\langle \mathcal{I}^h, \mathcal{I}^t \rangle$ such that $\mathcal{I}^h$ and $\mathcal{I}^t$ are partial interpretations of σ that have the same universe.

The *satisfaction* relation $\models_{pht}$ between a PHT-interpretation $\mathcal{I}$, a world $w \in \{h, t\}$ ordered by $h < t$, and a first-order sentence F of the signature σ is defined recursively:

- If F is an atomic formula, $\mathcal{I}, w \models_{pht} F$ if $\mathcal{I}^w \models_{p} F$;
- $\mathcal{I}, w \models_{pht} F \wedge G$ if $\mathcal{I}, w \models_{pht} F$ and $\mathcal{I}, w \models_{pht} G$;
- $\mathcal{I}, w \models_{pht} F \vee G$ if $\mathcal{I}, w \models_{pht} F$ or $\mathcal{I}, w \models_{pht} G$;
- $\mathcal{I}, w \models_{pht} F \rightarrow G$ if, for every world w' such that $w \leq w'$, $\mathcal{I}, w' \not\models_{pht} F$ or $\mathcal{I}, w' \models_{pht} G$;
- $\mathcal{I}, w \models_{pht} \forall x F(x)$ if, for every $\xi \in |\mathcal{I}|$, $\mathcal{I}, w \models_{pht} F(\xi^\diamond)$;
- $\mathcal{I}, w \models_{pht} \exists x F(x)$ if, for some $\xi \in |\mathcal{I}|$, $\mathcal{I}, w \models_{pht} F(\xi^\diamond)$.

We say that an HT-interpretation $\mathcal{I}$ *satisfies* F, written as $\mathcal{I} \models_{pht} F$, if $\mathcal{I}, h \models_{pht} F$.

A PHT-interpretation $\mathcal{I} = \langle I, I \rangle$ of signature σ is a *partial equilibrium model* of a sentence F relative to $\boldsymbol{c}$ if

- $\langle I, I \rangle \models_{pht} F$, and
- for every partial interpretation J such that $J \prec^{\mathbf{c}} I$, we have $\langle J, I \rangle \not\models_{pht} F$.

Example 3 continued *Consider again the formula describing the inertia of a box. Take $\mathcal{I}_1 = \langle I_1, I_1 \rangle$ and $\mathcal{I}_2 = \langle I_2, I_2 \rangle$ where I_1 and I_2 are from before. Relative to at, we can see that $\mathcal{I}_1$ is a partial equilibrium model while $\mathcal{I}_2$ is not. To show that $\mathcal{I}_1$ is a partial equilibrium model we consider the three partial interpretation J_1, J_2, J_3 that are such that $J_i \prec^{at} I_1$. These interpretations agree with I_1 except*

- $at(box, 0, l1)^{J_1} = u$;
- $at(box, 1, l1)^{J_2} = u$;

- $at(box, 0, l1)^{J_3} = u$ *and* $at(box, 1, l1)^{J_3} = u$.

$\langle J_1, I_1 \rangle$ *and* $\langle J_3, I_1 \rangle$ *both fail to satisfy* $at(box, 0, l1)$ *while* $\langle J_2, I_1 \rangle$ *fails to satisfy* $((at(box, 1, l1) \vee \neg at(box, 1, l1)) \leftarrow at(box, 0, l1))$. *Thus,* $\mathcal{I}_1$ *is a partial equilibrium model.*

On the other hand, if we consider J_4 *that agrees with* I_2 *except that* $at(box, 1, l2)^{J_4} = u$ *(so that* $J_4 \prec^{at} I_2$*), then we see that* $\langle I_2, J_4 \rangle \models_{pht} F$ *and so* $\mathcal{I}_2$ *is not a partial equilibrium model.*

3.7 Balduccini Semantics

Let us restrict a signature σ to be comprised of a set of *intensional* function and predicate constants denoted $\boldsymbol{c}$ as well as a set of *non-intensional* object constants $\sigma \setminus \boldsymbol{c}$.

In Balduccini (2012), Balduccini considered *terms* to have the form $f(c_1, \ldots, c_k)$ where f is an intensional function constant (in $\boldsymbol{c}$), and each c_i is a non-intensional object constant (in $\sigma \setminus \boldsymbol{c}$). He considered an *atom* to be an expression $p(c_1, \ldots, c_k)$ where p is an intensional predicate constant, and each c_i is a non-intensional object constant; a *t-atom* is an expression of the form $f = g$ where f is a term and g is either a term or a non-intensional object constant; a *seed t-atom* is a t-atom of the form $f = c$ where c is a non-intensional object constant. A *t-literal* is a t-atom $f = g$ or $\sim(f = g)$, where $\sim$ denotes *strong negation* [3] . A *seed literal* is an atom a, or $\sim a$, or a seed t-atom. A *literal* is an atom a, or $\sim a$, or a t-literal. An ASP{f} program

[3] The concept of *strong negation* is different from *default negation*. Intuitively, $\sim A$ represents that A is false while $\neg A$ represents that A is not known to be true. This will be explained in greater detail in Section 6.3.

consists of rules of the form

$$h \leftarrow l_1, \dots, l_m, not\ l_{m+1}, \dots, not\ l_n\ , \tag{3.3}$$

where h is a seed literal or $\bot$, and each l_i is a literal. An ASP{f} program is a finite set of rules. We identify rule (3.3) with an implication

$$l_1 \wedge \cdots \wedge l_m \wedge \neg l_{m+1} \wedge \cdots \wedge \neg l_n \rightarrow h\ ,$$

and an ASP{f} program as the conjunction of each implication corresponding to a rule in the program. Note that ASP{f} programs do not contain variables.

A set I of seed literals is said to be *consistent* if it contains no pair of an atom a and its strong negation $\sim a$; and contains no pair of seed t-atoms $t = c_1$ and $t = c_2$ such that $c_1 \neq c_2$. It is clear that any subset of a consistent set of seed literals is consistent as well.

The notion of satisfaction between a consistent set I of seed literals and literals is denoted by $\models_b$ and is defined as follows.

- For a seed literal l, $I \models_b l$ if $l \in I$;
- For a non-seed literal $f = g$, $I \models_b f = g$ if I contains both $f = c$ and $g = c$ for some object constant c;
- For a non-seed literal $\sim(f = g)$, $I \models_b \sim(f = g)$ if I contains both $f = c_1$ and $g = c_2$ for some object constants c_1 and c_2 such that $c_1 \neq c_2$.

This notion of satisfaction is extended to formulas allowing $\wedge$, $\neg$ and $\leftarrow$ as in classical logic.

The reduct of an ASP{f} program Π relative to a consistent set I of seed literals is denoted $\Pi^{\underline{I}}$ and is defined as

$$\Pi^{\underline{I}} = \{h \leftarrow l_1\ \dots, l_m \mid (3.3) \in \Pi \text{ and } I \models \neg l_{m+1} \wedge \cdots \wedge \neg l_n\}\ .$$

I is called a *Balducchini answer set* of Π if

- $I \models_b \Pi^{\underline{I}}$, and,
- for every proper subset J of I, we have $J \not\models_b \Pi^{\underline{I}}$.

Example 3 continued *Consider again the example of describing the inertia of a box. Since the head of a rule must be a seed literal or $\perp$, we express this as the program Π*

$$at(box,0) = l1$$
$$at(box,1) = l1 \leftarrow at(box,0) = l1, \neg \sim at(box,1) = l1$$
$$at(box,1) = l2 \leftarrow at(box,0) = l2, \neg \sim at(box,1) = l2$$

Recall the interpretations I_1 and I_2 from before; I_1 is such that $at(box,0,l1)^{I_1} = \boldsymbol{t}$ and $at(box,1,l1)^{I_1} = \boldsymbol{t}$ while I_2 is such that $at(box,0,l1)^{I_2} = \boldsymbol{t}$ and $at(box,1,l2)^{I_2} = \boldsymbol{t}$. So we consider the corresponding sets $J_1 = \{at(box,0) = l1, at(box,1) = l1\}$ and $J_2 = \{at(box,0) = l1, at(box,1) = l2\}$.

The reduct $\Pi^{\underline{J_1}}$ is

$$at(box,0) = l1$$
$$at(box,1) = l1 \leftarrow at(box,0) = l1.$$

It is clear that $J_1 \models_b \Pi^{\underline{J_1}}$. For any subset K of J_1 we have that $K \not\models_b \Pi^{\underline{I_1}}$ so I_1 is a Balducchini answer set of Π.

The reduct $\Pi^{\underline{J_2}}$ is

$$at(box,0) = l1$$
$$at(box,1) = l2 \leftarrow at(box,0) = l2.$$

It is clear that $J_1 \models_b \Pi^{\underline{J_2}}$. However, if we take $K = \{at(box,0) = l1\}$, we see that there is a subset K of J_2 such that $K \models_b \Pi^{\underline{J_2}}$ and so J_2 is not a Balducchini answer set of Π.

3.8 Multi-valued Propositional Formulas

The convenience of multi-valued propositional formulas for knowledge representation is demonstrated in the context of nonmonotonic causal theories and action language $\mathcal{C}+$ Giunchiglia *et al.* (2004). Multi-valued formulas serve as a simple but useful special case of first-order formulas for use in establishing some results and serve as the theoretical context for system MVSM described in Chapter 9.

A multi-valued signature is a set σ of symbols called *constants*, along with a finite set $Dom(c)$ of symbols that is disjoint from σ and contains at least two elements, assigned to each constant c. We call $Dom(c)$ the *domain* of c. A *multi-valued atom* of σ is $\bot$, or an expression of the form $c{=}v$ ("the value of c is v") where $c \in \sigma$ and $v \in Dom(c)$. A *multi-valued formula* of σ is a propositional combination of multi-valued atoms.

A *multi-valued interpretation* of σ is a function that maps every element of σ to an element in its domain. We often identify an interpretation with the set of atoms of σ that are satisfied by I. A multi-valued interpretation I *satisfies* an atom $c{=}v$ (symbolically, $I \models c{=}v$) if $I(c) = v$. The satisfaction relation is extended from atoms to arbitrary formulas according to the usual truth tables for the propositional connectives. We say that I is a *model* of F if it satisfies F.

An expression of the form $c{=}d$, where both c and d are constants, will be understood as an abbreviation for the formula

$$\bigvee_{v \in Dom(c) \cap Dom(d)} (c{=}v \land d{=}v). \tag{3.4}$$

Let F be a multi-valued formula of signature σ, and let I be a multi-valued interpretation of σ. The *reduct* of F relative to I (denoted $F^{\underline{I}}$) is the formula obtained from F by replacing each (maximal) subformula that is not satisfied by I with $\bot$. We call I a *multi-valued stable model* of F if I is the *only* multi-valued interpretation of σ that satisfies $F^{\underline{I}}$.

Example 6 *Take $\sigma = \{c\}$ and $Dom(c) = \{1, 2, 3\}$, and let F be*

$$c=1 \vee \neg(c=1), \tag{3.5}$$

and let I_i ($i = 1, 2, 3$) be the interpretation that maps c to i. All three interpretations satisfy (3.5)*, but I_1 is the only stable model of F: the reduct $F^{\underline{I_1}}$ is $c=1 \vee \bot$, and I_1 is the only model of the reduct; the reduct of F_1 relative to other interpretations is $\bot \vee \neg\bot$, which does not have a unique model.*

If we conjoin $c=2$ with (3.5)*, we can check that the only stable model is $\{c=2\}$.*

3.9 Partial Multi-valued Propositional Formulas

In this section we introduce a variant of the stable model semantics in the previous section, which allows multi-valued propositional constants to be mapped to nothing. This is essentially a simple special case of the semantics in Cabalar 2011, and later in Balduccini 2013, which allows functions to be partially defined. In other words, interpretations are allowed to leave some constants undefined. By *complete* interpretations, we mean a special case of partial interpretations where all constants are defined. Complete interpretations can be identified with classical ("total") interpretations.

We consider the same syntax of a multi-valued formula as in the previous section. As with total interpretations, a partial interpretation I *satisfies* an atom $c = v$ if $I(c)$ is defined and is mapped to v. This implies that an interpretation that is undefined on c does not satisfy any atom of the form $c = w$ for any $w \in Dom(c)$. As before, it is convenient to identify a partial interpretation I with the set of atoms of σ that are satisfied by this interpretation. For instance, an interpretation of $\sigma = \{c\}$ which is undefined on c is identified with the empty set. Again, the satisfaction relation is extended from atoms to arbitrary formulas according to the usual truth tables for the propositional connectives. We call I a *model* of F if it satisfies F.

The reduct $F^{\underline{I}}$ is defined to be the same as before. We say that a partial interpretation I is a *partial multi-valued stable model* of F if I satisfies F and no proper subset J of I satisfies $F^{\underline{I}}$.

Example 6 continued

In this context, $c\!=\!1 \vee \neg(c\!=\!1)$ *does not mean that* c *is mapped to* 1 *by default. Instead, it means that* c *can be mapped to* 1 *or nothing at all. As before, the reduct* $F^{\underline{I_1}}$ *relative to* I_1 *where* I_1 *is* $\{c\!=\!1\}$ *is* $c\!=\!1 \vee \bot$, *and* I_1 *is the minimal model of the reduct.* [4] *Further, the reduct* $F^{\underline{I_0}}$ *relative to* I_0 *where* I_0 *is* $\emptyset$ *is* $\bot \vee \neg\bot$, *and* I_0 *is the minimal model of the reduct.*

[4] Minimality is understood in terms of set inclusion.

Chapter 4

FUNCTIONAL STABLE MODEL SEMANTICS

4.1 Reduct-Based Characterization

4.1.1 Infinitary Ground Formulas and Grounding

We first present the reduct-based characterization of the functional stable model semantics. However, since we allow the universe to be infinite, grounding a quantified sentence introduces infinite conjunctions and disjunctions over the elements in the universe. Here we rely on the concept of grounding *relative to an interpretation* from Truszczynski (2012). The following is the definition of an *infinitary ground formula*, which is adapted from Truszczynski (2012). One difference is that we do not replace ground terms with their corresponding object names, leaving them unchanged during grounding. This change is necessary in defining a reduct for functional stable model semantics. For each element ξ in the universe $|I|$ of I, we introduce a new symbol $\xi^\diamond$, called an *object name*. By σ^I we denote the signature obtained from σ by adding all object names $\xi^\diamond$ as additional object constants. We will identify an interpretation I of signature σ with its extension to σ^I defined by $I(\xi^\diamond) = \xi$. [1]

We assume the primary connectives to be $\bot$, $\{\}^\wedge$, $\{\}^\vee$, and $\rightarrow$. Propositional connectives $\wedge, \vee, \neg, \top$ are considered as shorthands: $F \wedge G$ as $\{F, G\}^\wedge$; $F \vee G$ as $\{F, G\}^\vee$. $\neg$ and $\top$ are defined as before.

[1] For details, see Lifschitz *et al.* (2008).

Let A be the set of all ground atomic formulas of signature σ^I. The sets $\mathcal{F}_0, \mathcal{F}_1, \ldots$ are defined recursively as follows:

- $\mathcal{F}_0 = A \cup \{\bot\}$;
- $\mathcal{F}_{i+1} (i \geq 0)$ consists of expressions $\mathcal{H}^\vee$ and $\mathcal{H}^\wedge$, for all subsets $\mathcal{H}$ of $\mathcal{F}_0 \cup \ldots \cup \mathcal{F}_i$, and of the expressions $F \rightarrow G$, where $F, G \in \mathcal{F}_0 \cup \cdots \cup \mathcal{F}_i$.

We define $\mathcal{L}_A^{inf} = \bigcup_{i=0}^{\infty} \mathcal{F}_i$, and call elements of $\mathcal{L}_A^{inf}$ *infinitary ground formulas* of σ w.r.t. I.

For any interpretation I of σ and any infinitary ground formula F w.r.t. I, the definition of satisfaction, $I \models F$, is as follows:

- For atomic formulas, the definition of satisfaction is the same as in the standard first-order logic;
- $I \models \mathcal{H}^\vee$ if there is a formula $G \in \mathcal{H}$ such that $I \models G$;
- $I \models \mathcal{H}^\wedge$ if, for every formula $G \in \mathcal{H}$, $I \models G$;
- $I \models G \rightarrow H$ if $I \not\models G$ or $I \models H$.

Example 7 *Consider a domain that is comprised of a bucket that has a leak and initially contains some amount of water. By default, the bucket will lose one unit of water at each timepoint. If we consider ten timepoints, then we have* $\sigma = \{bucket_0, \ldots, bucket_9\}$.

Let $\mathcal{F}$ *be the infinite set of ground formulas* $\{bucket_0 = i : i \in \mathcal{N}\}$. *By* $\mathcal{F}^\vee$, *we can represent that the bucket initially contains some amount of water.*

Let $\mathcal{G}_i$ *be the infinite set of ground formulas* $\{bucket_i = j+1 \rightarrow bucket_{i+1} = j : j \in \mathcal{N}\}$. *By* $\mathcal{G}_i^\wedge$ *we can represent that from timepoint* i *to timepoint* $i+1$, *the bucket loses one unit of water. Then, if we let* $\mathcal{H}$ *be the finite set of sets* $\{G_i : i \in \{0, \ldots, 8\}\}$, *we* *can represent this behavior for every timepoint by* $\mathcal{H}^{\wedge}$. *Thus, we can represent this domain as* $\{\mathcal{F}^{\vee}, \mathcal{H}^{\wedge}\}^{\wedge}$.

Consider the interpretation I *such that* $|I| = \mathcal{N}$ *and* $bucket_0 = 15$, $bucket_1 = 14$, $\ldots$, $bucket_9 = 6$.

- *First, we see that* $I \models \mathcal{F}^{\vee}$ *since* $I \models bucket_0 = i$ *where* $i = 15$.
- *Then, we can see that* $I \models G_0$ *since* $I \models bucket_0 = j+1 \rightarrow bucket_1 = j$ *for every* $j \in \mathcal{N}$; *for every* $j \neq 14$, $I \not\models bucket_0 = j+1$ *and so* I *vacuously satisfies the implication, but when* $j = 14$, *we see that* $I \models bucket_1 = 14$ *and so* $I \models G_0$.
- *Similar arguments show that* $I \models G_i$ *for each* $i \in \{1, \ldots, 8\}$ *and so we conclude that* $I \models \mathcal{H}^{\wedge}$ *and consequently,* $I \models \{\mathcal{F}^{\vee}, \mathcal{H}^{\wedge}\}^{\wedge}$.

Given a first-order sentence F, and an interpretation I, by $gr_I[F]$ we denote the infinitary ground formula w.r.t. I that is obtained from F by the following process:

- If F is an atomic formula, $gr_I[F]$ is F;
- $gr_I[G \odot H] = gr_I[G] \odot gr_I[H] \quad (\odot \in \{\wedge, \vee, \rightarrow\})$;
- $gr_I[\exists x G(x)] = \{gr_I[G(\xi^{\diamond})] \mid \xi \in |I|\}^{\vee}$;
- $gr_I[\forall x G(x)] = \{gr_I[G(\xi^{\diamond})] \mid \xi \in |I|\}^{\wedge}$.

Example 7 continued *Consider an elaboration to the bucket example where an agent can fill up the bucket at time* t *where* $t \in \{0..8\}$ *with the action* $fillUp_t$, *which restores the bucket to its maximum capacity–10–at the next timestep* $t+1$. *We can* *represent this domain with the first-order formula F:*

$$\forall y((amount_1\!=\!y) \vee \neg(amount_1\!=\!y) \;\leftarrow\; amount_0\!=\!y\!+\!1)$$

$$\forall y((amount_2\!=\!y) \vee \neg(amount_2\!=\!y) \;\leftarrow\; amount_1\!=\!y\!+\!1)$$

$$\ldots$$

$$\forall y((amount_9\!=\!y) \vee \neg(amount_9\!=\!y) \;\leftarrow\; amount_8\!=\!y\!+\!1)$$

$$amount_1\!=\!10 \;\leftarrow\; \mathit{fillUp}_0$$

$$amount_2\!=\!10 \;\leftarrow\; \mathit{fillUp}_1$$

$$\ldots$$

$$amount_9\!=\!10 \;\leftarrow\; \mathit{fillUp}_8$$

Now, consider an interpretation I such that the universe $|I| = \mathcal{N}$. $gr_I[F]$ is the following set of formulas.

$$(amount_1\!=\!0) \vee \neg(amount_1\!=\!0) \;\leftarrow\; amount_0\!=\!0\!+\!1$$

$$(amount_1\!=\!1) \vee \neg(amount_1\!=\!1) \;\leftarrow\; amount_0\!=\!1\!+\!1$$

$$(amount_1\!=\!2) \vee \neg(amount_1\!=\!2) \;\leftarrow\; amount_0\!=\!2\!+\!1$$

$$\ldots$$

$$(amount_2\!=\!0) \vee \neg(amount_2\!=\!0) \;\leftarrow\; amount_1\!=\!0\!+\!1$$

$$(amount_2\!=\!1) \vee \neg(amount_2\!=\!1) \;\leftarrow\; amount_1\!=\!1\!+\!1$$

$$(amount_2\!=\!2) \vee \neg(amount_2\!=\!2) \;\leftarrow\; amount_1\!=\!2\!+\!1$$

$$\ldots$$

$$amount_1\!=\!10 \;\leftarrow\; \mathit{fillUp}_0$$

$$amount_2\!=\!10 \;\leftarrow\; \mathit{fillUp}_1$$

$$\ldots$$

$$amount_9\!=\!10 \;\leftarrow\; \mathit{fillUp}_8$$

4.1.2 Reduct-Based Characterization

Let F be any first-order sentence of a signature σ, and let I be an interpretation of σ.

For any two interpretations I, J of the same signature and any list $\boldsymbol{c}$ of distinct predicate and function constants, we write $J <^{\mathbf{c}} I$ if

- J and I have the same universe and agree on all constants not in $\boldsymbol{c}$;
- $p^J \subseteq p^I$ for all predicate constants p in $\boldsymbol{c}$; and
- J and I do not agree on $\boldsymbol{c}$.

The difference between the above definition and the definition in Section 3.1 is only in that here, $\boldsymbol{c}$ is not restricted to contain only predicate constants.

Example 8 *Consider four interpretations I, J, K, L with universe $\{1, 2\}$ and signature $\sigma = \{p, q, f\}$ where p and q are unary predicates and f is a unary function. Let $\boldsymbol{c}$ be $\{p, f\}$. When*

$$p^I = \{1\}, q^I = \{1\}, f^I = 1$$
$$p^J = \{1\}, q^J = \{1\}, f^J = 2$$
$$p^K = \emptyset, q^K = \{1\}, f^K = 1$$
$$p^L = \{1\}, q^L = \{2\}, f^L = 1$$

we can see that $J <^{\mathbf{c}} I$ holds since $p^J \subseteq p^I$ (both have an extent of $\{1\}$) and J and I do not agree on $\mathbf{c}$ since $f^I \neq f^J$. Similarly, $K <^{\mathbf{c}} I$ holds since $p^K \subseteq p^I$ (the former has an empty extent while the latter has extent $\{1\}$) and K and I disagree on p. On the other hand, $L <^{\mathbf{c}} I$ does not hold since L and I do not agree on the constant q, which is not in $\boldsymbol{c}$.

The *reduct* $F^{\underline{I}}$ of an infinitary ground formula F relative to an interpretation I is defined as follows:

- For each atomic formula F, $F^{\underline{I}} = F$
- $(\mathcal{H}^\wedge)^{\underline{I}} = \{G^{\underline{I}} \mid G \in \mathcal{H}\}^\wedge$;
- $(\mathcal{H}^\vee)^{\underline{I}} = \{G^{\underline{I}} \mid G \in \mathcal{H}\}^\vee$;
- $(G \to H)^{\underline{I}} = \bot$ if $I \not\models G \to H$; otherwise $(G \to H)^{\underline{I}} = G^{\underline{I}} \to H^{\underline{I}}$.

Similar to the definition in section 3.1, for any interpretation I of σ, I is an answer set of of an infinitary ground formula F iff

- I satisfies F, and
- every interpretation J of σ such that $J <^{\boldsymbol{c}} I$ does not satisfy $F^{\underline{I}}$.

Example 7 continued *For simplicity, let us consider the same domain but with only two timesteps–0 and 1. Consider the interpretation I_1 such that $|I_1| = \mathcal{N}$, $amount_1^{I_1} = 5$, $amount_0^{I_1} = 6$, and $fillUp_0^{I_1} = \boldsymbol{f}$. The reduct $(gr_{I_1}[F])^{\underline{I_1}}$ is*

$$\bot \vee \neg\bot \;\leftarrow\; \bot$$

$$\bot \vee \neg\bot \;\leftarrow\; \bot$$

$$\bot \vee \neg\bot \;\leftarrow\; \bot$$

$$\dots$$

$$(amount_1\!=\!5) \vee \bot \;\leftarrow\; amount_0\!=\!5\!+\!1.$$

$$\dots$$

$$\bot \;\leftarrow\; \bot$$

which is equivalent to

$$(amount_1\!=\!5) \leftarrow\; amount_0\!=\!6. \tag{4.1}$$

No interpretation that is different from I_1 only on $amount_1$ satisfies the reduct.

On the other hand consider the interpretation I_2 such that $|I_2| = \mathcal{N}$, $amount_1^{I_2} = 8$, $amount_0^{I_2} = 6$, and $fillUp_0^{I_2} = \boldsymbol{f}$, the reduct $(gr_{I_2}[F])^{\underline{I_2}}$ is equivalent to

$$\bot \vee \neg\bot \;\leftarrow\; amount_0 = 5+1,$$

or simply $\top$, and we can find another interpretation that is different from I_2 only on $amount_1$ which satisfies the reduct. For example, take J such that $|J| = \mathcal{N}$, $amount_1^J = 3$, $amount_0^J = 6$, and $fillUp_0^J = \boldsymbol{f}$.

4.2 Second-Order Logic Characterization

We now present a characterization of the functional stable model semantics for formulas which are built the same as in first-order logic. A signature consists of function constants and predicate constants. Function constants of arity 0 are called object constants. We assume the following set of primitive propositional connectives and quantifiers:

$$\bot \text{ (falsity)},\ \wedge,\ \vee,\ \rightarrow,\ \forall,\ \exists\ .$$

As before, we understand $\neg F$ as an abbreviation of $F \rightarrow \bot$, $\top$ as an abbreviation of $\bot \rightarrow \bot$, and $F \leftrightarrow G$ as an abbreviation for $(F \rightarrow G) \wedge (G \rightarrow F)$.

Our characterization of these formulas uses second-order logic, where we have quantifiers over function and predicate variables in addition to the quantifiers over object variables as in first-order logic. For predicate symbols (constants or variables) u and c, we define $u \leq c$ as $\forall \boldsymbol{x}(u(\boldsymbol{x}) \rightarrow c(\boldsymbol{x}))$. We define $u = c$ as $\forall \boldsymbol{x}(u(\boldsymbol{x}) \leftrightarrow c(\boldsymbol{x}))$ if u and c are predicate symbols, and $\forall \boldsymbol{x}(u(\boldsymbol{x}) = c(\boldsymbol{x}))$ if they are function symbols.

For lists of predicate symbols (constants or variables) $\boldsymbol{u}$ and $\boldsymbol{c}$, we define $\boldsymbol{u} \leq \boldsymbol{c}$ as the conjunction of $u \leq c$ for each $u \in \boldsymbol{u}$ and the corresponding $c \in \boldsymbol{c}$. We define $\boldsymbol{u} = \boldsymbol{c}$ as the conjunction of $u = c$ for each $u \in \boldsymbol{u}$ and the corresponding $c \in \boldsymbol{c}$.

Let $\boldsymbol{c}$ be a list of distinct predicate and function constants and let $\widehat{\boldsymbol{c}}$ be a list of distinct predicate and function variables corresponding to $\boldsymbol{c}$.

By $\boldsymbol{c}^{pred}$ we mean the list of the predicate constants in $\boldsymbol{c}$, and by $\widehat{\boldsymbol{c}}^{pred}$ the list of the corresponding predicate variables in $\widehat{\boldsymbol{c}}$. We define $\widehat{\boldsymbol{c}} < \boldsymbol{c}$ as

$$(\widehat{\boldsymbol{c}}^{pred} \leq \boldsymbol{c}^{pred}) \wedge \neg(\widehat{\boldsymbol{c}} = \boldsymbol{c})$$

and SM$[F;\ \boldsymbol{c}]$ as

$$F \wedge \neg\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F^*(\widehat{\boldsymbol{c}})),$$

where $F^*(\widehat{\boldsymbol{c}})$ is defined as follows.

- When F is an atomic formula, F^* is $F' \wedge F$, where F' is obtained from F by replacing all intensional (function and predicate) constants in it with the corresponding (function and predicate) variables; [2]
- $(F \wedge G)^* = F^* \wedge G^*$; $\quad (F \vee G)^* = F^* \vee G^*$;
- $(F \rightarrow G)^* = (F^* \rightarrow G^*) \wedge (F \rightarrow G)$;
- $(\forall x F)^* = \forall x F^*$; $\quad (\exists x F)^* = \exists x F^*$.

Example 7 continued *If F is the formula*

$$(amount_1 = Y) \vee \neg(amount_1 = Y) \ \leftarrow\ amount_0 = Y+1$$

$$amount_1 = 10 \ \leftarrow\ \mathit{fillUp}$$

[2] If an atomic formula F contains no intensional function constants, then F^* can be defined as F', as in Ferraris *et al.* (2011).

and $\boldsymbol{c}$ *is* $\{amount_1\}$ *then* $F^*(\widehat{amount}_1)$ *is equivalent to* [3]

$$(amount_1 = Y) \vee \neg(amount_1 = Y) \;\leftarrow\; amount_0 = Y+1$$
$$((amount_1 = Y) \wedge (\widehat{amount}_1) = Y)) \vee \neg(amount_1 = Y) \;\leftarrow\; amount_0 = Y+1$$
$$amount_1 = 10 \;\leftarrow\; \mathit{fillUp}$$
$$amount_1 = 10 \wedge \widehat{amount}_1 = 10 \;\leftarrow\; \mathit{fillUp}$$

When F is a sentence, the models of SM$[F;\ \boldsymbol{c}]$ are called the $\boldsymbol{c}$-*stable* models of F. They are the models of F that are "stable" on $\boldsymbol{c}$.

Example 7 continued *Consider interpretation* I *and* I_1 *from before:*
$|I_1| = \mathcal{N},\ amount_1^{I_1} = 5,\ amount_0^{I_1} = 6,\ \mathit{fillUp}^{I_1} = \boldsymbol{f}$, *and*
$|I_2| = \mathcal{N},\ amount_1^{I_2} = 8,\ amount_0^{I_2} = 6,\ \mathit{fillUp}^{I_2} = \boldsymbol{f}$.
$I_1 \models SM[F;\ amount_1]$ *but* $I_2 \not\models SM[F;\ amount_1]$.

The following theorem states the equivalence between this formulation and the formulation in terms of grounding and reduct from the previous section.

Theorem 1 *Let* F *be a first-order sentence of signature* σ *and let* $\mathbf{c}$ *be a list of intensional constants. For any interpretation* I *of* σ, $I \models SM[F; \boldsymbol{c}]$ *iff*

- I *satisfies* F, *and*
- *every interpretation* J *such that* $J <^{\mathbf{c}} I$ *does not satisfy* $(gr_I[F])^{\underline{I}}$.

If $\boldsymbol{c}$ contains predicate constants only, this definition of a stable model reduces to the one in Ferraris *et al.* (2011). The definition of F^* above is the same as in Ferraris *et al.* (2011) except for the case when F is an atomic formula.

[3]Recall $\neg(amount_1 = Y)$ is an abbreviation for $amount_1 = Y \to \bot$ so that $(\neg(amount_1 = Y))^*(\widehat{amount}_1)$ is $(amount_1 = Y \to \bot) \wedge (amount_1 = Y \wedge \widehat{amount}_1 = Y \to \bot)$ which is equivalent to $\neg(amount_1 = Y) \wedge (\neg(amount_1 = Y) \vee \neg(\widehat{amount}_1 = Y))$ or simply $\neg(amount_1 = Y)$.

4.3 HT Logic Characterization

The functional stable model semantics can be reformulated in terms of a modification to equilibrium logic, similar to the way the Cabalar semantics Cabalar (2011) is defined, which is reviewed in Section 3.6.2. Recharacterizing the semantics in a way similar to the Cabalar semantics helps to see the relationship between the two semantics, which is explored in detail in Chapter 10.

An *FHT-interpretation* ("Functional HT-interpretation") $\mathcal{I}$ of signature σ is a tuple $\langle \mathcal{I}^h, \mathcal{I}^t \rangle$ such that $\mathcal{I}^h$ and $\mathcal{I}^t$ are classical interpretations of σ that have the same universe. The satisfaction relation $\models_{fht}$ between an FHT-interpretation $\mathcal{I}$, a world $w \in \{h, t\}$ ordered by $h < t$, and a first-order sentence of signature σ is defined in the same way as $\models_{pht}$ for PHT-interpretations in Section 3.6.2 except for the base case:

- If F is an atomic formula, $\mathcal{I}, w \models_{fht} F$ if, for every world w' such that $w \leq w'$, $\mathcal{I}^{w'} \models F$;
- $\mathcal{I}, w \models_{fht} F \wedge G$ if $\mathcal{I}, w \models_{fht} F$ and $\mathcal{I}, w \models_{fht} G$;
- $\mathcal{I}, w \models_{pht} F \vee G$ if $\mathcal{I}, w \models_{fht} F$ or $\mathcal{I}, w \models_{fht} G$;
- $\mathcal{I}, w \models_{fht} F \rightarrow G$ if, for every world w' such that $w \leq w'$, we have $\mathcal{I}, w' \not\models_{fht} F$ or $\mathcal{I}, w' \models_{fht} G$;
- $\mathcal{I}, w \models_{fht} \forall x F(x)$ if for each $\xi \in |\mathcal{I}|$, we have $\mathcal{I}, w \models_{fht} F(\xi^\diamond)$;
- $\mathcal{I}, w \models_{fht} \exists x F(x)$ if for some $\xi \in |\mathcal{I}|$, we have $\mathcal{I}, w \models_{fht} F(\xi^\diamond)$.

We say that FHT-interpretation $\mathcal{I}$ *satisfies* F, written as $\mathcal{I} \models_{fht} F$, if $\mathcal{I}, h \models_{fht} F$.

Example 9 *Consider the formula F that is $\forall x(p(x) \to q(x))$. Now take an interpretation $\mathcal{I} = \langle \mathcal{I}^h, \mathcal{I}^t \rangle$ such that*

$$\begin{array}{ll} p(1)^{\mathcal{I}^h} = \boldsymbol{t} & p(1)^{\mathcal{I}^t} = \boldsymbol{f} \\ p(2)^{\mathcal{I}^h} = \boldsymbol{f} & p(2)^{\mathcal{I}^t} = \boldsymbol{t} \\ q(1)^{\mathcal{I}^h} = \boldsymbol{t} & q(1)^{\mathcal{I}^t} = \boldsymbol{t} \\ q(2)^{\mathcal{I}^h} = \boldsymbol{f} & q(2)^{\mathcal{I}^t} = \boldsymbol{t} \end{array}$$

We will see that $\mathcal{I} \models_{\overline{fht}} F$ or rather, $\mathcal{I}, h \models_{\overline{fht}} F$. We must show both

- *$\mathcal{I}, h \models_{\overline{fht}} p(1) \to q(1)$. To verify this, we must show both*
  - *$\mathcal{I}, h \not\models_{\overline{fht}} p(1)$ or $\mathcal{I}, h \models_{\overline{fht}} q(1)$. This holds since $\mathcal{I}^t \models q(1)$ and $\mathcal{I}^h \models q(1)$ and so we have $\mathcal{I}, h \models_{\overline{fht}} q(1)$.*
  - *$\mathcal{I}, t \not\models_{\overline{fht}} p(1)$ or $\mathcal{I}, t \models_{\overline{fht}} q(1)$ (recall $h < t$). This holds since $\mathcal{I}^t \models q(1)$ and so we have $\mathcal{I}, t \models_{\overline{fht}} q(1)$.*
- *$\mathcal{I}, h \models_{\overline{fht}} p(2) \to q(2)$. To verify this, we must show both*
  - *$\mathcal{I}, h \not\models_{\overline{fht}} p(2)$ or $\mathcal{I}, h \models_{\overline{fht}} q(2)$. This holds since $\mathcal{I}^h \not\models p(2)$ and so we have $\mathcal{I}, h \not\models_{\overline{fht}} p(2)$. Note, in this case, we do not have $\mathcal{I}, h \models_{\overline{fht}} q(2)$ since $\mathcal{I}^h \not\models q(2)$.*
  - *$\mathcal{I}, t \not\models_{\overline{fht}} p(2)$ or $\mathcal{I}, t \models_{\overline{fht}} q(2)$. This holds since $\mathcal{I}^t \models q(2)$ and so we have $\mathcal{I}, t \models_{\overline{fht}} q(2)$. Note, in this case, we do not have $\mathcal{I}, t \not\models_{\overline{fht}} p(2)$ since $\mathcal{I}^t \models p(2)$.*

The following theorem [4] asserts the correctness of the reformulation of the Func-

[4] Recall the definition of $J <^{\boldsymbol{c}} I$ from section 4.1

- J and I have the same universe and agree on all constants not in $\boldsymbol{c}$;
- $p^J \subseteq p^I$ for all predicate constants p in $\boldsymbol{c}$; and
- J and I do not agree on $\boldsymbol{c}$.

tional Stable Model semantics in terms of equilibrium logic style.

Theorem 2 *Let F be a first-order sentence of signature σ and let $\mathbf{c}$ be a list of predicate and function constants. For any interpretation I of σ, $I \models SM[F; \mathbf{c}]$ iff*

- $\langle I, I\rangle \models_{fht} F$, *and*
- *for every interpretation J of σ such that $J <^{\mathbf{c}} I$, we have $\langle J, I\rangle \not\models_{fht} F$.*

4.4 Proofs

4.4.1 Proof of Theorem 1

We will often use the following notation throughout this section. Let σ be a first-order signature, let $\boldsymbol{c}$ be a set of constants that is a subset of σ, and let $\boldsymbol{d}$ be a set of constants not belonging to σ corresponding to $\boldsymbol{c}$. [5] $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ denotes the interpretation of signature $(\sigma \setminus \boldsymbol{c}) \cup \boldsymbol{d}$ obtained from J by replacing every constant from $\boldsymbol{c}$ with the corresponding constant from $\boldsymbol{d}$. For two interpretations I and J of σ that agree on all constants in $\sigma \setminus \boldsymbol{c}$, we define $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$ to be the interpretation from the extended signature $\sigma \cup \boldsymbol{d}$ such that

- $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$ agrees with I on all constants in $\boldsymbol{c}$;
- $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$ agrees with $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ on all constants in $\boldsymbol{d}$;
- $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$ agrees with both I and J on all constants in $\sigma \setminus \boldsymbol{c}$.

Lemma 1 *For any sentence F of signature σ and any interpretations I and J of σ,*

(a) if $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$, then $I \models F$.

(b) if $\langle J, I\rangle \models_{\overline{f}ht} F$, then $\langle I, I\rangle \models_{\overline{f}ht} F$.

Proof. by induction on F. ∎

Lemma 2 *Let F be a sentence of signature σ, and let I and J be interpretations of σ such that $J <^{\mathbf{c}} I$. We have $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $J \models gr_I[F]^{\underline{I}}$.*

[5] That is to say, $\boldsymbol{d}$ and $\boldsymbol{c}$ have the same length and the corresponding members are either predicate constants of the same arity or function constants of the same arity.

Proof. By induction on F.

Case 1: F is an atomic sentence. Then $F^*(\boldsymbol{d})$ is $F(\boldsymbol{d}) \wedge F$, where $F(\boldsymbol{d})$ is obtained from F by replacing the members of $\boldsymbol{c}$ with the corresponding members of $\boldsymbol{d}$. Consider the following subcases:

- *Subcase 1:* $I \not\models F$. Then $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \not\models F^*(\boldsymbol{d})$. Further, $gr_I[F]^{\underline{I}} = \bot$, so $J \not\models gr_I[F]^{\underline{I}}$.

- *Subcase 2:* $I \models F$. Then $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $J^{\mathbf{c}}_{\boldsymbol{d}} \models F(\boldsymbol{d})$ iff $J \models F$. Further, $gr_I[F]^{\underline{I}} = F$, so $J \models gr_I[F]^{\underline{I}}$ iff $J \models F$.

Case 2: F is $G \wedge H$ or $G \vee H$. The claim follows immediately from I.H. on G and H.

Case 3: F is $G \rightarrow H$. Then $F^*(\boldsymbol{d}) = (G^*(\boldsymbol{d}) \rightarrow H^*(\boldsymbol{d})) \wedge (G \rightarrow H)$. Consider the following subcases:

- *Subcase 1:* $I \not\models G \rightarrow H$. Then $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \not\models F^*(\boldsymbol{d})$. Further, $gr_I[F]^{\underline{I}} = \bot$, which J does not satisfy.

- *Subcase 2:* $I \models G \rightarrow H$. Then $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \models G^*(\boldsymbol{d}) \rightarrow H^*(\boldsymbol{d})$. On the other hand, $gr_I[F]^{\underline{I}} = gr_I[G]^{\underline{I}} \rightarrow gr_I[H]^{\underline{I}}$ so this case holds by I.H. on G and H.

Case 4: F is $\exists x G(x)$. By I.H., $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \models G(\xi^\diamond)^*(\boldsymbol{d})$ iff $J \models gr_I[G(\xi^\diamond)]^{\underline{I}}$ for each $\xi \in |I|$. The claim follows immediately.

Case 5: F is $\forall x G(x)$. Similar to Case 4. ∎

Lemma 3 *For any interpretations I and J of signature σ, we have $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \models \boldsymbol{d} < \boldsymbol{c}$ iff $J <^{\mathbf{c}} I$.*

Proof. Recall that by definition, $\boldsymbol{d} < \boldsymbol{c}$ is

$$(\boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}) \wedge \neg(\boldsymbol{d} = \boldsymbol{c}),$$

and by definition, $J <^{\boldsymbol{c}} I$ is

- J and I have the same universe and agree on all constants not in $\boldsymbol{c}$;
- $p^J \subseteq p^I$ for all predicate constants p in $\boldsymbol{c}$; and
- J and I do not agree on $\boldsymbol{c}$.

First, by definition of $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$, J and I have the same universe and agree on all constants in $\sigma \setminus \boldsymbol{c}$.

Second, by definition, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$ iff, for every predicate constant p in $\boldsymbol{c}$,

$$J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models \forall \boldsymbol{x}(p(\boldsymbol{x})^{\boldsymbol{c}}_{\boldsymbol{d}} \rightarrow p(\boldsymbol{x})),$$ [6]

which is equivalent to saying that $(p^{\boldsymbol{c}}_{\boldsymbol{d}})^{J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I} \subseteq p^{J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I}$. Since I does not interpret any constant from $\boldsymbol{d}$, and $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ does not interpret any constant from $\boldsymbol{c}$, this is equivalent to $(p^{\boldsymbol{c}}_{\boldsymbol{d}})^{J^{\boldsymbol{c}}_{\boldsymbol{d}}} \subseteq p^I$ and further to $p^J \subseteq p^I$.

Third, since I does not interpret any constant from $\boldsymbol{d}$ and $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ does not interpret any constant from $\boldsymbol{c}$, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models \neg(\boldsymbol{d} = \boldsymbol{c})$ is equivalent to saying J and I do not agree on $\boldsymbol{c}$. ∎

Theorem 1 *Let F be a first-order sentence of signature σ and $\boldsymbol{c}$ be a list of intensional constants. For any interpretation I of σ, $I \models SM[F; \boldsymbol{c}]$ iff*

- *I satisfies F, and*

[6] $p(\boldsymbol{x})^{\boldsymbol{c}}_{\boldsymbol{d}}$ means the atom that is obtained from $p(\boldsymbol{x})$ by replacing p with the corresponding member of $\boldsymbol{d}$ if $p \in \boldsymbol{c}$, and no change otherwise.

- *every interpretation J such that $J <^{\mathbf{c}} I$ does not satisfy $(gr_I[F])^{\underline{I}}$.*

Proof. $I \models \mathrm{SM}[F; \boldsymbol{c}]$ is by definition

$$I \models F \wedge \neg\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F^*(\widehat{\boldsymbol{c}})). \tag{4.2}$$

The first item, "I satisfies F", is equivalent to the first conjunctive term of (4.2).

By Lemma 16 and Lemma 3, the second item, "no interpretation J of σ such that $J <^{\mathbf{c}} I$ satisfies $gr_I[F]^{\underline{I}}$", is equivalent to the second conjunctive term in (4.2). ∎

4.4.2 Proof of Theorem 2

Lemma 4 *Let F be a sentence of signature σ and let I and J be interpretations of σ such that $J <^{\mathbf{c}} I$. We have $J \models gr_I[F]^{\underline{I}}$ iff $\langle J, I\rangle \models_{fht} F$.*

Proof. By induction on F.

Case 1: F is an atomic sentence. $gr_I[F]$ is F.

- *Subcase 1:* $I \not\models F$. Then $gr_I[F]^{\underline{I}}$ is $\bot$, which J does not satisfy. Further, since $\langle J, I\rangle, t \not\models_{fht} F$, $\langle J, I\rangle \not\models_{fht} F$.

- *Subcase 2:* $I \models F$. Then $gr_I[F]^{\underline{I}}$ is F, and $\langle J, I\rangle, t \models_{fht} F$. It is clear that $J \models F$ iff $\langle J, I\rangle, h \models_{fht} F$.

Case 2: F is $G \wedge H$ or $G \vee H$. The claim follows immediately from I.H. on G and H.

Case 3: F is $G \rightarrow H$. Consider the following subcases:

- *Subcase 1:* $I \not\models G \rightarrow H$. Then $gr_I[G \rightarrow H]^{\underline{I}}$ is $\bot$, which J does not satisfy. Further, $\langle I, I\rangle \not\models_{fht} G \rightarrow H$. By Lemma 1 (b), $\langle J, I\rangle \not\models_{fht} G \rightarrow H$.

- *Subcase 2:* $I \models G \rightarrow H$. Then $gr_I[G \rightarrow H]^{\underline{I}}$ is equivalent to $gr_I[G]^{\underline{I}} \rightarrow gr_I[H]^{\underline{I}}$. Further, $\langle J, I\rangle \models_{fht} G \rightarrow H$ is equivalent to $\langle J, I\rangle \not\models_{fht} G$ or $\langle J, I\rangle \models_{fht} H$. Then the claim follows from I.H. on G and H.

Case 4: F is $\forall x G(x)$, or $\exists x G(x)$. By induction on $G(\xi^\diamond)$ for each ξ in the universe. ∎

Theorem 2 Let F be a first-order sentence of signature σ and $\mathbf{c}$ be a list of predicate and function constants. For any interpretation I of σ, $I \models SM[F; \mathbf{c}]$ iff

- $\langle I, I\rangle \models_{fht} F$, *and*
- *for any interpretation J of σ such that $J <^{\mathbf{c}} I$, we have $\langle J, I\rangle \not\models_{fht} F$.*

Proof. We use Theorem 1 to refer to the reduct-based reformulation and instead show

- I satisfies F, and
- every interpretation J such that $J <^{\mathbf{c}} I$ does not satisfy $(gr_I[F])^{\underline{I}}$.

iff

- $\langle I, I\rangle \models_{fht} F$, and
- for any interpretation J of σ such that $J <^{\mathbf{c}} I$, we have $\langle J, I\rangle \not\models_{fht} F$.

Clearly, $I \models F$ iff $\langle I, I\rangle \models_{fht} F$. By Lemma 4, for every interpretation J such that $J <^{\mathbf{c}} I$, we have $J \not\models (gr_I[F])^{\underline{I}}$ iff $\langle J, I\rangle \not\models_{fht} F$. ∎

Chapter 5

PROPERTIES OF THE FUNCTIONAL STABLE MODEL SEMANTICS

5.1 Constraints

Following Ferraris *et al.* (2009), we say that an occurrence of a constant, or any other subexpression, in a formula F is *positive* if the number of implications containing that occurrence in the antecedent is even, and *negative* otherwise. We say that the occurrence is *strictly positive* if the number of implications in F containing that occurrence in the antecedent is 0. For example, in $\neg(f = 1) \rightarrow g = 1$, the occurrences of f and g are both positive [1] , but only the occurrence of g is strictly positive.

We say that a formula F is *negative* on a list $\boldsymbol{c}$ of predicate and function constants if members of $\boldsymbol{c}$ have no strictly positive occurrences in F. We say that F is a *constraint* if it has no strictly positive occurrences of any constant. Clearly, a constraint is negative on any list of constants. For instance, a formula of the form $\neg H$ is a constraint.

Theorem 3 *For any first-order formulas F and G, if G is negative on $\boldsymbol{c}$, $\mathrm{SM}[F \wedge G; \boldsymbol{c}]$ is equivalent to $\mathrm{SM}[F; \boldsymbol{c}] \wedge G$.*

Example 10 *Consider a formula F*

$$(f = 1 \vee g = 1) \wedge (f = 2 \vee g = 2)$$

whose stable models are $\{f = 1, g = 2\}$ and $\{f = 2, g = 1\}$. Now, to find the stable models of $F \wedge \neg(f = 1)$, we observe that since $\neg(f = 1)$ is negative on $\{f, g\}$,

[1] Recall that $\neg(f = 1)$ is an abbreviation for $f = 1 \rightarrow \bot$.

according to Theorem 3, $SM[F \wedge \neg(f=1); fg]$ *is equivalent to* $SM[F; fg] \wedge \neg(f=1)$*, which leaves only* $\{f=2, g=1\}$ *as a stable model.*

5.2 Choice and Defaults

Similar to Theorem 2 from Ferraris *et al.* (2011), the theorem below shows that making the set of intensional constants smaller can only make the result of applying SM weaker, and that this can be compensated by adding "choice formulas." For any predicate constant p, by $Choice(p)$ we denote the formula $\forall \boldsymbol{x}(p(\boldsymbol{x}) \vee \neg p(\boldsymbol{x}))$, where $\boldsymbol{x}$ is a list of distinct object variables. For any function constant f, by $Choice(f)$ we denote the formula $\forall \boldsymbol{x}y((f(\boldsymbol{x})=y) \vee \neg(f(\boldsymbol{x})=y))$, where y is an object variable that is distinct from $\boldsymbol{x}$. For any finite list of predicate and function constants $\boldsymbol{c}$, $Choice(\boldsymbol{c})$ stands for the conjunction of the formulas $Choice(c)$ for all members c of $\boldsymbol{c}$.

Theorem 4 *For any first-order formula* F *and any disjoint lists* $\boldsymbol{c}$*,* $\boldsymbol{d}$ *of distinct constants, the following formulas are logically valid:*

$$SM[F; \boldsymbol{cd}] \rightarrow SM[F; \boldsymbol{c}],$$
$$SM[F \wedge Choice(\boldsymbol{d}); \boldsymbol{cd}] \leftrightarrow SM[F; \boldsymbol{c}].$$

For example, the formula $g=1 \rightarrow f=1$ has only one f-stable model–$\{f=1, g=1\}$. By Theorem 4, $\mathrm{SM}[g=1 \rightarrow f=1;\ f]$ is equivalent to

$$\mathrm{SM}[(g=1 \rightarrow f=1) \wedge Choice(g);\ fg]$$

or rather

$$\mathrm{SM}[(g=1 \rightarrow f=1) \wedge \forall y(g=y \vee \neg(g=y));\ fg]$$

which has only $\{f = 1, g = 1\}$ as a model. This allows capturing the notion of the $\boldsymbol{c}$-stable models without having to refer to the list of intensional constants, instead encoding this notion in the formula directly.

5.3 Strong Equivalence

Strong equivalence Lifschitz *et al.* (2001) is an important notion that allows us to substitute one subformula for another subformula without affecting the stable models. The theorem on strong equivalence can be extended to formulas with intensional functions as follows.

About first-order formulas F and G we say that F is *strongly equivalent* to G if, for any formula H, any occurrence of F in H, and any list $\boldsymbol{c}$ of distinct predicate and function constants, $\mathrm{SM}[H; \boldsymbol{c}]$ is equivalent to $\mathrm{SM}[H'; \boldsymbol{c}]$, where H' is obtained from H by replacing the occurrence of F by G. In this definition, H is allowed to contain function and predicate constants that do not occur in F, G; Theorem 5 below shows, however, that this is not essential.

Theorem 5 *Let F and G be first-order formulas, let $\mathbf{c}$ be the list of all constants occurring in F or G and let $\widehat{\mathbf{c}}$ be a list of distinct predicate and function variables corresponding to $\mathbf{c}$. The following conditions are equivalent to each other.*

- *F and G are strongly equivalent to each other;*
- *Formula*

$$(F \leftrightarrow G) \land (\widehat{\boldsymbol{c}} < \boldsymbol{c} \rightarrow (F^*(\widehat{\boldsymbol{c}}) \leftrightarrow G^*(\widehat{\boldsymbol{c}})))$$

  is logically valid.

According to the theorem, formula $\{F\}$ (shorthand for $F \vee \neg F$) is strongly equivalent to $\neg\neg F \rightarrow F$. This allows us to rewrite the formula representing inertia in Example 7

$$(amount_1 = Y) \vee \neg(amount_1 = Y) \ \leftarrow \ amount_0 = Y+1$$

as

$$(amount_1 = Y) \leftarrow \neg\neg(amount_1 = Y) \ \wedge \ amount_0 = Y+1.$$

This is useful for putting formulas in a standard form called Clark Normal Form, which is necessary for extending the Theorem on Completion from Ferraris *et al.* (2011) to our semantics. This is discussed in detail later in Section 5.5.

5.4 Splitting Theorem

For more complex formulas, it would be convenient to break the formula into separate smaller formulas for readability, modularity, and even efficiency of computation. However, arbitrarily breaking up a formula does not necessarily result in stable models that can then be composed to obtain the stable models of the original formula.

Example 11 *We will consider two possibilities:*

- *Simply taking the stable models with respect to all function constants in the signature common to both of the smaller formulas, and*
- *Taking the common stable models with respect to only function constants in the signature that appear in the head of a rule common.*

Consider the formula F that is

$$(f = 1 \leftarrow g = 1) \wedge g = 1$$

which has one stable model: $\{f = 1, g = 1\}$*. However, if we utilize the first option and break the formula into*

$$(f = 1 \leftarrow g = 1)$$

and

$$g = 1$$

The former has no stable models w.r.t. f, g *and so there are no common stable models. This demonstrates that the first method is incorrect.*

Consider the formula G *that is*

$$(f = 1 \leftarrow g = 1) \wedge (g = 1 \leftarrow f = 1)$$

which has no stable models w.r.t. f, g*. However, if we utilize the second option and break the formula into*

$$(f = 1 \leftarrow g = 1)$$

and

$$(g = 1 \leftarrow f = 1)$$

we obtain a common stable model (w.r.t. f *for the first and* g *for the second):* $\{f = 1, g = 1\}$*. This illustrates that the second method is incorrect.*

The second approach described in the example does work for the first example where there is no cyclic dependency among the function constants. In fact, it is precisely this notion that we will formally capture to ensure that the second approach will allow splitting the formula.

Definition 2 *Let* f *be a function constant. A first-order formula is called* f-plain [2] *if each atomic formula*

[2] The notion of f-plain comes from Lifschitz and Yang (2011).

* *does not contain f, or*

* *is of the form $f(\boldsymbol{t}) = u$ where $\boldsymbol{t}$ is a tuple of terms not containing f, and u is a term not containing f.*

For example, $f=1$ is f-plain, but each of $p(f)$, $g(f) = 1$, and $1=f$ are not f-plain.

For a list $\boldsymbol{c}$ of predicate and function constants, we say that F is $\boldsymbol{c}$-plain if F is f-plain for each function constant f in $\boldsymbol{c}$. Roughly speaking, $\boldsymbol{c}$-plain formulas do not allow the functions in $\boldsymbol{c}$ to be nested in another predicate or function, and at most one function in $\boldsymbol{c}$ is allowed in each atomic formula. For example, $f = g$ is not (f, g)-plain, and neither is $f(g) = 1 \rightarrow g = 1$.

A *rule* of a first-order formula F is a strictly positive occurrence of an implication in F.

Let F be a $\boldsymbol{c}$-plain formula. The *dependency graph* of F (relative to $\boldsymbol{c}$), denoted by $\mathrm{DG}_{\boldsymbol{c}}[F]$, is the directed graph that

* has all members of $\boldsymbol{c}$ as its vertices, and

* has an edge from c to d if, for some rule $G \rightarrow H$ of F,

  - c has a strictly positive occurrence in H, and

  - d has a positive occurrence in G that does not belong to any subformula of G that is negative on $\boldsymbol{c}$.

A *loop* of F (relative to a list $\boldsymbol{c}$ of intensional predicates) is a nonempty subset $\boldsymbol{l}$ of $\boldsymbol{c}$ such that the subgraph of $\mathrm{DG}_{\boldsymbol{c}}[F]$ induced by $\boldsymbol{l}$ is strongly connected.

The following theorem extends the Splitting Lemma from Ferraris *et al.* (2009) to allow intensional functions.

Theorem 6 *Let F be a $\boldsymbol{c}$-plain formula, and let $\boldsymbol{c}$ be a list of constants. If $\boldsymbol{l}^1, \dots, \boldsymbol{l}^n$ are all the loops of F relative to $\boldsymbol{c}$ then*

$$SM[F; \boldsymbol{c}] \quad \text{is equivalent to} \quad SM[F; \boldsymbol{l}^1] \wedge \cdots \wedge SM[F; \boldsymbol{l}^n].$$

The following theorem extends the splitting theorem from Ferraris *et al.* (2009) to allow intensional functions.

Theorem 7 *Let $\boldsymbol{c}$, $\boldsymbol{d}$ be finite disjoint lists of distinct constants and let F, G be $\boldsymbol{cd}$-plain first-order sentences. If*

(a) each strongly connected component of the dependency graph of $F \wedge G$ relative to $\boldsymbol{c}$, $\boldsymbol{d}$ is either a subset of $\boldsymbol{c}$ or a subset of $\boldsymbol{d}$,

(b) F is negative on $\boldsymbol{d}$, and

(c) G is negative on $\boldsymbol{c}$

then

$$SM[F \wedge G;\ \boldsymbol{c} \cup \boldsymbol{d}] \leftrightarrow SM[F;\ \boldsymbol{c}] \wedge SM[G;\ \boldsymbol{d}]$$

is logically valid.

It is clear that Theorem 3 is a special case of Theorem 7, when $\boldsymbol{d}$ is empty.

5.5 Completion

In Section 4.2, we presented a characterization of the functional stable model semantics in terms of second order logic. However, this does not provide much clarity of the relationship between classical logic and this formalism. As mentioned earlier, simple axioms from classical logic such as $F \vee \neg F$ being a tautology do not necessarily hold in the functional stable model semantics so it is natural to consider a formal relationship between this semantics and classical logic. This section extends the Theorem on Completion from Ferraris *et al.* (2011) to the functional stable models semantics, providing a method to capture a class of formulas under our semantics in classical logic.

We say that a formula F is in *Clark normal form* (relative to the list $\boldsymbol{c}$ of intensional constants) if it is a conjunction of sentences of the form

$$\forall \boldsymbol{x}(G \rightarrow p(\boldsymbol{x})) \tag{5.1}$$

and

$$\forall \boldsymbol{x} y(G \rightarrow f(\boldsymbol{x})=y) \tag{5.2}$$

one for each intensional predicate p and each intensional function f, where $\boldsymbol{x}$ is a list of distinct object variables, y is an object variable, and G is an arbitrary formula that has no free variables other than those in $\boldsymbol{x}$ and y.

The *completion* of a formula F in Clark normal form (relative to $\boldsymbol{c}$) is obtained from F by replacing each conjunctive term (5.1) with

$$\forall \boldsymbol{x}(p(\boldsymbol{x}) \leftrightarrow G)$$

and each conjunctive term (5.2) with

$$\forall \boldsymbol{x} y(f(\boldsymbol{x})=y \leftrightarrow G).$$

An occurrence of a symbol or a subformula in a formula F is called *strictly positive* in F if that occurrence is not in the antecedent of any implication in F. The *t-dependency graph* of F (relative to $\boldsymbol{c}$) is the directed graph that

- has all members of $\boldsymbol{c}$ as its vertices, and
- has an edge from c to d if, for some strictly positive occurrence of $G \to H$ in F,
  - c has a strictly positive occurrence in H, and
  - d has a strictly positive occurrence in G.

We say that F is *tight* (on $\boldsymbol{c}$) if the t-dependency graph of F (relative to $\boldsymbol{c}$) is acyclic. For example,

$$((p \to q) \to r) \to p$$

is tight on $\{p, q, r\}$ because its t-dependency graph has only one edge, which goes from p to r. On the other hand, the formula is not tight according to Ferraris *et al.* (2011) because, according to the definition of a dependency graph in that paper, there is an additional edge that goes from p to itself.

The following theorem is similar to the main theorem of Lifschitz and Yang (2013), which describes functional completion in nonmonotonic causal logic. Due to our more general definition of tightness, this theorem generalizes the Theorem on Completion in Ferraris *et al.* (2011) even when only predicates are allowed to be intensional.

Theorem 8 *For any formula F in Clark normal form that is tight on $\boldsymbol{c}$, an interpretation I that satisfies $\exists xy(x \neq y)$ is a model of $\mathrm{SM}[F; \boldsymbol{c}]$ iff I is a model of the completion of F relative to $\boldsymbol{c}$.*

Example 1 continued

We can represent the factors that affect the speed of the car at time point 1 (the *full description of this domain will be shown in 9.2.2) as*

$$\begin{aligned}
\mathit{speed}(1)\!=\!Y \;\leftarrow\;\; & \mathit{accel}(0)\!=\!\boldsymbol{t} \;\wedge\; \mathit{speed}(0)\!=\!X \;\wedge\; \mathit{duration}(0)\!=\!D \\
& \wedge\, (Y = X + a \times D) \\
\mathit{speed}(1)\!=\!Y \;\leftarrow\;\; & \mathit{decel}(0)\!=\!\boldsymbol{t} \;\wedge\; \mathit{speed}(0)\!=\!X \;\wedge\; \mathit{duration}(0)\!=\!D \\
& \wedge\, (Y = X - a \times D) \\
\mathit{speed}(1)\!=\!Y \;\leftarrow\;\; & \mathit{speed}(0)\!=\!Y \;\wedge\; \neg\neg(\mathit{speed}(1)\!=\!Y)
\end{aligned}$$

(recall that $c\!=\!v \vee \neg(c\!=\!v) \leftarrow G$ is strongly equivalent to $c\!=\!v \leftarrow G \wedge \neg\neg(c\!=\!v)$) and the completion with respect to the function $\mathit{speed}(1)$ will be the following equivalence.

$$\begin{aligned}
\mathit{speed}(1) = Y \;\leftrightarrow\; \exists XD(\quad & (\mathit{accel}(0)\!=\!\boldsymbol{t} \;\wedge\; \mathit{speed}(0)\!=\!X \;\wedge\; \mathit{duration}(0)\!=\!D \\
& \quad \wedge\; (Y = X + a \times D)) \\
& \vee\, (\mathit{decel}(0)\!=\!\boldsymbol{t} \;\wedge\; \mathit{speed}(0)\!=\!X \;\wedge\; \mathit{duration}(0)\!=\!D \\
& \quad \wedge\; (Y = X - a \times D)) \\
& \vee\, (\mathit{speed}(0) = Y)\;)
\end{aligned}$$

The assumption $\exists xy(x \neq y)$ in the statement of Theorem 8 is essential. For instance, take F to be $\top$ and $\boldsymbol{c}$ to be an intensional function constant f. If the universe $|I|$ of an interpretation I is a singleton, then I satisfies SM$[F]$, but does not satisfy the completion of F, which is $\forall \boldsymbol{x}y(f(\boldsymbol{x}) = y \leftrightarrow \bot)$.

5.6 Proofs

5.6.1 Proof of Theorem 3

Lemma 5 *The formula*

$$(\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge F^*(\widehat{\boldsymbol{c}}) \rightarrow F$$

is logically valid.

Proof: by induction on F. ∎

Lemma 6 *Formula*

$$\widehat{\boldsymbol{c}} < \boldsymbol{c} \rightarrow ((\neg F)^*(\widehat{\boldsymbol{c}}) \leftrightarrow \neg F)$$

is logically valid.

Proof: immediate from Lemma 5. ∎

Theorem 3 *For any first-order formulas F and G, if G is negative on $\boldsymbol{c}$, SM$[F \wedge G; \boldsymbol{c}]$ is equivalent to SM$[F; \boldsymbol{c}] \wedge G$.*

Proof. By Lemma 6,

$$\begin{aligned}
&\mathrm{SM}[F \wedge \neg G; \boldsymbol{c}] \\
&= F \wedge \neg G \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge (F \wedge \neg G)^*(\widehat{\boldsymbol{c}})) \\
&\Leftrightarrow F \wedge \neg G \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge F^*(\widehat{\boldsymbol{c}}) \wedge \neg G) \\
&\Leftrightarrow F \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge F^*(\widehat{\boldsymbol{c}})) \wedge \neg G \\
&= \mathrm{SM}[F; \boldsymbol{c}] \wedge \neg G.
\end{aligned}$$

∎

Lemma 7 *Choice*$(\boldsymbol{c})^*(\widehat{\boldsymbol{c}})$ *is equivalent to*

$$(\boldsymbol{c}^{pred} \leq \widehat{\boldsymbol{c}}^{pred}) \wedge (\boldsymbol{c}^{func} = \widehat{\boldsymbol{c}}^{func}).$$

Proof. *Choice*$(\boldsymbol{c})$ is the conjunction for each predicate p in $\boldsymbol{c}^{pred}$ of $\forall \boldsymbol{x}(p(\boldsymbol{x}) \vee \neg p(\boldsymbol{x}))$ and for each function f in $\boldsymbol{c}^{func}$ of $\forall \boldsymbol{x} y(f(\boldsymbol{x}) = y \vee \neg f(\boldsymbol{x}) = y)$.

Now,

$$[\forall \boldsymbol{x}(p(\boldsymbol{x}) \vee \neg p(\boldsymbol{x}))]^*(\widehat{\boldsymbol{c}})$$

is equivalent to

$$\forall \boldsymbol{x}(\widehat{p}(\boldsymbol{x}) \vee \neg p(\boldsymbol{x})),$$

which is further equivalent to

$$\forall \boldsymbol{x}(p(\boldsymbol{x}) \rightarrow \widehat{p}(\boldsymbol{x})),$$

or simply $p \leq \widehat{p}$.

Next,

$$[\forall \boldsymbol{x} y(f(\boldsymbol{x}) = y \vee \neg(f(\boldsymbol{x}) = y))]^*(\widehat{\boldsymbol{c}})$$

is equivalent to

$$\forall \boldsymbol{x} y((f(\boldsymbol{x}) = y \wedge (\widehat{f}(\boldsymbol{x}) = y)) \vee \neg(f(\boldsymbol{x}) = y)),$$

which is further equivalent to

$$\forall \boldsymbol{x} y(f(\boldsymbol{x}) = y \rightarrow \widehat{f}(\boldsymbol{x}) = y),$$

or simply $f = \widehat{f}$.

Thus, *Choice*$(\boldsymbol{c})^*(\widehat{\boldsymbol{c}})$ is the conjunction for each predicate p in $\boldsymbol{c}^{pred}$ of $p \leq \widehat{p}$ and for each function f in $\boldsymbol{c}^{func}$ of $f = \widehat{f}$, or simply *Choice*$(\boldsymbol{c})^*(\boldsymbol{c})$ is

$$(\boldsymbol{c}^{pred} \leq \widehat{\boldsymbol{c}}^{pred}) \wedge (\boldsymbol{c}^{func} = \widehat{\boldsymbol{c}}^{func}). \quad \blacksquare$$

5.6.2 Proof of Theorem 4

Theorem 4 *For any first-order formula F and any disjoint lists $\boldsymbol{c}$, $\boldsymbol{d}$ of distinct constants, the following formulas are logically valid:*

$$(i) \quad SM[F; \boldsymbol{cd}] \rightarrow SM[F; \boldsymbol{c}]$$

$$(ii) \quad SM[F \wedge Choice(\boldsymbol{d}); \boldsymbol{cd}] \leftrightarrow SM[F; \boldsymbol{c}].$$

Proof. The proof is not long, but there is a notational difficulty that we need to overcome before we can present it. The notation $F^*(\widehat{\boldsymbol{c}})$ does not take into account the fact that the construction of this formula depends on the choice of the list $\boldsymbol{c}$ of intensional predicates. Since the dependence on $\boldsymbol{c}$ is essential in the proof of Theorem 4, we use here the more elaborate notation $F^{*[\boldsymbol{c}]}(\widehat{\boldsymbol{c}})$. For instance, if F is $p(x) \wedge q(x)$ then

$$\begin{array}{rcl} F^{*[p]}(\widehat{p}) & \text{is} & \widehat{p}(x) \wedge q(x), \\ F^{*[pq]}(\widehat{p}, \widehat{q}) & \text{is} & \widehat{p}(x) \wedge \widehat{q}(x). \end{array}$$

It is easy to verify by induction on F that for any disjoint lists $\boldsymbol{c}$, $\boldsymbol{d}$ of distinct predicate constants,

$$F^{*[\boldsymbol{c}]}(\widehat{\boldsymbol{c}}) \;=\; F^{*[\boldsymbol{cd}]}(\widehat{\boldsymbol{c}}, \boldsymbol{d}). \tag{5.3}$$

(i) In the notation introduced above, SM$[F; \boldsymbol{c}]$ is

$$F \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge F^{*[\boldsymbol{c}]}(\widehat{\boldsymbol{c}})).$$

By (5.3), this formula can be written also as

$$F \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge F^{*[\boldsymbol{cd}]}(\widehat{\boldsymbol{c}}, \boldsymbol{d})),$$

which is equivalent to

$$F \wedge \neg\exists\widehat{\boldsymbol{c}}(((\widehat{\boldsymbol{c}},\boldsymbol{d}) < (\boldsymbol{c},\boldsymbol{d})) \wedge F^{*[\boldsymbol{cd}]}(\widehat{\boldsymbol{c}},\boldsymbol{d})).$$

On the other hand, SM$[F;\boldsymbol{cd}]$ is

$$F \wedge \neg\exists\widehat{\boldsymbol{c}}\widehat{\boldsymbol{d}}(((\widehat{\boldsymbol{c}},\widehat{\boldsymbol{d}}) < (\boldsymbol{c},\boldsymbol{d})) \wedge F^{*[\boldsymbol{cd}]}(\widehat{\boldsymbol{c}},\widehat{\boldsymbol{d}})).$$

To prove (ii), note that, by (5.3) and Lemma 7, the formula

$$\exists\widehat{\boldsymbol{c}}\widehat{\boldsymbol{d}}(((\widehat{\boldsymbol{c}},\widehat{\boldsymbol{d}}) < (c,d)) \wedge F^{*[\boldsymbol{cd}]}(\widehat{\boldsymbol{c}},\widehat{\boldsymbol{d}}) \wedge Choice(\boldsymbol{d})^{*[\boldsymbol{cd}]}(\widehat{\boldsymbol{c}},\widehat{\boldsymbol{d}}))$$

is equivalent to

$$\exists\widehat{\boldsymbol{c}}\widehat{\boldsymbol{d}}(((\widehat{\boldsymbol{c}},\widehat{\boldsymbol{d}}) < (c,d)) \wedge F^{*[\boldsymbol{cd}]}(\widehat{\boldsymbol{c}},\widehat{\boldsymbol{d}}) \wedge (\boldsymbol{d} = \widehat{\boldsymbol{d}})).$$

It follows that it can be also equivalently rewritten as

$$\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge F^{*[\boldsymbol{cd}]}(\widehat{\boldsymbol{c}},\boldsymbol{d})).$$

By (5.3), the last formula can be represented as

$$\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge F^{*[\boldsymbol{c}]}(\widehat{\boldsymbol{c}})).$$

∎

5.6.3 Proof of Theorem 5

Recall that about first-order formulas F and G we say that F is *strongly equivalent* to G if, for any formula H, any occurrence of F in H, and any list $\boldsymbol{c}$ of distinct predicate and function constants, SM$[H;\boldsymbol{c}]$ is equivalent to SM$[H';\boldsymbol{c}]$, where H' is obtained from H by replacing the occurrence of F by G.

Lemma 8 *Formula*

$$(F \leftrightarrow G) \wedge ((F^*(\widehat{\boldsymbol{c}}) \leftrightarrow G^*(\widehat{\boldsymbol{c}})) \rightarrow (H^*(\widehat{\boldsymbol{c}}) \leftrightarrow (H')^*(\widehat{\boldsymbol{c}})))$$

is logically valid.

Proof. By induction on H. ∎

The following lemma is equivalent to the "only if" part of the theorem. In these proofs, we will refer to the following formula

$$(F \leftrightarrow G) \wedge (\widehat{\boldsymbol{c}} < \boldsymbol{c} \rightarrow (F^*(\widehat{\boldsymbol{c}}) \leftrightarrow G^*(\widehat{\boldsymbol{c}}))) \tag{5.4}$$

Lemma 9 *If the formula (5.4) is logically valid, then F is strongly equivalent to G.*

Proof. Assume that (5.4) is logically valid. We need to show that

$$H \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge H^*(\widehat{\boldsymbol{c}})) \tag{5.5}$$

is equivalent to

$$H' \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge (H')^*(\widehat{\boldsymbol{c}})). \tag{5.6}$$

Since (5.4) is logically valid, the first conjunctive term of (5.5) is equivalent to the first conjunctive term of (5.6). By Lemma 8 it also follows that the same relationship holds between the two second conjunctive terms of the same formulas. ∎

Lemma 10 *If F is strongly equivalent to G then (5.4) is logically valid.*

Proof. Let C be the formula $Choice(\boldsymbol{c})$. Let E stand for $F \leftrightarrow G$, and E' be $F \leftrightarrow F$. Since F is strongly equivalent to G, the formula SM$[E \leftrightarrow C]$ is equivalent to SM$[E' \leftrightarrow C]$.

Note that by Lemma 7, $\mathit{Choice}(\boldsymbol{c})^*(\widehat{\boldsymbol{c}})$, which we abbreviate as C^*, is equivalent to

$$(\boldsymbol{c}^{pred} \leq \widehat{\boldsymbol{c}}^{pred}) \wedge (\boldsymbol{c}^{func} = \widehat{\boldsymbol{c}}^{func}).$$

On the other hand, $\widehat{\boldsymbol{c}} < \boldsymbol{c}$ can be equivalently written as

$$(\widehat{\boldsymbol{c}}^{pred} < \boldsymbol{c}^{pred}) \vee ((\widehat{\boldsymbol{c}}^{pred} = \boldsymbol{c}^{pred}) \wedge (\widehat{\boldsymbol{c}}^{func} \neq \boldsymbol{c}^{func})).$$

It follows that

$$\widehat{\boldsymbol{c}} < \boldsymbol{c} \rightarrow (C^* \leftrightarrow \bot)$$

is logically valid.

It is easy to see that $(E \leftrightarrow C)^*$ can be rewritten as

$$E \wedge (E^*(\widehat{\boldsymbol{c}}) \leftrightarrow C^*),$$

and that $E^*(\widehat{\boldsymbol{c}})$ is equivalent to

$$E \wedge (F^*(\widehat{\boldsymbol{c}}) \leftrightarrow G^*(\widehat{\boldsymbol{c}})).$$

Using these two facts and Lemma 5, we can simplify $\mathrm{SM}[E \leftrightarrow C]$ as follows:

$$\begin{aligned}
&\mathrm{SM}[E \leftrightarrow C] \\
&\Leftrightarrow (E \leftrightarrow C) \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge E \wedge (E^*(\widehat{\boldsymbol{c}}) \leftrightarrow C^*)) \\
&\Leftrightarrow E \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge (E^*(\widehat{\boldsymbol{c}}) \leftrightarrow \bot)) \\
&\Leftrightarrow E \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge \neg E^*(\widehat{\boldsymbol{c}})) \\
&\Leftrightarrow E \wedge \neg\exists\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \wedge \neg(F^*(\widehat{\boldsymbol{c}}) \leftrightarrow G^*(\widehat{\boldsymbol{c}}))) \\
&= (F \leftrightarrow G) \wedge \forall\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \rightarrow (F^*(\widehat{\boldsymbol{c}}) \leftrightarrow G^*(\widehat{\boldsymbol{c}})))
\end{aligned}$$

Similarly, $\mathrm{SM}[E' \leftrightarrow C]$ is equivalent to

$$(F \leftrightarrow F) \wedge \forall\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{c}} < \boldsymbol{c}) \rightarrow (F^*(\widehat{\boldsymbol{c}}) \leftrightarrow F^*(\widehat{\boldsymbol{c}}))),$$

which is logically valid. Consequently, (5.4) is logically valid also. ∎

Theorem 5 *Let F and G be first-order formulas, let $\boldsymbol{c}$ be the list of all constants occurring in F or G and let $\widehat{\boldsymbol{c}}$ be a list of distinct predicate/function variables corresponding to $\boldsymbol{c}$ The following conditions are equivalent to each other.*

- *F and G are strongly equivalent to each other;*
- *Formula (5.4) is logically valid.*

Proof.

Immediate from Lemma 9 and Lemma 10. ∎

5.6.4 Proof of Theorem 6

The proof of this theorem uses a reduction from functional SM to predicate SM. This is the topic of Chapter 7 but the necessary terminology and results required for this proof are presented here.

Lemma 11 *Given two lists of predicate and function constants $\boldsymbol{c}$ and $\boldsymbol{d}$ whose elements are in one-to-one correspondence, a formula F of signature $\sigma \supseteq \boldsymbol{c} \cup \{f\}$ that is f-plain, and an interpretation I over a signature $\sigma' \supseteq \sigma \cup \boldsymbol{d} \cup \{p, q, g\}$ that satisfies*

$$\forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y), \tag{5.7}$$

$I \models F^*(g\boldsymbol{d})$ iff $I \models (F^f_p)^*(q\boldsymbol{d})$.

Proof. By induction on F.

Case 1: F is an atomic formula not containing f.

F^f_p is exactly F thus $F^*(g\boldsymbol{d})$ is exactly $(F^f_p)^*(q\boldsymbol{d})$ so certainly the claim holds.

Case 2: F is $f(\boldsymbol{t}) = c$.

$F^*(g\boldsymbol{d})$ is $f(\boldsymbol{t}) = c \wedge g(\boldsymbol{t}) = c$.

F^f_p is $p(\boldsymbol{t}, c)$.

$(F^f_p)^*(q\boldsymbol{d})$ is $q(\boldsymbol{t}, c)$.

Since $I \models (5.7)$, it immediately follows that $I \models F^*(g\boldsymbol{d})$ iff $I \models (F^f_p)^*(q\boldsymbol{d})$.

Case 3: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.

By I.H. on G and H.

Case 4: F is $G \rightarrow H$.

By I.H. on G and H.

Case 5: F is $Q\boldsymbol{x}G(\boldsymbol{x})$ where $Q \in \{\forall, \exists\}$.

By I.H. on G.

Lemma 12 *Given two lists of predicate and function constants* $\mathbf{c}$ *and* $\mathbf{d}$ *whose elements are in one-to-one correspondence, two functions* f *and* g *and an interpretation* I *over a signature* $\sigma' \supseteq \boldsymbol{c} \cup \boldsymbol{d} \cup \{p, q, f, g\}$ *that satisfies*

$$\forall \boldsymbol{x} y(p(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y) \tag{5.8}$$

and

$$\forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y), \tag{5.9}$$

$I \models g\boldsymbol{d} < f\boldsymbol{c}$ *iff* $I \models q\boldsymbol{d} < p\boldsymbol{c}$.

Proof. ($\Rightarrow$) Assume $I \models g\boldsymbol{d} < f\boldsymbol{c}$. By definition, it follows that $I \models (g\boldsymbol{d})^{pred} \leq (f\boldsymbol{c})^{pred}$ and since g and f are not predicates, we have $I \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$. Since we assume $I \models$ (5.9), it follows that $I \models \forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \rightarrow f(\boldsymbol{x}) = y)$. Then from the assumption that $I \models$ (5.8), it follows that $I \models \forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \rightarrow p(\boldsymbol{x}, y))$ or simply that $I \models q \leq p$, from which it follows that $I \models (q\boldsymbol{d})^{pred} \leq (p\boldsymbol{c})^{pred}$.

Now since $I \models g\boldsymbol{d} < f\boldsymbol{c}$, it follows that $I \models \neg(g\boldsymbol{d} = f\boldsymbol{c})$. We consider two cases

- If $I \models \neg(d = c)$ for some corresponding d and c in $\boldsymbol{d}$ and $\boldsymbol{c}$ respectively, then we have $I \models \neg(\boldsymbol{d} = \boldsymbol{c})$ and further, $I \models \neg(q\boldsymbol{d} = p\boldsymbol{c})$.

- Otherwise, it must be that $I \models \neg(g = f)$. That is, for some $\boldsymbol{\xi}$ and ξ, $I \not\models f(\boldsymbol{\xi}) = \xi \leftrightarrow g(\boldsymbol{\xi}) = \xi$. For a given $\boldsymbol{\xi}$, I maps $f(\boldsymbol{\xi})$ to exactly one ξ and similarly for $g(\boldsymbol{\xi})$ and so it follows that $I \not\models f(\boldsymbol{\xi}) = \xi \wedge g(\boldsymbol{\xi}) = \xi$ for every ξ. Since $I \models$ (5.9), $I \not\models q(\boldsymbol{\xi}, \xi)$ for every ξ. However, since $I \models f(\boldsymbol{\xi}) = \xi$ for some ξ, from $I \models$ (5.8), we know $I \models p(\boldsymbol{\xi}, \xi)$ for some ξ. Thus, $I \models \neg(q = p)$ and further $I \models \neg(q\boldsymbol{d} = p\boldsymbol{c})$.

From either case, we then conclude that $I \models q\boldsymbol{d} < p\boldsymbol{c}$.

($\Leftarrow$) Assume $I \models q\boldsymbol{d} < p\boldsymbol{c}$. By definition, it follows that $I \models (q\boldsymbol{d})^{pred} \leq (p\boldsymbol{c})^{pred}$ and further, we have $I \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$. Then, since f and g are not predicates, we have $I \models (g\boldsymbol{d})^{pred} \leq (f\boldsymbol{c})^{pred}$.

Now since $I \models q\boldsymbol{d} < p\boldsymbol{c}$, it follows that $I \models \neg(q\boldsymbol{d} = p\boldsymbol{c})$. We consider two cases

- If $I \models \neg(d = c)$ for some corresponding d and c in $\boldsymbol{d}$ and $\boldsymbol{c}$ respectively, then we have $I \models \neg(\boldsymbol{d} = \boldsymbol{c})$ and further, $I \models \neg(g\boldsymbol{d} = f\boldsymbol{c})$

- Otherwise, it must be that $I \models \neg(q = p)$. That is, for some $\boldsymbol{\xi}$ and ξ, $I \not\models q(\boldsymbol{\xi}, \xi) \leftrightarrow p(\boldsymbol{\xi}, \xi)$. Since $I \models$ (5.8), there is exactly one $\boldsymbol{\xi}$ and ξ such that $I \models p(\boldsymbol{\xi}, \xi)$, which further means that $I \models f(\boldsymbol{\xi}) = \xi$. Thus since $I \models q < p$, it must be that $I \not\models q(\boldsymbol{\xi}, \xi)$, and since $I \models$ (5.9), it follows that $I \not\models g(\boldsymbol{\xi}) = \xi$. Thus, $I \models \neg(g = f)$ and further $I \models \neg(g\boldsymbol{d} = f\boldsymbol{c})$.

From either case, we then conclude that $I \models g\boldsymbol{d} < f\boldsymbol{c}$.

Lemma 13 *For any f-plain formula F,*

$$\forall \boldsymbol{x}y(p(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y) \tag{5.10}$$

and $\exists xy(x \neq y)$ entail

$$\mathit{SM}[F; f\boldsymbol{c}] \leftrightarrow \mathit{SM}[F^f_p; p\boldsymbol{c}].$$

Proof.

For any interpretation $\mathcal{I} = \langle I, X\rangle$ of signature $\sigma \supseteq \{f, p, \boldsymbol{c}\}$ satisfying (5.10), it is clear that $\mathcal{I} \models F$ iff $\mathcal{I} \models F^f_p$ since F^f_p is simply the result of replacing all $f(\boldsymbol{x}) = y$ with $p(\boldsymbol{x}, y)$. Thus it only remains to be shown that $\mathcal{I} \models \neg\exists\widehat{f}\widehat{\boldsymbol{c}}((\widehat{f}\widehat{\boldsymbol{c}} < f\boldsymbol{c}) \wedge F^*(\widehat{f}\widehat{\boldsymbol{c}}))$ iff $\mathcal{I} \models \neg\exists\widehat{p}\widehat{\boldsymbol{c}}((\widehat{p}\widehat{\boldsymbol{c}} < p\boldsymbol{c}) \wedge (F^f_p)^*(\widehat{p}\widehat{\boldsymbol{c}}))$ or equivalently, $\mathcal{I} \models \exists\widehat{f}\widehat{\boldsymbol{c}}((\widehat{f}\widehat{\boldsymbol{c}} < f\boldsymbol{c}) \wedge F^*(\widehat{f}\widehat{\boldsymbol{c}}))$ iff $\mathcal{I} \models \exists\widehat{p}\widehat{\boldsymbol{c}}((\widehat{p}\widehat{\boldsymbol{c}} < p\boldsymbol{c}) \wedge (F^f_p)^*(\widehat{p}\widehat{\boldsymbol{c}}))$.

($\Rightarrow$) Assume $\mathcal{I} \models \exists\widehat{f}\widehat{\boldsymbol{c}}((\widehat{f}\widehat{\boldsymbol{c}} < f\boldsymbol{c}) \wedge F^*(\widehat{f}, \widehat{\boldsymbol{c}}))$. We wish to show that $\mathcal{I} \models \exists\widehat{p}\widehat{\boldsymbol{c}}((\widehat{p}\widehat{\boldsymbol{c}} < p\boldsymbol{c}) \wedge (F^f_p)^*(\widehat{p}\widehat{\boldsymbol{c}}))$

That is, take any function g of the same arity as f and any list of predicates and functions $\boldsymbol{d}$ of the same length $\boldsymbol{c}$. Now let $\mathcal{I}' = \langle I \cup J^{f\boldsymbol{c}}_{g\boldsymbol{d}}, X \cup Y^{\boldsymbol{c}}_{\boldsymbol{d}}\rangle$ be from an extended signature $\sigma' = \sigma \cup \{g, q, \boldsymbol{d}\}$ where J is an interpretation of functions from the signature σ and I and J agree on all symbols not occurring in $\{f, \boldsymbol{c}\}$. $J^{f\boldsymbol{c}}_{g\boldsymbol{d}}$ denotes the interpretation from $\sigma^{f\boldsymbol{c}}_{g\boldsymbol{d}}$ (the signature obtained from σ by replacing f with g and all elements of $\boldsymbol{c}$ with all elements of $\boldsymbol{d}$) obtained from the interpretation J by replacing f with g and the functions in $\boldsymbol{c}$ with the corresponding functions in $\boldsymbol{d}$. Similarly, $Y^{\boldsymbol{c}}_{\boldsymbol{d}}$ is the interpretation from σ' obtained from the interpretation Y by replacing predicates from $\boldsymbol{c}$ by the corresponding predicates from $\boldsymbol{d}$. We assume

$$\mathcal{I}' \models (g\boldsymbol{d} < f\boldsymbol{c} \wedge F^*(g\boldsymbol{d}))$$

and wish to show that there is a predicate q of the same arity as p such that

$$\mathcal{I}' \models (q\boldsymbol{d} < p\boldsymbol{c} \wedge (F^f_p)^*(q\boldsymbol{d})).$$

We define the new predicate q in terms of f and g as follows:

$$q^{\mathcal{I}'}(\vec{\xi}, \xi') = \begin{cases} 1 & \text{if } \mathcal{I}' \models f(\vec{\xi}) = \xi' \wedge g(\vec{\xi}) = \xi' \\ 0 & \text{otherwise.} \end{cases}$$

Clearly $\mathcal{I}' \models \forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y)$ and we assumed $\mathcal{I}' \models (5.10)$ so by Lemma 12, it follows that $\mathcal{I}' \models q\boldsymbol{d} < p\boldsymbol{c}$. By Lemma 11, it follows that $\mathcal{I}' \models (F^f_p)^*(q\boldsymbol{d})$.

($\Leftarrow$) Assume $\mathcal{I} \models \exists \widehat{p}\widehat{\boldsymbol{c}}((\widehat{p}\widehat{\boldsymbol{c}} < p\boldsymbol{c}) \wedge (F^f_p)^*(\widehat{p}\widehat{\boldsymbol{c}}))$. We wish to show that $\mathcal{I} \models \exists \widehat{f}\widehat{\boldsymbol{c}}((\widehat{f}\widehat{\boldsymbol{c}} < f\boldsymbol{c}) \wedge F^*(\widehat{f}\widehat{\boldsymbol{c}}))$

That is, take any predicate q of the same arity as p and any list of predicates and functions $\boldsymbol{d}$ the same length as $\boldsymbol{c}$ and let $\mathcal{I}' = \langle I \cup J^{f\boldsymbol{c}}_{g\boldsymbol{d}}, X \cup Y^{\boldsymbol{c}}_{\boldsymbol{d}} \rangle$ is defined as before. We assume

$$\mathcal{I}' \models (q\boldsymbol{d} < p\boldsymbol{c} \wedge (F^f_p)^*(q\boldsymbol{d}))$$

and wish to show that there is a function g of the same arity as f such that

$$\mathcal{I}' \models (g\boldsymbol{d} < f\boldsymbol{c} \wedge F^*(g\boldsymbol{d})).$$

We define the new function g in terms of p and q as follows:

$$g^{\mathcal{I}'}(\vec{\xi}) = \begin{cases} \xi' & \text{if } \mathcal{I}' \models p(\vec{\xi}, \xi') \wedge q(\vec{\xi}, \xi') \\ \xi'' & \text{if } \mathcal{I}' \models p(\vec{\xi}, \xi') \wedge \neg q(\vec{\xi}, \xi') \text{ where } \xi' \neq \xi''. \end{cases}$$

Note that the assumption that there are at least two elements in the universe is essential to this definition. This is a well-defined function by (5.10) entailing $\forall \vec{\xi} \exists \xi' (p(\vec{\xi}, \xi'))$.

We show that $\mathcal{I}' \models \forall \boldsymbol{x} y (q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y)$. Since we assume $\mathcal{I}' \models$ (5.10), it follows that for any given $\boldsymbol{\xi}$, there is only one ξ such that $\mathcal{I}' \models p(\boldsymbol{\xi}, \xi)$. Then, since we assume $\mathcal{I}' \models q \leq p$, we know $\mathcal{I}' \not\models q(\boldsymbol{\xi}, \xi')$ for any $\xi' \neq \xi$. If $\mathcal{I}' \models q(\boldsymbol{\xi}, \xi)$, then $\mathcal{I}' \models g(\boldsymbol{\xi}) = \xi$. Otherwise, $\mathcal{I}' \models g(\boldsymbol{\xi}) = \xi'$ for some $\xi' \neq \xi$. Since this is true for any $\boldsymbol{\xi}$, it follows that $\mathcal{I}' \models \forall \boldsymbol{x} y (q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y)$.

We assumed $\mathcal{I}' \models$ (5.10) so by Lemma 12, it follows that $\mathcal{I}' \models q\boldsymbol{d} < p\boldsymbol{c}$. By Lemma 11, it follows that $\mathcal{I}' \models (F_p^f)^*(q\boldsymbol{d})$.

Lemma 14 *Let F be an f-plain sentence. (a) An interpretation I of the signature of F that satisfies $\exists xy(x \neq y)$ is a model of SM$[F; f\boldsymbol{c}]$ iff I_p^f is a model of SM$[F_p^f; p\boldsymbol{c}]$. (b) An interpretation J of the signature of F_p^f that satisfies $\exists xy(x \neq y)$ is a model of SM$[F_p^f \wedge UEC_p; p\boldsymbol{c}]$ iff $J = I_p^f$ for some model I of SM$[F; f\boldsymbol{c}]$.*

Proof.

For two interpretations I of signature σ_1 and J of signature σ_2, by $I \cup J$ we denote the interpretation of signature $\sigma_1 \cup \sigma_2$ and universe $|I| \cup |J|$ that interprets all symbols occurring only in σ_1 in the same way I does and similarly for σ_2 and J. For symbols appearing in both σ_1 and σ_2, I must interpret these the same as J does, in which case $I \cup J$ also interprets the symbol in this way.

(a$\Rightarrow$) Assume $I \models \text{SM}[F; f\boldsymbol{c}] \wedge \exists xy(x \neq y)$. Since $I \models \exists xy(x \neq y)$, $I \cup I_p^f \models \exists xy(x \neq$

y) since by definition of I_p^f, I and I_p^f share the same universe. By definition of I_p^f, $I \cup I_p^f \models$ (5.10). Thus by Lemma 13, $I \cup I_p^f \models \mathrm{SM}[F; f\boldsymbol{c}] \leftrightarrow \mathrm{SM}[F_p^f; p\boldsymbol{c}]$.

(a$\Leftarrow$) Assume $I \models \exists xy(x \neq y)$ and $I_p^f \models \mathrm{SM}[F_p^f; p\boldsymbol{c}]$. Since $I \models \exists xy(x \neq y)$, $I \cup I_p^f \models \exists xy(x \neq y)$ since by definition of I_p^f, I and I_p^f share the same universe. By definition of I_p^f, $I \cup I_p^f \models$ (5.10). Thus by Lemma 13, $I \cup I_p^f \models \mathrm{SM}[F; f\boldsymbol{c}] \leftrightarrow \mathrm{SM}[F_p^f; p\boldsymbol{c}]$.

Since we assume $I_p^f \models \mathrm{SM}[F_p^f; p\boldsymbol{c}]$, it is the case that $I \cup I_p^f \models \mathrm{SM}[F_p^f; p\boldsymbol{c}]$ and thus it must be the case that $I \cup I_p^f \models \mathrm{SM}[F; f\boldsymbol{c}]$. Therefore since the signature of I_p^f does contain f, we conclude $I \models \mathrm{SM}[F; f\boldsymbol{c}]$.

(b$\Rightarrow$) Assume $J \models \exists xy(x \neq y)$ and $J \models \mathrm{SM}[F_p^f \wedge UEC_p; p\boldsymbol{c}]$. Let $I = J_f^p$ where J_f^p denotes the interpretation of the signature of F obtained from J by replacing the set p^J with the function f such that $f^I(\xi_1, \ldots, \xi_k) = \xi_{k+1}$ for all tuples $\langle \xi_1, \ldots, \xi_k, \xi_{k+1} \rangle$ in p^J. This is a valid definition of a function since we assume $J \models \mathrm{SM}[F_p^f \wedge UEC_p; p\boldsymbol{c}]$, from which we obtain by Theoreom 3 that $J \models \mathrm{SM}[F_p^f; p\boldsymbol{c}] \wedge UEC_p$ and specifically, $J \models UEC_p$. Clearly, $J = I_p^f$ so it only remains to be shown that $I \models \mathrm{SM}[F; f\boldsymbol{c}]$.

Since I and J have the same universe and $J \models \exists xy(x \neq y)$, it follows that $I \cup J \models \exists xy(x \neq y)$. Also by the definition of J_f^p $I \cup J \models$ (5.10). Thus by Lemma 13, $I \cup J \models \mathrm{SM}[F; f\boldsymbol{c}] \leftrightarrow \mathrm{SM}[F_p^f; p\boldsymbol{c}]$.

Since we assume $J \models \mathrm{SM}[F_p^f; p\boldsymbol{c}]$, it is the case that $I \cup J \models \mathrm{SM}[F_p^f; p\boldsymbol{c}]$ and thus it must be the case that $I \cup J \models \mathrm{SM}[F; f\boldsymbol{c}]$. Now since the signature of J does not contain f, we conclude $I \models \mathrm{SM}[F; f\boldsymbol{c}]$.

(b$\Leftarrow$)Take any I such that $J = I_p^f$ and $I \models \mathrm{SM}[F; f\boldsymbol{c}]$. Since $J \models \exists xy(x \neq y)$ and I and J share the same universe, $I \cup J \models \exists xy(x \neq y)$. By definition of $J = I_p^f$,

$I \cup J \models$ (5.10). Thus by Lemma 13, $I \cup J \models \mathrm{SM}[F; f\boldsymbol{c}] \leftrightarrow \mathrm{SM}[F_p^f; p\boldsymbol{c}]$.

Since we assume $I \models \mathrm{SM}[F; f\boldsymbol{c}]$, it is the case that $I \cup J \models \mathrm{SM}[F; f\boldsymbol{c}]$ and thus it must be the case that $I \cup J \models \mathrm{SM}[F_p^f; p\boldsymbol{c}]$. Further, due to the nature of functions, (5.10) entails UEC_p so $I \cup J \models UEC_p$. However since the signature of I does not contain p, we conclude $J \models \mathrm{SM}[F_p^f; p\boldsymbol{c}] \wedge UEC_p$ and since UEC_p is comprised of constraints, by Theorem 3 $J \models \mathrm{SM}[F_p^f \wedge UEC_p; p\boldsymbol{c}]$. ∎

Theorem 6 *Let F be a $\boldsymbol{c}$-plain formula, and let $\boldsymbol{c}$ be a list of constants. If $\boldsymbol{l}^1, \ldots, \boldsymbol{l}^n$ are all the loops of F relative to $\boldsymbol{c}$ then*

$$SM[F; \boldsymbol{c}] \quad \textit{is equivalent to} \quad SM[F; \boldsymbol{l}^1] \wedge \cdots \wedge SM[F; \boldsymbol{l}^n].$$

Proof.

The proof is by reduction to predicate SM.

By repeated applications of Lemma 14 we can obtain $F_{\boldsymbol{p}}^{\boldsymbol{f}} \wedge UEC_{\boldsymbol{p}}$ where $\boldsymbol{f}$ are all the functions in $\boldsymbol{c}$ and $\boldsymbol{p}$ is a list of new predicates and the stable models of $F_{\boldsymbol{p}}^{\boldsymbol{f}} \wedge UEC_{\boldsymbol{p}}$ will coincide with the stable models of F.

The dependency graph of F is isomorphic to the dependency graph of $F_{\boldsymbol{p}}^{\boldsymbol{f}} \wedge UEC_{\boldsymbol{p}}$ by the obvious isomorphism that maps p to p for each predicate $p \in \boldsymbol{c}$ and maps f to q for each function $f \in \boldsymbol{f}$ and the corresponding $q \in \boldsymbol{p}$.

Then the claims follow by Splitting Lemma, Version 1 in Ferraris *et al.* (2009). ∎

Theorem 7 *Let $\boldsymbol{c}$, $\boldsymbol{d}$ be finite disjoint lists of distinct constants and let F, G be $\boldsymbol{cd}$-plain first-order sentences. If*

(a) each strongly connected component of the predicate dependency graph of $F \wedge G$ relative to $\boldsymbol{c}$, $\boldsymbol{d}$ is either a subset of $\boldsymbol{c}$ or a subset of $\boldsymbol{d}$,

(b) F is negative on $\boldsymbol{d}$, and

(c) G is negative on $\boldsymbol{c}$

then

$$SM[F \wedge G;\ \boldsymbol{c} \cup \boldsymbol{d}] \leftrightarrow SM[F;\ \boldsymbol{c}] \wedge SM[G;\ \boldsymbol{d}]$$

is logically valid.

Proof.

The proof is by reduction to predicate SM.

By repeated applications of Lemma 14, we can obtain $(F \wedge G)^{\boldsymbol{fg}}_{\boldsymbol{pq}} \wedge UEC_{\boldsymbol{pq}}$ where $\boldsymbol{f}$ are all the functions in $\boldsymbol{c}$ and $\boldsymbol{p}$ is a list of new predicates of the same length and $\boldsymbol{g}$ are all the functions in $\boldsymbol{d}$ and $\boldsymbol{q}$ is a list of new predicates of the same length. The stable models of $(F \wedge G)^{\boldsymbol{fg}}_{\boldsymbol{pq}} \wedge UEC_{\boldsymbol{pq}}$ will coincide with the stable models of $F \wedge G$.

Similarly, by repeated applications of Lemma 14, we can obtain $F^{\boldsymbol{fg}}_{\boldsymbol{pq}} \wedge UEC_{\boldsymbol{pq}}$ whose stable models will coincide with those of F. Again, by repeated applications of Lemma 14, we can obtain $G^{\boldsymbol{fg}}_{\boldsymbol{pq}} \wedge UEC_{\boldsymbol{pq}}$ whose stable models will coincide with those of G.

The dependency graph of $F \wedge G$ is isomorphic to the dependency graph of $(F \wedge G)^{\boldsymbol{fg}}_{\boldsymbol{pq}} \wedge UEC_{\boldsymbol{pq}}$ by the obvious isomorphism that maps p to p for each predicate $p \in \boldsymbol{c} \cup \boldsymbol{d}$ and maps f to q for each function $f \in \boldsymbol{f} \cup \boldsymbol{g}$ and the corresponding

$q \in \boldsymbol{p} \cup \boldsymbol{q}$. Similar for the dependency graphs of F and $F^{\boldsymbol{fg}}_{\boldsymbol{pq}} \wedge UEC_{\boldsymbol{pq}}$ as well as those of G and $G^{\boldsymbol{fg}}_{\boldsymbol{pq}} \wedge UEC_{\boldsymbol{pq}}$. In particular, we note that since $UEC_{\boldsymbol{pq}}$ is negative on $\boldsymbol{pq}$, we have that $F^{\boldsymbol{fg}}_{\boldsymbol{pq}} \wedge UEC_{\boldsymbol{pq}}$ is negative on $\boldsymbol{d}^{\boldsymbol{g}}_{\boldsymbol{q}}$ and that $G^{\boldsymbol{fg}}_{\boldsymbol{pq}} \wedge UEC_{\boldsymbol{pq}}$ is negative on $\boldsymbol{c}^{\boldsymbol{f}}_{\boldsymbol{p}}$

Then the claims follow by Splitting Lemma, Version 2 in Ferraris *et al.* (2009). ∎

5.6.6 Proof of Theorem 8

Lemma 15 *For an infinitary ground formula F, a set of constants $\mathbf{c}$ and two interpretations I and J such that $J <^{\mathbf{c}} I$, if $I \models gr_I(F)^{\underline{I}}$ and $J \not\models gr_I(F)^{\underline{I}}$, then there is some constant d occurring strictly positively in $gr_I(F)^{\underline{I}}$ such that $d(\boldsymbol{\xi})^I \neq d(\boldsymbol{\xi})^J$ for some tuple $\boldsymbol{\xi}$ of elements from $|I|$.*

Proof.

The proof of this claim is by induction:

- Case 1: F is an atomic formula. In this case $gr_I(F)^{\underline{I}} = F$ since $I \models gr_I(F)^{\underline{I}}$. And since $J \not\models gr_I(F)^{\underline{I}}$, there must be at least one constant in $gr_I(F)^{\underline{I}}$ that I and J disagree on and since $gr_I(F)^{\underline{I}}$ is an atomic formula, this is a strictly positively occurrence.

- Case 2: F is $\mathcal{H}^\wedge$. Since $I \models gr_I(F)^{\underline{I}}$, $gr_I(F)^{\underline{I}}$ is $\mathcal{H}'^\wedge$ (and not $\bot$) where $\mathcal{H}' = \{gr_I(G)^{\underline{I}} | G \in \mathcal{H}\}$. Since $J \not\models gr_I(F)^{\underline{I}}$, $J \not\models gr_I(G)^{\underline{I}}$ for at least one $G \in \mathcal{H}$ so the claim follows by induction on whichever subformula J does not satisfy since in any case, the subformula occurs strictly positively.

- Case 3: F is $\mathcal{H}^\vee$. Since $I \models gr_I(F)^{\underline{I}}$, $gr_I(F)^{\underline{I}}$ is $\mathcal{H}'^\vee$ (and not $\bot$) where $\mathcal{H}' = \{gr_I(G)^{\underline{I}} | G \in \mathcal{H}\}$. Since $J \not\models gr_I(F)^{\underline{I}}$, $J \not\models gr_I(G)^{\underline{I}}$ for every $G \in \mathcal{H}$. Now it could be that $I \not\models gr_I(G)^{\underline{I}}$ for some $G \in \mathcal{H}$ but not all of them. In such a case $gr_I(G)^{\underline{I}}$ would be $\bot$, which I also does not satisfy. Thus the claim follows by induction on whichever of $G \in \mathcal{H}$ whose reduct I satisfies.

- Case 4: F is $G \rightarrow H$. $gr_I(F)^{\underline{I}}$ is $gr_I(G)^{\underline{I}} \rightarrow gr_I(H)^{\underline{I}}$ (and not $\bot$). Since $J \not\models gr_I(F)^{\underline{I}}$, $J \models gr_I(G)^{\underline{I}}$ and $J \not\models gr_I(H)^{\underline{I}}$. Note that it must be the case then that $I \models gr_I(G)^{\underline{I}}$ since if not, it must be that $gr_I(G)^{\underline{I}}$ is $\bot$ and thus it is impossible for it to be that $J \models gr_I(G)^{\underline{I}}$. Consequently, it also follows that $I \models gr_I(H)^{\underline{I}}$ since $I \models gr_I(F)^{\underline{I}}$ so the claim follows by induction on H since the subformula occurs strictly positively. ∎

Theorem 8 *For any sentence F in Clark normal form that is tight on $\mathbf{c}$, an interpretation I that satisfies $\exists xy(x \neq y)$ is a model of SM$[F; \mathbf{c}]$ iff I is a model of the completion of F relative to $\mathbf{c}$.*

Proof.

In this proof, we use Theorem 1 and refer to the reduct characterization.

($\Leftarrow$) Take an interpretation I that is a model of the completion of F. We wish to show that for any interpretation J where $J <^{\mathbf{c}} I$, $J \not\models gr_I[F]^{\underline{I}}$. Let S be a subset of $\mathbf{c}$ such that I and J disagree on all constants in S–that is, those constants c for which there exists some tuple ξ such that $c(\xi)^I \neq c(\xi)^J$. Now let s_0 be a constant from S such that there is no edge in the dependency graph from s_0 to any constant in S. Such an s_0 is guaranteed to exist since F is tight on $\mathbf{c}$.

If s_0 is a predicate, then for some ξ, $s_0(\boldsymbol{\xi})^I = 1$ and $s_0(\boldsymbol{\xi})^J = 0$ by definition of

$J <^{\mathbf{c}} I$. If s_0 is a function, let v be $s_0(\boldsymbol{\xi})^I$. Note that since I is a model of the completion of F and since F is in clark normal form, there must be a rule in $gr_I[F]$ of the form $B \to s_0(\boldsymbol{\xi}^\diamond)$ if s_0 is a predicate ($B \to s_0(\boldsymbol{\xi}^\diamond) = v$ if s_0 is a function) where B may be $\top$. Further it must be that $I \models B$ since if not, I would not be a model of the completion of F. Thus, the corresponding rule in $gr_I[F]^{\underline{I}}$ is $B^{\underline{I}} \to s_0(\boldsymbol{\xi}^\diamond)$ ($B^{\underline{I}} \to s_0(\boldsymbol{\xi})^\diamond = v$ if s_0 is a function).

Now there are two cases to consider:

- Case 1: $J \models B^{\underline{I}}$. In this case, $J \not\models B^{\underline{I}} \to s_0(\boldsymbol{\xi}^\diamond)$ (or $J \not\models B^{\underline{I}} \to s_0(\boldsymbol{\xi}^\diamond) = v$ if s_0 is a function) and so $J \not\models gr_I[F]^{\underline{I}}$.

- Case 2: $J \not\models B^{\underline{I}}$. By Lemma 15, there is a constant d occurring strictly positively in B that I and J disagree on. However, this means there is an edge from s_0 to d and since I and J disagree on d, d belongs to S which contradicts the fact that s_0 was chosen so that it had no edge to any element in S. Thus this case cannot arise.

($\Rightarrow$) Assume $I \models \mathrm{SM}[F; \boldsymbol{c}]$. Now for every rule r in F of the form $\forall \boldsymbol{x}(H(\boldsymbol{x}) \leftarrow G(\boldsymbol{x}))$, for each of the ground rules in $gr_I[F]$ corresponding to r of the form $H(\xi) \leftarrow G(\xi)$ there are two cases:

- Case 1: $I \models G(\xi)$.

  In this case, since $I \models F$, it must also be that $I \models H(\xi)$. Thus, $I \models H(\xi) \leftrightarrow G(\xi)$.

- Case 2: $I \not\models G(\xi)$.

  The corresponding rule in the reduct $gr_I[F]^{\underline{I}}$ is either

$$H(\xi) \leftarrow \bot$$

or

$$\bot \leftarrow \bot$$

depending on if $I \models H(\xi)$. However, since F is in clark normal form, $H(\xi)$ appears in the head of no other rule. Thus, if $I \models H(\xi)$, $I \not\models \mathrm{SM}[F; \boldsymbol{c}]$ since we can take $J <^{\boldsymbol{c}} I$ ($I \models \exists xy(x \neq y)$ means there are at least two elements in the universe so this is possible) that differs from I only in that $J \not\models H(\xi)$ which will satisfy $F^{\underline{I}}$. Thus, it must be that $I \not\models H(\xi)$. It then follows that $I \models H(\xi) \leftrightarrow G(\xi)$. ∎

Chapter 6

ELIMINATING INTENSIONAL PREDICATES IN FAVOR OF INTENSIONAL FUNCTIONS

6.1 Embedding 1988 Definition of a Stable Model

Before considering the general case of eliminating intensional predicates in favor of intensional functions, we first explore a special case. We will see how to turn propositional logic programs under the semantics in Gelfond and Lifschitz (1988) into formulas under the functional stable model semantics which have no predicate constants.

Let Π be a finite set of rules of the form

$$A_0 \leftarrow A_1, \ldots, A_m, \text{not } A_{m+1}, \ldots, \text{not } A_n \tag{6.1}$$

($n \geq m \geq 0$), where each A_i is a propositional atom from the signature σ. The stable models of Π in the sense of Gelfond and Lifschitz (1988) can be characterized in terms of SM, in the same way as is handled in IF programs Lifschitz (2012). Lifschitz 2012 defines the *functional image* of Π as follows. First, reclassify all propositional atoms as intensional object constants, and add to the σ two non-intensional object constants 0 and 1 to obtain a new signature σ_{func}. Each rule (6.1) is rewritten as

$$A_0 = 1 \leftarrow A_1 = 1 \wedge \cdots \wedge A_m = 1 \wedge A_{m+1} \neq 1 \wedge \cdots \wedge A_n \neq 1$$

($A \neq 1$ is shorthand for $\neg(A = 1)$). For each atom A in the signature of Π we add the default rule

$$A = 0 \leftarrow \neg\neg(A = 0)$$

(by default, atoms get the value false). Finally, we add constraints

$$
\begin{aligned}
0 &\neq 1, \\
x = 0 &\vee x = 1.
\end{aligned}
\tag{6.2}
$$

The resulting program is called the *functional image* of Π. Clearly, the models of (6.2) can be viewed as sets of propositional atoms. Given a program Π whose signature is σ and interpretation I of the functional image of Π, the corresponding set X_I of propositional atoms is

$$\{p|p^I = 1 \text{ where } p \text{ is an object constant in } \sigma_{func}\}.$$

Inversely, given a program Π whose signature is σ and a set of propositional atoms X, the corresponding interpretation I_X is defined such that

$$p^{I_X} = \begin{cases} 1 & p \in X \\ 0 & \text{otherwise} \end{cases}$$

for each atom p in σ. The following theorem is similar to Proposition 5 from Lifschitz (2012), but applies to the functional stable model semantics presented here.

Theorem 9 *Let* Π *be a program of signature* σ*.*

- *If* X *is a stable model of* Π*, then* I_X *is a stable model of the functional image of* Π*.*
- *If* I *is a stable model of the functional image of* Π*, then* X_I *is a stable model of* Π*.*

Example 12 *Consider the program* Π*:*

$$
\begin{aligned}
p &\leftarrow q. \\
q &\leftarrow p.
\end{aligned}
$$

This has $X = \emptyset$ as its only stable model. We now consider the functional image of Π:

$$\begin{array}{rcl} p = 1 & \leftarrow & q = 1. \\ q = 1 & \leftarrow & p = 1. \\ p = 0 & \leftarrow & \neg\neg(p = 0). \\ q = 0 & \leftarrow & \neg\neg(q = 0). \\ & 0 \neq 1. & \\ x = 0 & \vee & x = 1. \end{array}$$

We can see that in the functional image, the idea of minimizing predicates is made explicit by the third and fourth lines establishing that p and q should both be 0 (signifying false) by default. Consider the interpretation I such that $|I| = \{0, 1\}$, $p^I = 0$, $q^I = 0$, $1^I = 1$, $0^I = 0$ (the interpretation corresponding to X). We can see that the reduct of the functional image of Π w.r.t. to I is equivalent to

$$\begin{array}{rcl} p = 1 & \leftarrow & q = 1. \\ q = 1 & \leftarrow & p = 1. \\ p = 0 & \leftarrow \top. & \\ q = 0 & \leftarrow \top. & \\ x = 0 & \vee & x = 1. \end{array}$$

and no other interpretation different from I on p, q satisfies this reduct. Thus, I is a stable model of the functional image of Π which corresponds to the only stable model of Π.

On the other hand, we can see that for the interpretation J such that $|J| = \{0, 1\}$, $p^J = 1$, $q^J = 1$, $1^J = 1$, $0^J = 0$, the reduct of the functional image of Π w.r.t. to J is *equivalent to*

$$p = 1 \;\leftarrow\; q = 1.$$

$$q = 1 \;\leftarrow\; p = 1.$$

$$x = 0 \;\vee\; x = 1.$$

however, the interpretation I which is different from J on p and q is a model of this reduct and so J is not a stable model of the functional image of Π *just as the corresponding set $Y = \{p, q\}$ is not a stable model of* Π.

6.2 Eliminating Intensional Predicates

The process in the previous section can be extended to eliminate intensional predicates in favor of intensional functions. Given a formula F and an intensional predicate constant p, formula F^p_f is obtained from F as follows:

- in the signature of F, replace p with a new intensional function constant f of arity n, where n is the arity of p, and add two non-intensional object constants 0 and 1;
- replace each subformula $p(\boldsymbol{t})$ in F with $f(\boldsymbol{t}) = 1$.

By FC_f ("Functional Constraint on f") we denote the conjunction of the following formulas, which enforces f to behave like predicates:

$$0 \neq 1, \tag{6.3}$$

$$\neg\neg\forall \boldsymbol{x}(f(\boldsymbol{x}) = 0 \vee f(\boldsymbol{x}) = 1). \tag{6.4}$$

where $\boldsymbol{x}$ is a list of distinct object variables. By DF_f ("Default False on f") we denote the following formula:

$$\forall \boldsymbol{x}(\neg\neg(f(\boldsymbol{x})=0) \rightarrow f(\boldsymbol{x})=0). \tag{6.5}$$

Example 13 *Let F be the conjunction of the universal closures of the following formulas, which describes the effect of a monkey moving:*

$$loc(monkey, 0) = l1,$$
$$loc(monkey, 1) = l2,$$
$$move(monkey, L, T) \rightarrow loc(monkey, T+1) = L$$

We eliminate the intensional predicate move in favor of an intensional function $move_f$ to obtain $F^{move}_{move_f} \wedge FC_{move_f} \wedge DF_{move_f}$, which is the conjunction of the universal closures of the following formulas:

$$loc(monkey, 0) = l1,$$
$$loc(monkey, 1) = l2,$$
$$move_f(monkey, L, T) = 1 \rightarrow loc(monkey, T+1) = L,$$
$$0 \neq 1,$$
$$\neg\neg\forall xyz(move_f(x,y,z) = 0 \vee move_f(x,y,z) = 1)$$
$$\forall xyz(\neg\neg(move_f(x,y,z) = 0) \rightarrow move_f(x,y,z) = 0).$$

Theorem 10 *Formulas $\forall \boldsymbol{x}(f(\boldsymbol{x}) = 1 \leftrightarrow p(\boldsymbol{x}))$, FC_f entail $SM[F; p\boldsymbol{c}] \leftrightarrow SM[F^p_f \wedge DF_f; f\boldsymbol{c}]$.*

The following corollary shows that there is a 1–1 correspondence between the stable models of F and the stable models of its "functional image" $F^p_f \wedge DF_f \wedge FC_f$. For any interpretation I of the signature of F, by I^p_f we denote the interpretation with universe $|I|$ and with the signature of F^p_f obtained from I by replacing the set p^I with the function f^I such that

$$f^I(\xi_1,\ldots,\xi_n)=1 \text{ if } p^I(\xi_1,\ldots,\xi_n)=1$$
$$f^I(\xi_1,\ldots,\xi_n)=0 \text{ otherwise .}$$

where each $\xi_i \in |I|$. A further constraint on I^p_f is that $I^p_f \models 1 \neq 0$. Consequently, I^p_f satisfies FC_f.

Corollary 1 *(a) An interpretation I of the signature of F is a model of $SM[F;p\boldsymbol{c}]$ iff I^p_f is a model of $SM[F^p_f \wedge DF_f; f\boldsymbol{c}]$. (b) An interpretation J of the signature of F^p_f is a model of $SM[F^p_f \wedge DF_f \wedge FC_f; f\boldsymbol{c}]$ iff $J = I^p_f$ for some model I of $SM[F;p\boldsymbol{c}]$.*

6.3 Relating Strong Negation to Boolean Functions

6.3.1 Representing Strong Negation in Multi-Valued Propositional Formulas

The notion of strong negation (or classical negation) has been useful in logic program. In particular, in combination with default negation (or negation as failure) in solving the frame problem–that the world does not arbitrarily change.

Example 14 *Consider the program that describes a simple transition system consisting of two states depending on whether fluent p is true or false, and an action a that makes p true (subscripts 0 and 1 represent time stamps).*

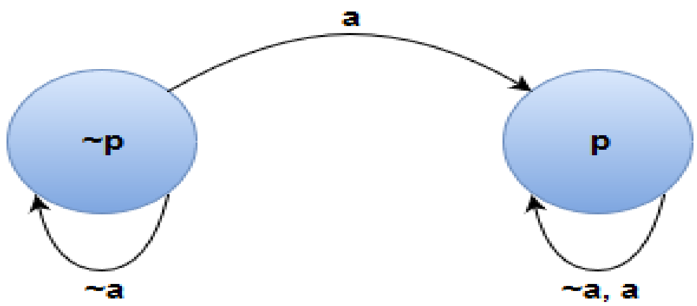

Figure 6.1: Transition System

$$
\begin{array}{rcl}
p_1 & \leftarrow & a \\
p_1 & \leftarrow & p_0, not \;\; {\sim}p_1 \\
{\sim}p_1 & \leftarrow & {\sim}p_0, not \; p_1 \\
p_0 & \leftarrow & not \;\; {\sim}p_0 \\
{\sim}p_0 & \leftarrow & not \; p_0 \\
a & \leftarrow & not \;\; {\sim}a \\
{\sim}a & \leftarrow & not \;\; a.
\end{array}
\tag{6.6}
$$

The program has four answer sets, each of which corresponds to one of the four edges of the transition system. For instance, $\{{\sim}p_0, a, p_1\}$ *is an answer set.*

However, strong negation in the stable model semantics is not a primitive connective [1] . We provide an alternate characterization of the notion of strong negation by translating a propositional logic program into a multi-valued propositional formula in which all constants are Boolean.

Given a traditional propositional logic program Π of a signature σ Gelfond and Lifschitz (1991), we identify σ with the multi-valued propositional signature whose constants are the same symbols from σ and every constant is Boolean. By Π^{mv} we mean the multi-valued propositional formula that is obtained from Π by replacing

[1] Strong negation can only appear in front of an atom so that ${\sim}(p \vee q)$ is not a valid formula.

negative literals of the form $\sim p$ with $p = 0$ and positive literals of the form p with $p = 1$.

We say that a set X of literals from σ is *complete* if, for each atom $a \in \sigma$, either a or $\sim a$ is in X. We identify a complete set of literals from σ with the corresponding multi-valued propositional interpretation.

Theorem 11 *A complete set of literals is an answer set of Π in the sense of Gelfond and Lifschitz (1991) iff it is a stable model of Π^{mv} in the sense of the functional stable model semantics.*

The theorem tells us that checking the minimality of positive and negative literals under the traditional stable model semantics is essentially the same as checking the uniqueness of corresponding function values under the functional stable model semantics.

Example 14 continued *According to Theorem 11, the stable models of this program are the same as the stable models of the following multi-valued propositional formula (written in a logic program style; '$\neg$' represents default negation):*

$$\begin{array}{rcl}
p_0\!=\!1 & \leftarrow & \neg(p_0\!=\!0) \\
p_0\!=\!0 & \leftarrow & \neg(p_0\!=\!1) \\
\\
a\!=\!1 & \leftarrow & \neg(a\!=\!0) \\
a\!=\!0 & \leftarrow & \neg(a\!=\!1)
\end{array}
\qquad
\begin{array}{rcl}
p_1\!=\!1 & \leftarrow & a\!=\!1 \\
\\
p_1\!=\!1 & \leftarrow & p_0\!=\!1 \wedge \neg(p_1 = 0) \\
p_1\!=\!0 & \leftarrow & p_0\!=\!0 \wedge \neg(p_1 = 1)
\end{array}$$

6.3.2 Representing Strong Negation Using Boolean Functions in the First-Order Case

Theorem 11 can be extended to the first-order case as follows. However, we first define a syntactic restriction for the extension.

Let F be a formula possibly containing strong negation. Formula $F^{(p,\sim p)}_{b}$ is obtained from F as follows:

- in the signature of F, replace p and $\sim p$ with a new intensional function constant b of arity n, where n is the arity of p (or $\sim p$), and add two non-intensional object constants 1 and 0;
- replace every occurrence of $\sim p(\boldsymbol{t})$, where $\boldsymbol{t}$ is a list of terms, with $b(\boldsymbol{t}) = 0$, and then replace every occurrence of $p(\boldsymbol{t})$ with $b(\boldsymbol{t}) = 1$.

By BC_b ("Boolean Constraint on b") we denote the conjunction of the following formulas, which enforces b to be a Boolean function:

$$1 \neq 0, \tag{6.7}$$

$$\neg\neg\forall \boldsymbol{x}(b(\boldsymbol{x}) = 1 \lor b(\boldsymbol{x}) = 0). \tag{6.8}$$

where $\boldsymbol{x}$ is a list of distinct object variables.

Theorem 12 *Let $\boldsymbol{c}$ be a set of predicate and function constants, and let F be a $\boldsymbol{c}$-plain formula. Formulas*

$$\forall \boldsymbol{x}((p(\boldsymbol{x}) \leftrightarrow b(\boldsymbol{x}){=}1) \land (\sim p(\boldsymbol{x}) \leftrightarrow b(\boldsymbol{x}){=}0)), \tag{6.9}$$

and BC_b entail

$$SM[F;\ p, \sim p, \boldsymbol{c}] \leftrightarrow SM[F^{(p,\sim p)}_{b};\ b, \boldsymbol{c}]\ .$$

Example 14 continued *Consider the simple transition system from before. We obtain* $F_b^{(p,\sim p)}$

$$\begin{aligned}
b_1 = 1 &\leftarrow a \\
b_1 = 1 &\leftarrow b_0 = 1, not\ b_1 = 0 \\
b_1 = 0 &\leftarrow b_0 = 0, not\ b_1 = 1 \\
b_0 = 1 &\leftarrow not\ b_0 = 0 \\
b_0 = 0 &\leftarrow not\ b_0 = 1 \\
a &\leftarrow not\ \sim a \\
\sim a &\leftarrow not\ a.
\end{aligned}$$

We can see that the interpretation I such that

$$(b_0)^I = 0, \quad (\sim p_0)^I = \boldsymbol{t}, \quad (p_0)^I = \boldsymbol{f},$$
$$a^I = 0,$$
$$(b_1)^I = 1, \quad (p_1)^I = \boldsymbol{t}, \quad (\sim p_1)^I = \boldsymbol{f}$$

satisfies 6.9 and BC_b. Then we can see that $I \models SM[F;\ p, \sim p, \boldsymbol{c}] \leftrightarrow SM[F_b^{(p,\sim p)};\ b, \boldsymbol{c}]$ since $I \models SM[F;\ p, \sim p, \boldsymbol{c}]$ and $I \models SM[F_b^{(p,\sim p)};\ b, \boldsymbol{c}]$.

If we drop the requirement that F be $\boldsymbol{c}$-plain, the statement does not hold as the following example demonstrates.

Example 15 *Take $\boldsymbol{c}$ to be (f, g) and let F be $p(f) \wedge \sim p(g)$. $F_b^{(p,\sim p)}$ is $b(f) = 1 \wedge b(g) = 0$. Consider the interpretation I whose universe is $\{1, 2\}$ such that I contains $p(1), \sim p(2)$ and with the mappings $b^I(1) = 1, b^I(2) = 0, f^I = 1, g^I = 2$. I certainly satisfies BC_b and (6.9). I also satisfies $SM[F;\ p, \sim p, f, g]$ but does not satisfy $SM[F_b^{(p,\sim p)};\ b, f, g]$: we can take I such that $\widehat{b}^I(1) = 0, \widehat{b}^I(2) = 1, \widehat{f}^I = 2, \widehat{g}^I = 1$ to satisfy both $(\widehat{b}, \widehat{f}, \widehat{g}) < (b, f, g)$ and $(F_b^{(p,\sim p)})^*(\widehat{b}, \widehat{f}, \widehat{g})$, which is*

$$b(f) = 1 \wedge \widehat{b}(\widehat{f}) = 1 \wedge b(g) = 0 \wedge \widehat{b}(\widehat{g}) = 0.$$

Note that any interpretation that satisfies both (6.9) and BC_b is complete on p. Theorem 12 tells us that for any interpretation I that is complete on p, minimizing the extents of both p and $\sim p$ has the same effect as ensuring that the corresponding Boolean function b has a unique value.

The following corollary shows that there is a 1–1 correspondence between the stable models of F and the stable models of $F^{(p,\sim p)}_b$. We say an interpretation I is *coherent* if for every predicate p in the signature of I, we have $I \models \forall \boldsymbol{x}(\neg p(\boldsymbol{x}) \vee \neg \sim p(\boldsymbol{x}))$. For any coherent interpretation I of the signature of F that is complete on p, by $I^{(p,\sim p)}_b$ we denote the interpretation of the signature of $F^{(p,\sim p)}_b$ obtained from I by replacing the relation p^I with function b^I such that

$$
\begin{aligned}
b^I(\xi_1,\ldots,\xi_n) &= 1^I \;\text{ if }\; p^I(\xi_1,\ldots,\xi_n) = \boldsymbol{t};\\
b^I(\xi_1,\ldots,\xi_n) &= 0^I \;\text{ if }\; (\sim p)^I(\xi_1,\ldots,\xi_n) = \boldsymbol{t}\ .
\end{aligned}
$$

Since I is complete on p and coherent, b^I is well-defined. We also require that $I^{(p,\sim p)}_b$ satisfy (6.7). Consequently, $I^{(p,\sim p)}_b$ satisfies BC_b.

Corollary 2 *Let $\boldsymbol{c}$ be a set of predicate and function constants, and let F be a $\boldsymbol{c}$-plain sentence.* (I) *A coherent interpretation I of the signature of F that is complete on p is a model of $SM[F;\ p, \sim p, \boldsymbol{c}]$ iff $I^{(p,\sim p)}_b$ is a model of $SM[F^{(p,\sim p)}_b;\ b, \boldsymbol{c}]$.* (II) *An interpretation J of the signature of $F^{(p,\sim p)}_b$ is a model of $SM[F^{(p,\sim p)}_b \wedge BC_b;\ b, \boldsymbol{c}]$ iff $J = I^{(p,\sim p)}_b$ for some model I of $SM[F; p, \sim p, \boldsymbol{c}]$.*

The other direction, eliminating Boolean intensional functions in favor of symmetric predicates, is similar as we show in the following.

Let F be a $(b, \boldsymbol{c})$-plain formula such that every atomic formula containing b has the form $b(\boldsymbol{t}) = 1$ or $b(\boldsymbol{t}) = 0$, where $\boldsymbol{t}$ is any list of terms. Formula $F^{b}_{(p,\sim p)}$ is obtained from F as follows:

- in the signature of F, replace b with predicate constants p and $\sim p$, whose arities are the same as that of b;

- replace every occurrence of $b(\boldsymbol{t}) = 1$, where $\boldsymbol{t}$ is any list of terms, with $p(\boldsymbol{t})$, and $b(\boldsymbol{t}) = 0$ with $\sim p(\boldsymbol{t})$.

Theorem 13 *Let $\boldsymbol{c}$ be a set of predicate and function constants, let b be a function constant, and let F be a $(b, \boldsymbol{c})$-plain formula such that every atomic formula containing b has the form $b(\boldsymbol{t}) = 1$ or $b(\boldsymbol{t}) = 0$. Formulas (6.9) and BC_b entail*

$$SM[F;\ b, \boldsymbol{c}] \leftrightarrow SM[F^{\,b}_{(p,\sim p)};\ p, \sim p, \boldsymbol{c}]\ .$$

The following corollary shows that there is a 1–1 correspondence between the stable models of F and the coherent stable models of $F^{\,b}_{(p,\sim p)}$. For any interpretation I of the signature of F that satisfies BC_b, by $I^{\,b}_{(p,\sim p)}$ we denote the interpretation of the signature of $F^{\,b}_{(p,\sim p)}$ obtained from I by replacing the function b^I with predicate p^I such that

$$\begin{array}{l} p^I(\xi_1, \dots, \xi_n) = \boldsymbol{t} \ \text{ iff } \ b^I(\xi_1, \dots, \xi_n) = 1^I\ ; \\ (\sim p)^I(\xi_1, \dots, \xi_n) = \boldsymbol{t} \ \text{ iff } \ b^I(\xi_1, \dots, \xi_n) = 0^I\ . \end{array}$$

Corollary 3 *Let $\boldsymbol{c}$ be a set of predicate and function constants, let b be a function constant, and let F be a $(b, \boldsymbol{c})$-plain sentence such that every atomic formula containing b has the form $b(\boldsymbol{t}) = 1$ or $b(\boldsymbol{t}) = 0$.* (I) *A coherent interpretation I of the signature of F is a model of $SM[F \wedge BC_b;\ b, \boldsymbol{c}]$ iff $I^{\,b}_{(p,\sim p)}$ is a model of $SM[F^{\,b}_{(p,\sim p)};\ p, \sim p, \boldsymbol{c}]$.* (II) *An interpretation J of the signature of $F^{\,b}_{(p,\sim p)}$ is a model of $SM[F^{\,b}_{(p,\sim p)}; p, \sim p, \boldsymbol{c}]$ iff $J = I^{\,b}_{(p,\sim p)}$ for some model I of $SM[F \wedge BC_b; b, \boldsymbol{c}]$.*

6.4 Proofs

6.4.1 Proof of Theorem 9

Theorem 9 *Let Π be a program of signature σ.*

- *If X is a stable model of Π, then I_X is a stable model of the functional image of Π.*
- *If I is a stable model of the functional image of Π, then X_I is a stable model of Π.*

Proof. Let $\boldsymbol{p}$ denote all of the atoms in σ and let $\boldsymbol{f}$ denote all of the corresponding object constants in the signature of the functional image of Π. We first note that I_X is the same as $I^{\boldsymbol{p}}_{\boldsymbol{f}}$. We also note that the added rules

$$\{A\!=\!0\}^{ch}$$

and

$$0 \neq 1,$$
$$x = 0 \vee x = 1.$$

are precisely $DF_f \wedge FC_f$ when considering their first-order representation. Finally, we note then that the first-order representation of functional image of Π is exactly $F^{\boldsymbol{p}}_{\boldsymbol{f}} \wedge DF_f \wedge FC_f$ where F is the first-order representation of Π. Then, the claim follows from multiple applications of Corollary 1 for each p in $\boldsymbol{p}$ and the corresponding f in $\boldsymbol{f}$. ∎

6.4.2 Proof of Theorem 10

Theorem 10 *Formulas*

$$\forall \boldsymbol{x}(f(\boldsymbol{x}) = 1 \leftrightarrow p(\boldsymbol{x})), \tag{6.10}$$

and FC_f *entail* $SM[F; p\boldsymbol{c}] \leftrightarrow SM[F^p_f \wedge DF_f; f\boldsymbol{c}]$.

Proof.

For any interpretation $\mathcal{I} = \langle I, X\rangle$ of signature $\sigma \supseteq \{f, p, \boldsymbol{c}\}$ satisfying (6.10), it is clear that $\mathcal{I} \models F$ iff $\mathcal{I} \models F^p_f \wedge DF_f$ since DF_f is a tautology and F^p_f is equivalent to F under (6.10). Thus it only remains to be shown that

$$\mathcal{I} \models \neg\exists \widehat{p}\widehat{\boldsymbol{c}}((\widehat{p}\widehat{\boldsymbol{c}} < p\boldsymbol{c}) \wedge F^*(\widehat{p}, \widehat{\boldsymbol{c}}))$$

iff

$$\mathcal{I} \models \neg\exists \widehat{f}\widehat{\boldsymbol{c}}((\widehat{f}\widehat{\boldsymbol{c}} < f\boldsymbol{c}) \wedge (F^p_f)^*(\widehat{f}, \widehat{\boldsymbol{c}}) \wedge DF^*_f(\widehat{p}\widehat{\boldsymbol{c}}))$$

or equivalently,

$$\mathcal{I} \models \exists \widehat{p}\widehat{\boldsymbol{c}}((\widehat{p}\widehat{\boldsymbol{c}} < p\boldsymbol{c}) \wedge F^*(\widehat{p}, \widehat{\boldsymbol{c}}))$$

iff

$$\mathcal{I} \models \exists \widehat{f}\widehat{\boldsymbol{c}}((\widehat{f}\widehat{\boldsymbol{c}} < f\boldsymbol{c}) \wedge (F^p_f)^*(\widehat{f}, \widehat{\boldsymbol{c}}) \wedge DF^*_f(\widehat{f}\widehat{\boldsymbol{c}})).$$

($\Rightarrow$) Assume $\mathcal{I} \models \exists \widehat{p}\widehat{\boldsymbol{c}}((\widehat{p}\widehat{\boldsymbol{c}} < p\boldsymbol{c}) \wedge F^*(\widehat{p}, \widehat{\boldsymbol{c}}))$. We wish to show that $\mathcal{I} \models \exists \widehat{f}\widehat{\boldsymbol{c}}((\widehat{f}\widehat{\boldsymbol{c}} < f\boldsymbol{c}) \wedge (F^p_f)^*(\widehat{f}, \widehat{\boldsymbol{c}}) \wedge DF^*_f(\widehat{f}\widehat{\boldsymbol{c}}))$.

That is, take any predicate q of the same arity as p and any list of predicates and functions $\boldsymbol{d}$ of the same length as $\boldsymbol{c}$. Now let $\mathcal{I}' = \langle I \cup J^{f\boldsymbol{c}}_{g\boldsymbol{d}}, X \cup Y^{\boldsymbol{c}}_{\boldsymbol{d}}\rangle$ be from an extended signature $\sigma' = \sigma \cup \{g, q, \boldsymbol{d}\}$ where J is an interpretation of functions from the signature σ and I and J agree on all functions not in $\{f, \boldsymbol{c}\}$. $J^{f\boldsymbol{c}}_{g\boldsymbol{d}}$ denotes the interpretation from $\sigma^{f\boldsymbol{c}}_{g\boldsymbol{d}}$ (the signature obtained from σ by replacing f with g and all elements of $\boldsymbol{c}$ with all elements of $\boldsymbol{d}$) obtained from the interpretation J by replacing

f with g and the functions in $\boldsymbol{c}$ with the corresponding functions in $\boldsymbol{d}$. Similarly, $Y^{\boldsymbol{c}}_{\boldsymbol{d}}$ is the interpretation from $\sigma^{\boldsymbol{c}}_{\boldsymbol{d}}$ obtained from the interpretation Y by replacing predicates from $\boldsymbol{c}$ by the corresponding predicates from $\boldsymbol{d}$. We assume

$$\mathcal{I}' \models (q\boldsymbol{d} < p\boldsymbol{c} \wedge F^*(q\boldsymbol{d}))$$

and wish to show that there is a function g of the same arity as f such that

$$\mathcal{I}' \models (g\boldsymbol{d} < f\boldsymbol{c} \wedge (F^p_f)^*(g\boldsymbol{d}) \wedge DF^*_f(g\boldsymbol{d})).$$

We define the new function g in terms of q as follows:

$$g^{\mathcal{I}'}(\vec{\xi}) = \begin{cases} 1 & \text{if } \mathcal{I}' \models q(\vec{\xi}) \\ 0 & \text{otherwise} \end{cases}$$

We now show $\mathcal{I}' \models g\boldsymbol{d} < f\boldsymbol{c}$:

Case 1: $\mathcal{I}' \models (q = p)$.

Since $\mathcal{I}' \models q\boldsymbol{d} < p\boldsymbol{c}$, by definition $\mathcal{I}' \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$ and $\mathcal{I}' \models \neg(q\boldsymbol{d} = p\boldsymbol{c})$ and since in this case, $\mathcal{I}' \models (q = p)$, $\mathcal{I}' \models \neg(\boldsymbol{d} = \boldsymbol{c})$. From this, we conclude $\mathcal{I}' \models \neg(g\boldsymbol{d} = f\boldsymbol{c})$. Further, since $\mathcal{I}' \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$, we conclude $\mathcal{I}' \models g\boldsymbol{d} < f\boldsymbol{c}$.

Case 2: $\mathcal{I}' \models \neg(q = p)$.

Since $\mathcal{I}' \models q\boldsymbol{d} < p\boldsymbol{c}$, by definition, $\mathcal{I}' \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$ and $\mathcal{I}' \models (q \leq p)$. Thus, since in this case $\mathcal{I}' \models \neg(q = p)$, then it follows that $\mathcal{I}' \models \exists\boldsymbol{x}(p(\boldsymbol{x}) \wedge \neg q(\boldsymbol{x}))$. From the definition of g and from (6.10), this is equivalent to $\mathcal{I}' \models \exists\boldsymbol{x}(f(\boldsymbol{x}) = 1 \wedge g(\boldsymbol{x}) = 0)$. Thus, we conclude $\mathcal{I}' \models \neg(f = g)$ and since $\mathcal{I}' \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$, we further conclude that $\mathcal{I}' \models g\boldsymbol{d} < f\boldsymbol{c}$.

We now show $\mathcal{I}' \models DF^*_f(g\boldsymbol{d})$:

Since $\mathcal{I}' \models q\boldsymbol{d} < p\boldsymbol{c}$, by definition, $\mathcal{I}' \models (q \leq p)$, or equivalently $\mathcal{I}' \models \forall\boldsymbol{x}(q(\boldsymbol{x}) \rightarrow p(\boldsymbol{x}))$ and by contraposition, $\mathcal{I}' \models \forall\boldsymbol{x}(\neg p(\boldsymbol{x}) \rightarrow \neg q(\boldsymbol{x}))$. Finally, by (6.10),$FC_f$, and the definition of g, $\mathcal{I}' \models \forall\boldsymbol{x}(f(\boldsymbol{x}) = 0 \rightarrow g(\boldsymbol{x}) = 0)$ or simply $\mathcal{I}' \models DF^*_f(g\boldsymbol{d})$.

We now show $\mathcal{I}' \models (F^p_f)^*(g\boldsymbol{d})$ by proving that $\mathcal{I}' \models (F^p_f)^*(g\boldsymbol{d})$ iff $\mathcal{I}' \models F^*(q\boldsymbol{d})$:

Case 1: F is an atomic formula not containing p.

F^p_f is exactly F thus $F^*(q\boldsymbol{d})$ is exactly $(F^p_f)^*(g\boldsymbol{d})$ so certainly the claim holds.

Case 2: F is $p(\boldsymbol{t})$ where $\boldsymbol{t}$ contains an intensional function constant from $\boldsymbol{c}$.

$F^*(q\boldsymbol{d})$ is $p(\boldsymbol{t}) \wedge q(\boldsymbol{t}')$

where $\boldsymbol{t}'$ is the result of replacing all intensional functions from $\boldsymbol{c}$ occurring in $\boldsymbol{t}$ with the corresponding function from $\boldsymbol{d}$.

F^p_f is $f(\boldsymbol{t}) = 1$.

$(F^p_f)^*(g\boldsymbol{d})$ is $f(\boldsymbol{t}) = 1 \wedge g(\boldsymbol{t}') = 1$.

Since $\mathcal{I}' \models p(\boldsymbol{t}) \wedge q(\boldsymbol{t}')$, it follows from (6.10) and the definition of g that $\mathcal{I}' \models f(\boldsymbol{t}) = 1 \wedge g(\boldsymbol{t}') = 1$.

Case 3: F is $p(\boldsymbol{t})$ where $\boldsymbol{t}$ does not contain any intensional function constant from $\boldsymbol{c}$.

$F^*(q\boldsymbol{d})$ is $q(\boldsymbol{t})$.

F^p_f is $f(\boldsymbol{t}) = 1$.

$(F^p_f)^*(g\boldsymbol{d})$ is $f(\boldsymbol{t}) = 1 \wedge g(\boldsymbol{t}) = 1$.

Now, since $\mathcal{I}' \models (q \leq p)$, if $\mathcal{I}' \models q(\boldsymbol{t})$, then $\mathcal{I}' \models p(\boldsymbol{t})$. From (6.10), it follows that $\mathcal{I}' \models f(\boldsymbol{t}) = 1$ and from the definition of g, it follows that $\mathcal{I}' \models g(\boldsymbol{t}) = 1$.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.

By I.H. on G and H.

Case 5: F is $G \rightarrow H$.

By I.H. on G and H.

Case 6: F is $Q\boldsymbol{x}G(\boldsymbol{x})$ where $Q \in \{\forall, \exists\}$.

By I.H. on G.

($\Leftarrow$) Assume $\mathcal{I} \models \exists \widehat{f}\widehat{\boldsymbol{c}}((\widehat{f}\widehat{\boldsymbol{c}} < f\boldsymbol{c}) \wedge (F^p_f)^*(\widehat{f},\widehat{\boldsymbol{c}}) \wedge DF^*_f(\widehat{f}\widehat{\boldsymbol{c}}))$. We wish to show that $\mathcal{I} \models \exists \widehat{p}\widehat{\boldsymbol{c}}((\widehat{p}\widehat{\boldsymbol{c}} < p\boldsymbol{c}) \wedge F^*(\widehat{p},\widehat{\boldsymbol{c}}))$.

That is, take any function g of the same arity as f and any list of predicates and functions $\boldsymbol{d}$ of the same length $\boldsymbol{c}$ and let $\mathcal{I}' = \langle I \cup J^{f\boldsymbol{c}}_{g\boldsymbol{d}}, X \cup Y^{\boldsymbol{c}}_{\boldsymbol{d}}\rangle$ be defined as before. We assume

$$\mathcal{I}' \models (g\boldsymbol{d} < f\boldsymbol{c} \wedge (F^p_f)^*(g\boldsymbol{d}) \wedge DF^*_f(g\boldsymbol{d}))$$

We wish to show that there is a predicate q of the same arity as p such that

$$\mathcal{I}' \models (q\boldsymbol{d} < p\boldsymbol{c} \wedge F^*(q\boldsymbol{d})).$$

We define the new predicate q in terms of g as follows:

$$q^{\mathcal{I}'}(\vec{\xi}) = \begin{cases} 1 & \text{if } \mathcal{I}' \models g(\vec{\xi}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

We now show $\mathcal{I}' \models q\boldsymbol{d} < p\boldsymbol{c}$:

Case 1: $\mathcal{I}' \models (g = f)$.

By definition of q and by (6.10), in this case, $\mathcal{I}' \models q = p$ and in particular, $\mathcal{I}' \models q \leq p$. Since $\mathcal{I}' \models g\boldsymbol{d} < f\boldsymbol{c}$, by definition $\mathcal{I}' \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$ and $\mathcal{I}' \models \neg(g\boldsymbol{d} = f\boldsymbol{c})$ and since in this case, $\mathcal{I}' \models (g = f)$, then $\mathcal{I}' \models \neg(\boldsymbol{d} = \boldsymbol{c})$. From this, we conclude $\mathcal{I}' \models \neg(q\boldsymbol{d} = p\boldsymbol{c})$. Finally, we conclude $\mathcal{I}' \models q\boldsymbol{d} < p\boldsymbol{c}$.

Case 2: $\mathcal{I}' \models \neg(g = f)$.

Since $\mathcal{I}' \models DF^*_f(g\boldsymbol{d})$, then $\mathcal{I}' \models \forall \boldsymbol{x}(f(\boldsymbol{x}) = 0 \rightarrow g(\boldsymbol{x}) = 0)$. From this, we conclude by definition of q, FC_f (note that $0 \neq 1$ is essential here) and (6.10) that $\mathcal{I}' \models \forall \boldsymbol{x}(\neg p(\boldsymbol{x}) \rightarrow \neg q(\boldsymbol{x}))$. Equivalently, this is $\mathcal{I}' \models \forall \boldsymbol{x}(q(\boldsymbol{x}) \rightarrow p(\boldsymbol{x}))$ or simply $\mathcal{I}' \models q \leq p$.

Now, since $\mathcal{I}' \models FC_f$, then $\mathcal{I}' \models \forall \boldsymbol{x}(f(\boldsymbol{x}) = 0 \vee f(\boldsymbol{x}) = 1)$. Thus, for the assumption in this case that $\mathcal{I}' \models \neg(g = f)$ to hold, it must be that $\mathcal{I}' \models \exists \boldsymbol{x}(f(\boldsymbol{x}) = 1 \wedge \neg(g(\boldsymbol{x}) = 1))$. By defintion of q and (6.10), it follows that $\mathcal{I}' \models \exists \boldsymbol{x}(p(\boldsymbol{x}) \wedge \neg q(\boldsymbol{x}))$. Thus, since $\mathcal{I}' \models \neg(q = p)$, then $\mathcal{I}' \models \neg(q\boldsymbol{d} = p\boldsymbol{c})$. Also, since $\mathcal{I}' \models g\boldsymbol{d} < f\boldsymbol{c}$, by definition $\mathcal{I}' \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$, and thus we conclude that $\mathcal{I}' \models q\boldsymbol{d} < p\boldsymbol{c}$.

We now show $\mathcal{I}' \models F^*(q\boldsymbol{d})$ by proving that $\mathcal{I}' \models (F^p_f)^*(g\boldsymbol{d})$ iff $\mathcal{I}' \models F^*(q\boldsymbol{d})$:

Case 1: F is an atomic formula not containing p.

F^p_f is exactly F thus $F^*(q\boldsymbol{d})$ is exactly $(F^p_f)^*(g\boldsymbol{d})$ so certainly the claim holds.

Case 2: F is $p(\boldsymbol{t})$ where $\boldsymbol{t}$ contains an intensional function constant from $\boldsymbol{c}$.

$F^*(q\boldsymbol{d})$ is $p(\boldsymbol{t}) \wedge q(\boldsymbol{t}')$

where $\boldsymbol{t}'$ is the result of replacing all intensional functions from $\boldsymbol{c}$ occurring in $\boldsymbol{t}$ with the corresponding function from $\boldsymbol{d}$

F^p_f is $f(\boldsymbol{t}) = 1$.

$(F^p_f)^*(g\boldsymbol{d})$ is $f(\boldsymbol{t}) = 1 \wedge g(\boldsymbol{t}') = 1$.

Since $\mathcal{I}' \models f(\boldsymbol{t}) = 1 \wedge g(\boldsymbol{t}') = 1$, by definition of q and (6.10), $\mathcal{I}' \models p(\boldsymbol{t}) \wedge q(\boldsymbol{t}')$ and thus $\mathcal{I}' \models F^*(q\boldsymbol{d})$.

Case 3: F is $p(\boldsymbol{t})$ where $\boldsymbol{t}$ does not contain any intensional function constant from $\boldsymbol{c}$.

$F^*(q\boldsymbol{d})$ is $q(\boldsymbol{t})$.

F^p_f is $f(\boldsymbol{t}) = 1$.

$(F^p_f)^*(g\boldsymbol{d})$ is $f(\boldsymbol{t}) = 1 \wedge g(\boldsymbol{t}) = 1$.

By definition of q and since $\mathcal{I}' \models f(\boldsymbol{t}) = 1 \wedge g(\boldsymbol{t}) = 1$, $\mathcal{I}' \models q(\boldsymbol{t})$ and thus $\mathcal{I}' \models F^*(q\boldsymbol{d})$ in this case.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.

By I.H. on G and H.

Case 5: F is $G \rightarrow H$.

By I.H. on G and H.

Case 6: F is $Q\boldsymbol{x}G(\boldsymbol{x})$ where $Q \in \{\forall, \exists\}$.

By I.H. on G. ∎

6.4.3 Proof of Corollary 1

Corollary 1 *(a) An interpretation I of the signature of F is a model of SM$[F; p\boldsymbol{c}]$ iff I^p_f is a model of SM$[F^p_f \wedge DF_f; f\boldsymbol{c}]$. (b) An interpretation J of the signature of F^p_f is a model of SM$[F^p_f \wedge DF_f \wedge FC_f; f\boldsymbol{c}]$ iff $J = I^p_f$ for some model I of SM$[F; p\boldsymbol{c}]$.*

Proof.

(a$\Rightarrow$) Assume I of the signature of F is a model of SM$[F; p\boldsymbol{c}]$. By definition of I^p_f, $I \cup I^p_f \models \forall \boldsymbol{x}(f(\boldsymbol{x}) = 1 \leftrightarrow p(\boldsymbol{x}))$. Now, since $I \models$ SM$[F; p\boldsymbol{c}]$ by our assumption, it must be that $I \cup I^p_f \models$ SM$[F; p\boldsymbol{c}]$ and further by Theorem 10, since $I \cup I^p_f \models$ SM$[F; p\boldsymbol{c}] \leftrightarrow$ SM$[F^p_f \wedge DF_f; f\boldsymbol{c}]$, it must be that $I \cup I^p_f \models$ SM$[F^p_f \wedge DF_f; f\boldsymbol{c}]$. Finally, since the signature of I does not contain f, we conclude $I^p_f \models$ SM$[F^p_f \wedge DF_f; f\boldsymbol{c}]$.

(a$\Leftarrow$) Assume I^p_f is a model of SM$[F^p_f \wedge DF_f; f\boldsymbol{c}]$. By definition of I^p_f, $I \cup I^p_f \models \forall \boldsymbol{x}(f(\boldsymbol{x}) = 1 \leftrightarrow p(\boldsymbol{x}))$. Now, since $I^p_f \models$ SM$[F^p_f \wedge DF_f; f\boldsymbol{c}]$ by our assumption, it must be that $I \cup I^p_f \models$ SM$[F^p_f \wedge DF_f; f\boldsymbol{c}]$ and further by Theorem 10, since $I \cup I^p_f \models$

$\mathrm{SM}[F; p\boldsymbol{c}] \leftrightarrow \mathrm{SM}[F_f^p \wedge DF_f; f\boldsymbol{c}]$, it must be that $I \cup I_f^p \models \mathrm{SM}[F; p\boldsymbol{c}]$. Finally, since the signature of I_f^p does not contain p, we conclude $I \models \mathrm{SM}[F; p\boldsymbol{c}]$.

(b$\Rightarrow$) Assume an interpretation J of the signature of F_f^p is a model of $\mathrm{SM}[F_f^p \wedge FC_f \wedge DF_f; f\boldsymbol{c}]$. Let $I = J_p^f$, where J_p^f denotes the interpretation of the signature F obtainted from J by replacing f^J with the set p^I that consists of the tuples $\langle \xi_1, \ldots, \xi_n \rangle$ for all $\xi_1, \ldots, \xi_n$ from the universe of J such that $f(\xi_1, \ldots, \xi_n) = 1$. By definition of I, $I \cup J \models \forall \boldsymbol{x}(f(\boldsymbol{x}) = 1 \leftrightarrow p(\boldsymbol{x}))$. Now, since $J \models \mathrm{SM}[F_f^p \wedge FC_f \wedge DF_f; f\boldsymbol{c}]$ by our assumption, it must be that $I \cup J \models \mathrm{SM}[F_f^p \wedge FC_f \wedge DF_f; f\boldsymbol{c}]$. Since FC_f is comprised of constraints, by Theorem 3, $I \cup J \models \mathrm{SM}[F_f^p \wedge DF_f; f\boldsymbol{c}] \wedge FC_f$. In particular, $I \cup J \models \mathrm{SM}[F_f^p \wedge DF_f; f\boldsymbol{c}]$ and further by Theorem 10, $I \cup J \models \mathrm{SM}[F; p\boldsymbol{c}]$. Finally, since the signature of J does not contain p, we conclude $I \models \mathrm{SM}[F; p\boldsymbol{c}]$.

(b$\Leftarrow$) Take any I such that $J = I_f^p$ and $I \models \mathrm{SM}[F; p\boldsymbol{c}]$. By definition of I_f^p, $I \cup J \models \forall \boldsymbol{x}(f(\boldsymbol{x}) = 1 \leftrightarrow p(\boldsymbol{x}))$. Now, since $I \models \mathrm{SM}[F; p\boldsymbol{c}]$ by our assumption, it must be that $I \cup J \models \mathrm{SM}[F; p\boldsymbol{c}]$ and further by Theorem 10, since $I \cup J \models \mathrm{SM}[F; p\boldsymbol{c}] \leftrightarrow \mathrm{SM}[F_f^p \wedge DF_f; f\boldsymbol{c}]$, it must be that $I \cup J \models \mathrm{SM}[F_f^p \wedge DF_f; f\boldsymbol{c}]$. Since the signature of I does not contain f, we conclude $J \models \mathrm{SM}[F_f^p \wedge DF_f; f\boldsymbol{c}]$. Finally, since by definition of I_f^p, $J \models FC_f$, and since FC_f is comprised of constraints, by Theorem 3 we conclude $J \models \mathrm{SM}[F_f^p \wedge FC_f \wedge DF_f; f\boldsymbol{c}]$ ∎

6.4.4 *Proof of Theorem 11*

Theorem 11 *A complete set of literals is an answer set of* Π *in the sense of Gelfond* et al. *(1991) iff it is a stable model of* Π^{mv} *in the sense of Bartholomew and Lee (2012).*

Proof. Let I be the interpretation formed from including all of the literals from X and all the assignments from the multi-valued view of X. Let us denote the set of all predicate symbols from X as $\boldsymbol{p}$ and their negative counterparts as $\sim\!\boldsymbol{p}$ and all of the function symbols from the multi-valued view of X as $\boldsymbol{b}$. Clearly I satisfies

$$\forall \boldsymbol{x}((p(\boldsymbol{x}) \leftrightarrow b(\boldsymbol{x})\!=\!1) \wedge (\sim\!p(\boldsymbol{x}) \leftrightarrow b(\boldsymbol{x})\!=\!0)),$$

for each $p \in \boldsymbol{p}$ and the corresponding $b \in \boldsymbol{b}$. From this and since X is complete, it follows that $I \models BC_b$ for each $b \in \boldsymbol{b}$. Thus, we can apply Theorem 12 (multiple times) to conclude that $\mathrm{SM}[\Pi^{FOL};\ \boldsymbol{p} \sim\!\boldsymbol{p}] \leftrightarrow \mathrm{SM}[(\Pi^{mv})^{FOL};\ \boldsymbol{b}]$.

6.4.5 *Proof of Theorem 12*

Theorem 12 *Let* $\boldsymbol{c}$ *be a set predicate and function constants, and let* F *be a* $\boldsymbol{c}$*-plain formula. Formulas*

$$\forall \boldsymbol{x}((p(\boldsymbol{x}) \leftrightarrow b(\boldsymbol{x})\!=\!1) \wedge (\sim\!p(\boldsymbol{x}) \leftrightarrow b(\boldsymbol{x})\!=\!0)),$$

and BC_b *entail*

$$SM[F;\ p, \sim\!p,\ \boldsymbol{c}] \leftrightarrow SM[F_b^{(p,\sim p)};\ b,\ \boldsymbol{c}]\ .$$

Proof. For any interpretation $\mathcal{I} = \langle I, X \rangle$ of signature $\sigma \supseteq \{b, p, \boldsymbol{c}\}$ satisfying (6.9), it is clear that $\mathcal{I} \models F$ iff $\mathcal{I} \models F_b^{p\sim p}$ since $F_b^{p\sim p}$ is simply the result of replacing all $p(\boldsymbol{t})$ with $b(\boldsymbol{t}) = 1$ and all $\sim p(\boldsymbol{t})$ with $b(\boldsymbol{t}) = 0$. Thus it only remains to be shown that $\mathcal{I} \models \neg\exists\widehat{b}, \widehat{\boldsymbol{c}}((\widehat{b}, \widehat{\boldsymbol{c}} < b, \boldsymbol{c}) \wedge (F_b^{(p,\sim p)})^*(\widehat{b}, \widehat{\boldsymbol{c}}))$ iff $\mathcal{I} \models \neg\exists\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}((\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}} <\sim p, p, \boldsymbol{c}) \wedge F^*(\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}))$ or equivalently, $\mathcal{I} \models \exists\widehat{b}, \widehat{\boldsymbol{c}}((\widehat{b}, \widehat{\boldsymbol{c}} < b, \boldsymbol{c}) \wedge (F_b^{(p,\sim p)})^*(\widehat{f}, \widehat{\boldsymbol{c}}))$ iff $\mathcal{I} \models \exists\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}((\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}} <\sim p, p, \boldsymbol{c}) \wedge F^*(\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}))$.

($\Rightarrow$) Assume $\mathcal{I} \models \exists\widehat{b}, \widehat{\boldsymbol{c}}((\widehat{b}, \widehat{\boldsymbol{c}} < b, \boldsymbol{c}) \wedge (F_b^{(p\sim p)})^*(\widehat{b}, \widehat{\boldsymbol{c}}))$. We wish to show that $\mathcal{I} \models \exists\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}((\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}} <\sim p, p, \boldsymbol{c}) \wedge F^*(\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}))$

That is, take any function a of the same arity as b and any list of predicates and functions $\boldsymbol{d}$ of the same length $\boldsymbol{c}$. Now let $\mathcal{I}' = \langle I \cup J_{(a,\boldsymbol{d})}^{(b,\boldsymbol{c})}, X \cup Y_{\boldsymbol{d}}^{\boldsymbol{c}} \rangle$ be from an extended signature $\sigma' = \sigma \cup \{a, q, \boldsymbol{d}\}$ where J is an interpretation of functions from the signature σ and I and J agree on all symbols not occurring in $\{b, \boldsymbol{c}\}$. $J_{(a,\boldsymbol{d})}^{(b,\boldsymbol{c})}$ denotes the interpretation from $\sigma_{(a,\boldsymbol{d})}^{(b,\boldsymbol{c})}$ (the signature obtained from σ by replacing b with a and all elements of $\boldsymbol{c}$ with all elements of $\boldsymbol{d}$) obtained from the interpretation J by replacing b with a and the functions in $\boldsymbol{c}$ with the corresponding functions in $\boldsymbol{d}$. Similarly, $Y_{\boldsymbol{d}}^{\boldsymbol{c}}$ is the interpretation from σ' obtained from the interpretation Y by replacing predicates from $\boldsymbol{c}$ by the corresponding predicates from $\boldsymbol{d}$. We assume

$$\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c} \wedge (F_b^{(p,\sim p)})^*(a, \boldsymbol{d}))$$

and wish to show that there are predicates $\sim q$, q of the same arity as $\sim p$, p such that

$$\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c} \wedge F^*(\sim q, q, \boldsymbol{d})).$$

We define the new predicates $\sim q$, q in terms of b and a as follows:

$$\sim q(\boldsymbol{x}) \leftrightarrow a(\boldsymbol{x}) = 0 \wedge b(\boldsymbol{x}) = 0$$
$$q(\boldsymbol{x}) \leftrightarrow a(\boldsymbol{x}) = 1 \wedge b(\boldsymbol{x}) = 1$$

We first show if $\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c})$ then $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} < \sim p, p, \boldsymbol{c})$:

Observe that from the definition of $\sim q$ and q, it follows that $\mathcal{I}' \models \forall \boldsymbol{x}(\sim q(\boldsymbol{x}) \rightarrow b(\boldsymbol{x}) = 0) \wedge \forall \boldsymbol{x}(q(\boldsymbol{x}) \rightarrow b(\boldsymbol{x}) = 1)$ and from (6.9), this is equivalent to $\mathcal{I}' \models \forall \boldsymbol{x}(\sim q(\boldsymbol{x}) \rightarrow \sim p(\boldsymbol{x})) \wedge \forall \boldsymbol{x}(q(\boldsymbol{x}) \rightarrow p(\boldsymbol{x}))$ or simply $\mathcal{I}' \models \sim q, q \leq \sim p, p$. Thus, since $\mathcal{I}' \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$, it follows that $\mathcal{I}' \models \sim q, q, \boldsymbol{d}^{pred} \leq \sim p, p, \boldsymbol{c}^{pred}$.

Case 1: $\mathcal{I}' \models \forall \boldsymbol{x}(b(\boldsymbol{x}) = a(\boldsymbol{x}))$.

In this case it then must be that $\mathcal{I}' \models \boldsymbol{d} \neq \boldsymbol{c}$. Thus it follows that $\mathcal{I}' \models \sim q, q, \boldsymbol{d} \neq \sim p, p, \boldsymbol{c}$. Consequently we conclude that

$$\mathcal{I}' \models (\sim q, q, \boldsymbol{d}^{pred} \leq \sim p, p, \boldsymbol{c}^{pred}) \wedge \sim q, q, \boldsymbol{d} \neq \sim p, p, \boldsymbol{c}$$

or simply, $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} < \sim p, p, \boldsymbol{c})$.

Case 2: $\mathcal{I}' \models \neg \forall \boldsymbol{x}, y(b(\boldsymbol{x}) = a(\boldsymbol{x}))$.

That is, since $\mathcal{I}' \models BC_b$, there is some list of object names $\boldsymbol{t}$ such that either $\mathcal{I}' \models b(\boldsymbol{t}) = 0 \wedge a(\boldsymbol{t}) \neq 0$ or $\mathcal{I}' \models b(\boldsymbol{t}) = 1 \wedge a(\boldsymbol{t}) \neq 1$.

Subcase 1: $\mathcal{I}' \models b(\boldsymbol{t}) = 0 \wedge a(\boldsymbol{t}) \neq 0$

By (6.9), $\mathcal{I}' \models \sim p(\boldsymbol{t})$ and by definition of $\sim q$, $\mathcal{I}' \models \neg \sim q(\boldsymbol{t})$ so $\mathcal{I}' \models \sim q \neq \sim p$.

Subcase 2: $\mathcal{I}' \models b(\boldsymbol{t}) = 1 \wedge a(\boldsymbol{t}) \neq 1$

By (6.9), $\mathcal{I}' \models p(\boldsymbol{t})$ and by definition of q, $\mathcal{I}' \models \neg q(\boldsymbol{t})$ so $\mathcal{I}' \models q \neq p$.

Therefore, no matter which subcase holds, we have $\sim q, q \neq \sim p, p$ and thus $\sim q, q, \boldsymbol{d} \neq \sim p, p, \boldsymbol{c}$. Consequently we conclude

$$\mathcal{I}' \models (\sim q, q, \boldsymbol{d}^{pred} \leq \sim p, p, \boldsymbol{c}^{pred}) \wedge \sim q, q, \boldsymbol{d} \neq \sim p, p, \boldsymbol{c}$$

or simply, $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} < \sim p, p, \boldsymbol{c})$.

We now show by induction that $\mathcal{I}' \models F^*(\sim q, q, \boldsymbol{d})$:

Case 1: F is an atomic formula not containing p.

$F_b^{(p,\sim p)}$ is exactly F thus $(F_b^{(p,\sim p)})^*(a, \boldsymbol{d})$ is exactly $F^*(\sim q, q, \boldsymbol{d})$ so certainly the claim holds.

Case 2: F is $\sim p(\boldsymbol{t})$, where $\boldsymbol{t}$ contains no intensional function constants.

$F^*(\sim q, q, \boldsymbol{d})$ is $\sim q(\boldsymbol{t})$.

$F_b^{(p,\sim p)}$ is $b(\boldsymbol{t}) = 0$.

$(F_b^{(p,\sim p)})^*(a, \boldsymbol{d})$ is $b(\boldsymbol{t}) = 0 \wedge a(\boldsymbol{t}) = 0$.

By the definition of $\sim q$, it is clear that $\mathcal{I}' \models F^*(\sim q, q, \boldsymbol{d})$ so certainly the claim holds.

Case 3: F is $p(\boldsymbol{t})$, where $\boldsymbol{t}$ contains no intensional function constants.

$F^*(\sim q, q, \boldsymbol{d})$ is $q(\boldsymbol{t})$.

$F_b^{(p,\sim p)}$ is $b(\boldsymbol{t}) = 1$.

$(F_b^{(p,\sim p)})^*(a, \boldsymbol{d})$ is $b(\boldsymbol{t}) = 1 \wedge a(\boldsymbol{t}) = 1$.

By the definition of q, it is clear that $\mathcal{I}' \models F^*(\sim q, q, \boldsymbol{d})$ so certainly the claim holds.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.

By I.H. on G and H.

Case 5: F is $G \rightarrow H$.

By I.H. on G and H.

Case 6: F is $Q\boldsymbol{x}G(\boldsymbol{x})$ where $Q \in \{\forall, \exists\}$.

By I.H. on G.

($\Leftarrow$) Assume $\mathcal{I} \models \exists\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}((\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}} <\sim p, p, \boldsymbol{c}) \wedge F^*(\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}))$. We wish to show that $\mathcal{I} \models \exists\widehat{b}, \widehat{\boldsymbol{c}}((\widehat{b}, \widehat{\boldsymbol{c}} < b, \boldsymbol{c}) \wedge (F_b^{(p,\sim p)})^*(\widehat{b}, \widehat{\boldsymbol{c}}))$

That is, take any predicates $\sim q$, q of the same arity as $\sim p$, p and any list of predicates and functions $\boldsymbol{d}$ of the same length as $\boldsymbol{c}$ and let $\mathcal{I}' = \langle I \cup J_{(a,\boldsymbol{d})}^{(b,\boldsymbol{c})}, X \cup Y_{\boldsymbol{d}}^{\boldsymbol{c}}\rangle$ is defined as before. We assume

$$\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c} \wedge F^*(\sim q, q, \boldsymbol{d}))$$

and wish to show that there is a function a of the same arity as b such that

$$\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c} \wedge (F_b^{(p,\sim p)})^*(a, \boldsymbol{d})).$$

We define the new function a in terms of $\sim p$, p, $\sim q$, and q as follows:

$$\mathcal{I}' \models a(\boldsymbol{x}) = 1 \text{ iff } \mathcal{I}' \models ((p(\boldsymbol{x}) \wedge q(\boldsymbol{x})) \vee (\sim p(\boldsymbol{x}) \wedge \neg \sim q(\boldsymbol{x})))$$
$$\mathcal{I}' \models a(\boldsymbol{x}) = 0 \text{ iff } \mathcal{I}' \models\leftrightarrow ((\sim p(\boldsymbol{x}) \wedge \sim q(\boldsymbol{x})) \vee (p(\boldsymbol{x}) \wedge \neg q(\boldsymbol{x})))$$

Note that since $\mathcal{I}' \models (6.9)$, $\mathcal{I}' \models BC_b$ and $\mathcal{I}' \models \sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c}$ this is a well-defined function. This is because $\mathcal{I}' \models (6.9)$ and $\mathcal{I}' \models BC_b$ guarantee that $\mathcal{I}'$ is complete on p. In addition to this, $\mathcal{I}' \models \sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c}$ guarantees that the four cases covered in this definition are the only ones possible; for any given $\boldsymbol{t}$ exactly one of $p(\boldsymbol{t})$ and $\sim p(\boldsymbol{t})$ is true. Wlog, assume $p(\boldsymbol{t})$ then $\mathcal{I}' \models \sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c}$ gives us that $\sim q(\boldsymbol{t})$ must be false and $q(\boldsymbol{t})$ may be true or false. The other two cases are symmetric by considering when $\sim p(\boldsymbol{t})$ is true.

We first show if $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c})$ then $\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c})$:

Observe that $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c})$ by definition entails $\mathcal{I}' \models (\sim q, q, \boldsymbol{d}^{pred} \leq\sim$

$p, p, \boldsymbol{c}^{pred})$ and further by definition, $\mathcal{I}' \models (\boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred})$ and then since b and a are not predicates, $\mathcal{I}' \models ((a, \boldsymbol{d})^{pred} \leq (b, \boldsymbol{c})^{pred})$.

Case 1: $\mathcal{I}' \models \forall \boldsymbol{x}(p(\boldsymbol{x}) \leftrightarrow q(\boldsymbol{x})) \wedge \forall \boldsymbol{x}({\sim}p(\boldsymbol{x}) \leftrightarrow {\sim}q(\boldsymbol{x}))$.
In this case, $\mathcal{I}' \models ({\sim}p, p = {\sim}q, q)$ so for it to be the case that $\mathcal{I}' \models ({\sim}q, q, \boldsymbol{d} < {\sim}p, p, \boldsymbol{c})$, it must be that $\mathcal{I}' \models \neg(\boldsymbol{c} = \boldsymbol{d})$. It then follows that $\mathcal{I}' \models \neg(b, \boldsymbol{c} = a, \boldsymbol{d})$. Consequently in this case, $\mathcal{I}' \models ((a, \boldsymbol{d})^{pred} \leq (b, \boldsymbol{c})^{pred}) \wedge \neg(b, \boldsymbol{c} = a, \boldsymbol{d})$ or simply $\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c})$.

Case 2: $\mathcal{I}' \models \neg(\forall \boldsymbol{x}y(p(\boldsymbol{x}) \leftrightarrow q(\boldsymbol{x})) \wedge \forall \boldsymbol{x}({\sim}p(\boldsymbol{x}) \leftrightarrow {\sim}q(\boldsymbol{x})))$.
Since $\mathcal{I}' \models {\sim}q, q < {\sim}p, p$ and $\mathcal{I}' \models (6.9)$ and since $\mathcal{I}'$ is complete on p, there is some list of object names $\boldsymbol{t}$ such that either $\mathcal{I}' \models p(\boldsymbol{t}) \wedge \neg q(\boldsymbol{t})$ or $\mathcal{I}' \models {\sim}p(\boldsymbol{t}) \wedge \neg {\sim}q(\boldsymbol{t})$.
Subcase 1: $\mathcal{I}' \models p(\boldsymbol{t}) \wedge \neg q(\boldsymbol{t})$.
By (6.9), $\mathcal{I}' \models b(\boldsymbol{t}) = 1$ and by definition of a, $\mathcal{I}' \models a(\boldsymbol{t}) = 0$. Thus, $\mathcal{I}' \models a \neq b$. Consequently, in this case $\mathcal{I}' \models ((a, \boldsymbol{d})^{pred} \leq (b, \boldsymbol{c})^{pred}) \wedge \neg(b, \boldsymbol{c} = a, \boldsymbol{d})$ or simply $\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c})$.
Subcase 2: $\mathcal{I}' \models {\sim}p(\boldsymbol{t}) \wedge \neg {\sim}q(\boldsymbol{t})$.
By (6.9), $\mathcal{I}' \models b(\boldsymbol{t}) = 0$ and by definition of a, $\mathcal{I}' \models a(\boldsymbol{t}) = 1$. Thus, $\mathcal{I}' \models a \neq b$. Consequently, in this case $\mathcal{I}' \models ((a, \boldsymbol{d})^{pred} \leq (b, \boldsymbol{c})^{pred}) \wedge \neg(b, \boldsymbol{c} = a, \boldsymbol{d})$ or simply $\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c})$.

We now show by induction that $\mathcal{I}' \models (F_b^{(p,\sim p)})^*(a, \boldsymbol{d})$:

Case 1: F is an atomic formula not containing p.
$F_b^{(p,\sim p)}$ is exactly F thus $(F_b^{(p,\sim p)})^*(a, \boldsymbol{d})$ is exactly $F^*({\sim}q, q, \boldsymbol{d})$ so certainly the claim holds.

Case 2: F is $\sim p(\boldsymbol{t})$.

$F^*(q, \boldsymbol{d})$ is $\sim q(\boldsymbol{t})$.

$F_b^{(p,\sim p)}$ is $b(\boldsymbol{t}) = 0$.

$(F_b^{(p,\sim p)})^*(a, \boldsymbol{d})$ is $b(\boldsymbol{t}) = 0 \wedge a(\boldsymbol{t}) = 0$.

By (6.9), $\mathcal{I}' \models b(\boldsymbol{t}) = 0$. By definition of a, $\mathcal{I}' \models a(\boldsymbol{t}) = 0$.

Case 3: F is $p(\boldsymbol{t})$.

$F^*(q, \boldsymbol{d})$ is $q(\boldsymbol{t})$.

$F_b^{(p,\sim p)}$ is $b(\boldsymbol{t}) = 1$.

$(F_b^{(p,\sim p)})^*(a, \boldsymbol{d})$ is $b(\boldsymbol{t}) = 1 \wedge a(\boldsymbol{t}) = 1$.

By (6.9), $\mathcal{I}' \models b(\boldsymbol{t}) = 1$. By definition of a, $\mathcal{I}' \models a(\boldsymbol{t}) = 1$.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.

By I.H. on G and H.

Case 5: F is $G \to H$.

By I.H. on G and H.

Case 6: F is $Q\boldsymbol{x}G(\boldsymbol{x})$ where $Q \in \{\forall, \exists\}$.

By I.H. on G. ∎

Corollary 2 *For any formula F and any interpretation I of the signature of F that is complete on p, (a) I is a model of SM$[F;\ p, \sim p,\ \boldsymbol{c}]$ iff $I^{(p,\ \sim p)}_{b}$ is a model of SM$[F^{(p,\ \sim p)}_{b} \wedge BC_b;\ b,\ \boldsymbol{c}]$. (b) An interpretation J of the signature of $F^{(p,\ \sim p)}_{b}$ is a model of SM$[F^{(p,\ \sim p)}_{b} \wedge BC_b;\ b,\ \boldsymbol{c}]$ iff $J = I^{(p,\ \sim p)}_{b}$ for some model I of SM$[F; p, \sim p,\ \boldsymbol{c}]$.*

Proof. For two interpretations I of signature σ_1 and J of signature σ_2, by $I \cup J$ we denote the interpretation of signature $\sigma_1 \cup \sigma_2$ and universe $|I| \cup |J|$ that interprets all symbols occurring only in σ_1 in the same way I does and similarly for σ_2 and J. For symbols appearing in both σ_1 and σ_2, I must interpret these the same as J does, in which case $I \cup J$ also interprets the symbol in this way.

(a$\Rightarrow$) Assume $I \models 1 \neq 0$ and $I \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$. Since $I \models 1 \neq 0$, $I \cup I^{(p \sim p)}_{b} \models 1 \neq 0$ since by definition of $I^{(p \sim p)}_{b}$, I and $I^{(p \sim p)}_{b}$ share the same universe. By definition of $I^{(p \sim p)}_{b}$, $I \cup I^{(p \sim p)}_{b} \models (6.9)$. Therefore, since I is complete on p and by (6.9), $I \cup I^{(p \sim p)}_{b} \models BC_b$. Thus by Theorem 12, $I \cup I^{(p \sim p)}_{b} \models \mathrm{SM}[F^{p \sim p}_{b} \wedge BC_b;\ b\ \boldsymbol{c}] \leftrightarrow \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$.

Since we assume $I \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$, it is the case that $I \cup I^{(p \sim p)}_{b} \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$ and thus it must be the case that $I \cup I^{(p \sim p)}_{b} \models \mathrm{SM}[F^{(p \sim p)}_{b};\ b,\ \boldsymbol{c}]$. Since $I \cup I^{(p \sim p)}_{b} \models BC_b$ and BC_b is a constraint, $I \cup I^{(p \sim p)}_{b} \models \mathrm{SM}[F^{p \sim p}_{b} \wedge BC_b;\ b,\ \boldsymbol{c}]$. Therefore since the signature of I does not contain b, we conclude $I^{(p \sim p)}_{b} \models \mathrm{SM}[F^{(p \sim p)}_{b} \wedge BC_b;\ b,\ \boldsymbol{c}]$.

(a$\Leftarrow$) Assume $I^{(p \sim p)}_{b} \models \mathrm{SM}[F^{(p \sim p)}_{b} \wedge BC_b; b, \boldsymbol{c}] \wedge (1 \neq 0)$. Since $I^{(p \sim p)}_{b} \models 1 \neq 0$, $I \cup I^{(p \sim p)}_{b} \models 1 \neq 0$ since by definition of $I^{(p \sim p)}_{b}$, I and $I^{(p \sim p)}_{b}$ share the same universe. By definition of $I^{(p \sim p)}_{b}$, $I \cup I^{(p \sim p)}_{b} \models (6.9)$. Since we assume $I^{(p \sim p)}_{b} \models \mathrm{SM}[F^{(p \sim p)}_{b} \wedge BC_b; b, \boldsymbol{c}]$, it follows that $I^{(p \sim p)}_{b} \models BC_b$. Thus by Theorem 12, $I \cup I^{(p \sim p)}_{b} \models \mathrm{SM}[F^{(p \sim p)}_{b} \wedge BC_b; b, \boldsymbol{c}] \leftrightarrow \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$.

Since we assume $I^{(p\sim p)}_{b} \models \mathrm{SM}[F^{(p\sim p)}_{b} \wedge BC_b; b, \boldsymbol{c}]$, it is the case that $I \cup I^{(p\sim p)}_{b} \models \mathrm{SM}[F^{(p\sim p)}_{b} \wedge BC_b; b, \boldsymbol{c}]$ and thus since BC_b is a constraint, it follows that $I \cup I^{(p\sim p)}_{b} \models \mathrm{SM}[F^{(p\sim p)}_{b}; b, \boldsymbol{c}]$. It then follows that $I \cup I^{(p\sim p)}_{b} \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$. However since the signature of $I^{(p\sim p)}_{b}$ does not contain p, we conclude $I \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$.

(b$\Rightarrow$) Assume $J \models 1 \neq 0$ and $J \models \mathrm{SM}[F^{(p\sim p)}_{b} \wedge BC_b;\ b\ \boldsymbol{c}]$. Let $I = J^{\ b}_{(p\sim p)}$ where $J^{\ b}_{(p\sim p)}$ denotes the interpretation of the signature of $F^{(p\sim p)}_{b} \wedge BC_b$ obtained from J by replacing the boolean function b with the predicate p such that

$I \models p^I(\xi_1, \dots, \xi_k)$ for all tuples such that $b^I(\xi_1, \dots, \xi_k) = 1$ and,

$I \models \sim p^I(\xi_1, \dots, \xi_k)$ for all tuples such that $b^I(\xi_1, \dots, \xi_k) = 0$.

Since $J \models BC_b$, this is a well-defined function.

Clearly, $J = I^{(p\sim p)}_{b}$ so it only remains to be shown that $I \models \mathrm{SM}[F; p, \sim p,\ \boldsymbol{c}]$.

Since I and J have the same universe and $J \models 1 \neq 0$, it follows that $I \cup J \models 1 \neq 0$. Also by the definition of $J^{\ b}_{(p\sim p)}$ $I \cup J \models$ (6.9). Also, since $J \models BC_b$, it follows that $I \cup J \models BC_b$.Thus by Theorem 12, $I \cup J \models \mathrm{SM}[F^{p\sim p}_{b};\ b,\ \boldsymbol{c}] \leftrightarrow \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$.

Since we assume $J \models \mathrm{SM}[F^{(p\sim p)}_{b} \wedge BC_b;\ b,\ \boldsymbol{c}]$, it is the case that $I \cup J \models \mathrm{SM}[F^{(p\sim p)}_{b} \wedge BC_b;\ b,\ \boldsymbol{c}]$ and since BC_b is a constraint, $I \cup J \models \mathrm{SM}[F^{(p\sim p)}_{b};\ b,\ \boldsymbol{c}]$. Thus it must be the case that $I \cup J \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$. Now since the signature of J does not contain p, we conclude $I \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$.

(b$\Leftarrow$)Take any I such that $J = I^{(p\sim p)}_{b}$ and $I \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$. Since $J \models 1 \neq 0$ and I and J share the same universe, $I \cup J \models 1 \neq 0$. By definition of $J = I^{(p\sim p)}_{b}$, $I \cup J \models$ (6.9). Since I is complete on p and $I \cup J \models$ (6.9), it follows that $I \cup I^b_p \models BC_b$. Thus by Theorem 12, $I \cup J \models \mathrm{SM}[F^{p\sim p}_{b};\ b,\ \boldsymbol{c}] \leftrightarrow \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$

Since we assume $I \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$, it is the case that $I \cup J \models \mathrm{SM}[F; p, \sim p, \boldsymbol{c}]$

and thus it must be the case that $I \cup J \models \mathrm{SM}[F^{p\,\sim p}_{\;b};\ b,\ \boldsymbol{c}]$. Since BC_b is a constraint, it then follows that $I \cup J \models \mathrm{SM}[F^{p\,\sim p}_{\;b} \wedge BC_b;\ b,\ \boldsymbol{c}]$. However since the signature of I does not contain b, we conclude $J \models \mathrm{SM}[F^{p\,\sim p}_{\;b} \wedge BC_b;\ b,\ \boldsymbol{c}]$. ∎

6.4.7 Proof of Theorem 13

Theorem 13 *Let $\boldsymbol{c}$ be a set of predicate and function constants, let b be a function constant, and let F be a $(b, \boldsymbol{c})$-plain formula such that every atomic formula containing b has the form $b(\boldsymbol{t}) = 1$ or $b(\boldsymbol{t}) = 0$. Formulas (6.9) and BC_b entail*

$$SM[F;\ b, \boldsymbol{c}] \leftrightarrow SM[F^b_{(p,\sim p)};\ p, \sim p, \boldsymbol{c}]\ .$$

Proof.

For any interpretation $\mathcal{I} = \langle I, X\rangle$ of signature $\sigma \supseteq \{b, p, \boldsymbol{c}\}$ satisfying (6.9) and BC_b, it is clear that $\mathcal{I} \models F$ iff $\mathcal{I} \models F^b_{(p,\sim p)}$ since $F^b_{(p,\sim p)}$ is simply the result of replacing all $b(\boldsymbol{x}) = 1$ with $p(\boldsymbol{x})$ and all $b(\boldsymbol{x}) = 0$ with $\sim p(\boldsymbol{x})$. Thus it only remains to be shown that $\mathcal{I} \models \neg\exists\widehat{b}, \widehat{\boldsymbol{c}}((\widehat{b}, \widehat{\boldsymbol{c}} < b, \boldsymbol{c}) \wedge F^*(\widehat{b}, \widehat{\boldsymbol{c}}))$ iff $\mathcal{I} \models \neg\exists\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}((\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}} <\sim p, p, \boldsymbol{c}) \wedge (F^b_{(p,\sim p)})^*(\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}))$ or equivalently, $\mathcal{I} \models \exists\widehat{b}, \widehat{\boldsymbol{c}}((\widehat{b}, \widehat{\boldsymbol{c}} < b, \boldsymbol{c}) \wedge F^*(\widehat{f}, \widehat{\boldsymbol{c}}))$ iff $\mathcal{I} \models \exists\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}((\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}} <\sim p, p, \boldsymbol{c}) \wedge (F^b_{(p,\sim p)})^*(\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}))$.

($\Rightarrow$) Assume $\mathcal{I} \models \exists\widehat{b}, \widehat{\boldsymbol{c}}((\widehat{b}, \widehat{\boldsymbol{c}} < b, \boldsymbol{c}) \wedge F^*(\widehat{b}, \widehat{\boldsymbol{c}}))$. We wish to show that $\mathcal{I} \models \exists\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}((\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}} <\sim p, p, \boldsymbol{c}) \wedge (F^b_{(p,\sim p)})^*(\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}))$

That is, take any function a of the same arity as b and any list of predicates and functions $\boldsymbol{d}$ of the same length $\boldsymbol{c}$. Now let $\mathcal{I}' = \langle I \cup J^{(b,\boldsymbol{c})}_{(a,\boldsymbol{d})}, X \cup Y^{\boldsymbol{c}}_{\boldsymbol{d}}\rangle$ be from an extended signature $\sigma' = \sigma \cup \{a, q, \boldsymbol{d}\}$ where J is an interpretation of functions from the signature σ and I and J agree on all symbols not occurring in $\{b, \boldsymbol{c}\}$. $J^{(b,\boldsymbol{c})}_{(a,\boldsymbol{d})}$ denotes the interpretation from $\sigma^{(b,\boldsymbol{c})}_{(a,\boldsymbol{d})}$ (the signature obtained from σ by replacing b with a and all elements of $\boldsymbol{c}$ with all elements of $\boldsymbol{d}$) obtained from the interpretation J by replacing b with a and the functions in $\boldsymbol{c}$ with the corresponding functions in $\boldsymbol{d}$. Similarly, $Y^{\boldsymbol{c}}_{\boldsymbol{d}}$ is the interpretation from σ' obtained from the interpretation Y by replacing predicates from $\boldsymbol{c}$ by the corresponding predicates from $\boldsymbol{d}$. We assume

$$\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c} \wedge F^*(a, \boldsymbol{d}))$$

and wish to show that there are predicates $\sim q$, q of the same arity as $\sim p$, p such that

$$\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c} \wedge (F^b_{(p,\sim p)})^*(\sim q, q, \boldsymbol{d})).$$

We define the new predicates $\sim q$, q in terms of b and a as follows:

$$\sim q(\boldsymbol{x}) \leftrightarrow a(\boldsymbol{x}) = 0 \wedge b(\boldsymbol{x}) = 0$$
$$q(\boldsymbol{x}) \leftrightarrow a(\boldsymbol{x}) = 1 \wedge b(\boldsymbol{x}) = 1$$

We first show if $\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c})$ then $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c})$:
Observe that from the definition of $\sim q$ and q, it follows that $\mathcal{I}' \models \forall \boldsymbol{x}(\sim q(\boldsymbol{x}) \rightarrow b(\boldsymbol{x}) = 0) \wedge \forall \boldsymbol{x}(q(\boldsymbol{x}) \rightarrow b(\boldsymbol{x}) = 1)$ and from (6.9), this is equivalent to $\mathcal{I}' \models \forall \boldsymbol{x}(\sim q(\boldsymbol{x}) \rightarrow \sim p(\boldsymbol{x})) \wedge \forall \boldsymbol{x}(q(\boldsymbol{x}) \rightarrow p(\boldsymbol{x}))$ or simply $\mathcal{I}' \models \sim q, q \leq \sim p, p$. Thus, since $\mathcal{I}' \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$, it follows that $\mathcal{I}' \models q, \boldsymbol{d}^{pred} \leq p, \boldsymbol{c}^{pred}$.
Case 1: $\mathcal{I}' \models \forall \boldsymbol{x}(b(\boldsymbol{x}) = a(\boldsymbol{x}))$.
In this case it then must be that $\mathcal{I}' \models \boldsymbol{d} \neq \boldsymbol{c}$. Thus it follows that $\mathcal{I}' \models \sim q, q, \boldsymbol{d} \neq \sim p, p, \boldsymbol{c}$. Consequently we conclude that

$$\mathcal{I}' \models (\sim q, q, \boldsymbol{d}^{pred} \leq \sim p, p, \boldsymbol{c}^{pred}) \wedge \sim q, q, \boldsymbol{d} \neq \sim p, p, \boldsymbol{c}$$

or simply, $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c})$.

Case 2: $\mathcal{I}' \models \neg\forall \boldsymbol{x}y(b(\boldsymbol{x}) = a(\boldsymbol{x}))$.

That is, since $\mathcal{I}' \models BC_b$, there is some list of object names $\boldsymbol{t}$ such that either $\mathcal{I}' \models b(\boldsymbol{t}) = 0 \wedge a(\boldsymbol{t}) \neq 0$ or $\mathcal{I}' \models b(\boldsymbol{t}) = 1 \wedge a(\boldsymbol{t}) \neq 1$.

Subcase 1: $\mathcal{I}' \models b(\boldsymbol{t}) = 0 \wedge a(\boldsymbol{t}) \neq 0$

By (6.9), $\mathcal{I}' \models \sim\!p(\boldsymbol{t})$ and by definition of $\sim\!q$, $\mathcal{I}' \models \neg \sim\!q(\boldsymbol{t})$ so $\mathcal{I}' \models \sim\!q \neq \sim\!p$.

Subcase 2: $\mathcal{I}' \models b(\boldsymbol{t}) = 1 \wedge a(\boldsymbol{t}) \neq 1$

By (6.9), $\mathcal{I}' \models p(\boldsymbol{t})$ and by definition of q, $\mathcal{I}' \models \neg q(\boldsymbol{t})$ so $\mathcal{I}' \models q \neq p$.

Therefore, no matter which subcase holds, we have $\sim\!q, q \neq \sim\!p, p$ and thus $\sim\!q, q, \boldsymbol{d} \neq \sim\!p, p, \boldsymbol{c}$. Consequently we conclude

$$\mathcal{I}' \models (\sim\!q, q, \boldsymbol{d}^{pred} \leq \sim\!p, p, \boldsymbol{c}^{pred}) \wedge \sim\!q, q, \boldsymbol{d} \neq \sim\!p, p, \boldsymbol{c}$$

or simply, $\mathcal{I}' \models (\sim\!q, q, \boldsymbol{d} < \sim\!p, p, \boldsymbol{c})$.

We now show by induction that $\mathcal{I}' \models (F^b_{(p,\sim p)})^*(\sim\!q, q, \boldsymbol{d})$:

Case 1: F is an atomic formula not containing b.

$F^b_{(p,\sim p)}$ is exactly F thus $F^*(a, \boldsymbol{d})$ is exactly $(F^b_{(p,\sim p)})^*(\sim\!q, q, \boldsymbol{d})$ so certainly the claim holds.

Case 2: F is $b(\boldsymbol{t}) = 0$.

$F^*(a, \boldsymbol{d})$ is $b(\boldsymbol{t}) = 0 \wedge a(\boldsymbol{t}) = 0$.

$F^b_{(p,\sim p)}$ is $\sim\!p(\boldsymbol{t})$.

$(F^b_{(p,\sim p)})^*(\sim\!q, q, \boldsymbol{d})$ is $\sim\!q(\boldsymbol{t})$.

By the definition of $\sim\!q$, it is clear that $\mathcal{I}' \models (F^b_{(p,\sim p)})^*(\sim\!q, q, \boldsymbol{d})$ so certainly the claim holds.

Case 3: F is $b(\boldsymbol{t}) = 1$.

$F^*(a, \boldsymbol{d})$ is $b(\boldsymbol{t}) = 1 \wedge a(\boldsymbol{t}) = 1$.

$F^b_{(p,\sim p)}$ is $p(\boldsymbol{t})$.

$(F^b_{(p,\sim p)})^*(\sim q, q, \boldsymbol{d})$ is $q(\boldsymbol{t})$.

By the definition of q, it is clear that $\mathcal{I}' \models (F^b_{(p,\sim p)})^*(\sim q, q, \boldsymbol{d})$ so certainly the claim holds.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.

By I.H. on G and H.

Case 5: F is $G \to H$.

By I.H. on G and H.

Case 6: F is $Q\boldsymbol{x}G(\boldsymbol{x})$ where $Q \in \{\forall, \exists\}$.

By I.H. on G.

($\Leftarrow$) Assume $\mathcal{I} \models \exists \widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}((\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}} <\sim p, p, \boldsymbol{c}) \wedge (F^b_{(p,\sim p)})^*(\widehat{\sim p}, \widehat{p}, \widehat{\boldsymbol{c}}))$. We wish to show that $\mathcal{I} \models \exists \widehat{b}, \widehat{\boldsymbol{c}}((\widehat{b}, \widehat{\boldsymbol{c}} < b, \boldsymbol{c}) \wedge F^*(\widehat{b}, \widehat{\boldsymbol{c}}))$

That is, take any predicates $\sim q$, q of the same arity as $\sim p$, p and any list of predicates and functions $\boldsymbol{d}$ of the same length as $\boldsymbol{c}$ and let $\mathcal{I}' = \langle I \cup J^{(b,\boldsymbol{c})}_{(a,\boldsymbol{d})}, X \cup Y^{\boldsymbol{c}}_{\boldsymbol{d}} \rangle$ is defined as before. We assume

$$\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c} \wedge (F^b_{(p,\sim p)})^*(\sim q, q, \boldsymbol{d}))$$

and wish to show that there is a function a of the same arity as b such that

$$\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c} \wedge F^*(a, \boldsymbol{d})).$$

We define the new function a in terms of $\sim p$, p, $\sim q$, and q as follows:

$$a(\boldsymbol{x}) = 1 \leftrightarrow ((p(\boldsymbol{x}) \wedge q(\boldsymbol{x})) \vee (\sim p(\boldsymbol{x}) \wedge \neg \sim q(\boldsymbol{x})))$$

$$a(\boldsymbol{x}) = 0 \leftrightarrow ((\sim p(\boldsymbol{x}) \wedge \sim q(\boldsymbol{x})) \vee (p(\boldsymbol{x}) \wedge \neg q(\boldsymbol{x})))$$

Note that since $\mathcal{I}' \models$ (6.9) and $\mathcal{I}' \models \sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c}$ this is a well-defined function.

We first show if $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c})$ then $\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c})$:
Observe that $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c})$ by definition entails $\mathcal{I}' \models (\sim q, q, \boldsymbol{d}^{pred} \leq\sim p, p, \boldsymbol{c}^{pred})$ and further by definition, $\mathcal{I}' \models (\boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred})$ and then since b and a are not predicates, $\mathcal{I}' \models ((a, \boldsymbol{d})^{pred} \leq (b, \boldsymbol{c})^{pred})$.

Case 1: $\mathcal{I}' \models \forall \boldsymbol{x}(p(\boldsymbol{x}) \leftrightarrow q(\boldsymbol{x})) \wedge \forall \boldsymbol{x}(\sim p(\boldsymbol{x}) \leftrightarrow\sim q(\boldsymbol{x}))$.
In this case, $\mathcal{I}' \models (\sim p, p =\sim q, q)$ so for it to be the case that $\mathcal{I}' \models (\sim q, q, \boldsymbol{d} <\sim p, p, \boldsymbol{c})$, it must be that $\mathcal{I}' \models \neg(\boldsymbol{c} = \boldsymbol{d})$. It then follows that $\mathcal{I}' \models \neg(b, \boldsymbol{c} = a, \boldsymbol{d})$. Consequently in this case, $\mathcal{I}' \models ((a, \boldsymbol{d})^{pred} \leq (b, \boldsymbol{c})^{pred}) \wedge \neg(b, \boldsymbol{c} = a, \boldsymbol{d})$ or simply $\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c})$.

Case 2: $\mathcal{I}' \models \neg(\forall \boldsymbol{x}y(p(\boldsymbol{x}) \leftrightarrow q(\boldsymbol{x})) \wedge \forall \boldsymbol{x}(\sim p(\boldsymbol{x}) \leftrightarrow\sim q(\boldsymbol{x})))$.
Since $\mathcal{I}' \models\sim q, q <\sim p, p$ and $\mathcal{I}' \models$ (6.9), there is some list of object names $\boldsymbol{t}$ such that either $\mathcal{I}' \models p(\boldsymbol{t}) \wedge \neg q(\boldsymbol{t})$ or $\mathcal{I}' \models\sim p(\boldsymbol{t}) \wedge \neg \sim q(\boldsymbol{t})$.
Subcase 1: $\mathcal{I}' \models p(\boldsymbol{t}) \wedge \neg q(\boldsymbol{t})$.
By (6.9) $\mathcal{I}' \models b(\boldsymbol{t}) = 1$ and by definition of a, $\mathcal{I}' \models a(\boldsymbol{t}) = 0$. Thus, $\mathcal{I}' \models a \neq b$. Consequently, in this case $\mathcal{I}' \models ((a, \boldsymbol{d})^{pred} \leq (b, \boldsymbol{c})^{pred}) \wedge \neg(b, \boldsymbol{c} = a, \boldsymbol{d})$ or simply $\mathcal{I}' \models (a, \boldsymbol{d} < b, \boldsymbol{c})$.
Subcase 2: $\mathcal{I}' \models\sim p(\boldsymbol{t}) \wedge \neg \sim q(\boldsymbol{t})$.
By (6.9) $\mathcal{I}' \models b(\boldsymbol{t}) = 0$ and by definition of a, $\mathcal{I}' \models a(\boldsymbol{t}) = 1$. Thus, $\mathcal{I}' \models a \neq b$.

Consequently, in this case $\mathcal{I}' \models ((a,\boldsymbol{d})^{pred} \leq (b,\boldsymbol{c})^{pred}) \wedge \neg(b,\boldsymbol{c} = a,\boldsymbol{d})$ or simply $\mathcal{I}' \models (a,\boldsymbol{d} < b,\boldsymbol{c})$.

We now show by induction that $\mathcal{I}' \models F^*(a,\boldsymbol{d})$:

Case 1: F is an atomic formula not containing b.

$F^b_{(p,\sim p)}$ is exactly F thus $F^*(a,\boldsymbol{d})$ is exactly $(F^b_{(p,\sim p)})^*(\sim q,\boldsymbol{d})$ so certainly the claim holds.

Case 2: F is $b(\boldsymbol{t}) = 0$.

$F^*(a,\boldsymbol{d})$ is $b(\boldsymbol{t}) = 0 \wedge a(\boldsymbol{t}) = 0$.

$F^b_{(p,\sim p)}$ is $\sim p(\boldsymbol{t})$.

$(F^b_{(p,\sim p)})^*(q,\boldsymbol{d})$ is $\sim q(\boldsymbol{t})$.

By (6.9), $\mathcal{I}' \models b(\boldsymbol{t}) = 0$. By definition of a, $\mathcal{I}' \models a(\boldsymbol{t}) = 0$.

Case 3: F is $b(\boldsymbol{t}) = 1$.

$F^*(a,\boldsymbol{d})$ is $b(\boldsymbol{t}) = 1 \wedge a(\boldsymbol{t}) = 1$.

$F^b_{(p,\sim p)}$ is $p(\boldsymbol{t})$.

$(F^b_{(p,\sim p)})^*(q,\boldsymbol{d})$ is $q(\boldsymbol{t})$.

By (6.9), $\mathcal{I}' \models b(\boldsymbol{t}) = 1$. By definition of a, $\mathcal{I}' \models a(\boldsymbol{t}) = 1$.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.

By I.H. on G and H.

Case 5: F is $G \to H$.

By I.H. on G and H.

Case 6: F is $Q\boldsymbol{x}G(\boldsymbol{x})$ where $Q \in \{\forall, \exists\}$.

By I.H. on G. ∎

6.4.8 Proof of Corollary 3

Corollary 3 *Let $\boldsymbol{c}$ be a set of predicate and function constants, let b be a function constant, and let F be a $(b, \boldsymbol{c})$-plain sentence such that every atomic formula containing b has the form $b(\boldsymbol{t}) = 1$ or $b(\boldsymbol{t}) = 0$. (a) A coherent interpretation I of the signature of F is a model of SM$[F \wedge BC_b;\ b, \boldsymbol{c}]$ iff $I^b_{(p,\sim p)}$ is a model of SM$[F^b_{(p,\sim p)};\ p, \sim p, \boldsymbol{c}]$. (b) An interpretation J of the signature of $F^b_{(p,\sim p)}$ is a model of SM$[F^b_{(p,\sim p)}; p, \sim p, \boldsymbol{c}]$ iff $J = I^b_{(p,\sim p)}$ for some model I of SM$[F \wedge BC_b; b, \boldsymbol{c}]$.*

Proof.

For two interpretations I of signature σ_1 and J of signature σ_2, by $I \cup J$ we denote the interpretation of signature $\sigma_1 \cup \sigma_2$ and universe $|I| \cup |J|$ that interprets all symbols occurring only in σ_1 in the same way I does and similarly for σ_2 and J. For symbols appearing in both σ_1 and σ_2, I must interpret these the same as J does, in which case $I \cup J$ also interprets the symbol in this way.

(a$\Rightarrow$) Assume $I \models \text{SM}[F; b, \boldsymbol{c}] \wedge (1 \neq 0)$. Since $I \models 1 \neq 0$, $I \cup I^b_p \models 1 \neq 0$ since by definition of I^b_p, I and I^b_p share the same universe. By definition of I^b_p, $I \cup I^b_p \models$ (6.9). Since we assume $I \models \text{SM}[F \wedge BF_b; b, \boldsymbol{c}]$, it follows that $I \models BF_b$ which further means that $I \models BC_b$ and so $I \cup I^b_p \models BC_b$. Thus by Theorem 13,

$I \cup I_p^b \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}] \leftrightarrow \mathrm{SM}[(F \wedge BF_b)_p^b; p, \sim p, \boldsymbol{c}]$.

Since we assume $I \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}]$, it is the case that $I \cup I_p^b \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}]$ and thus it must be the case that $I \cup I_p^b \models \mathrm{SM}[(F \wedge BF_b)_p^b; p, \sim p, \boldsymbol{c}]$. However since the signature of I does not contain p, we conclude $I_p^b \models \mathrm{SM}[(F \wedge BF_b)_p^b; p, \sim p, \boldsymbol{c}]$.

(a$\Leftarrow$) Assume $I \models 1 \neq 0$ and $I_p^b \models \mathrm{SM}[(F \wedge BF_b)_p^b; p, \sim p, \boldsymbol{c}]$. Since $I \models 1 \neq 0$, $I \cup I_p^b \models 1 \neq 0$ since by definition of I_p^b, I and I_p^b share the same universe. By definition of I_p^b, $I \cup I_p^b \models$ (6.9). Therefore, since $I_p^b \models (BF_b)_p^b$, it follows that $I \models BF_b$ and thus, $I \cup I_p^b \models BC_b$. Thus by Theorem 13, $I \cup I_p^b \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}] \leftrightarrow \mathrm{SM}[(F \wedge BF_b)_p^b; p, \sim p, \boldsymbol{c}]$.

Since we assume $I_p^b \models \mathrm{SM}[(F \wedge BF_b)_p^b; p, \sim p, \boldsymbol{c}]$, it is the case that $I \cup I_p^b \models \mathrm{SM}[(F \wedge BF_b)_p^b; p, \sim p, \boldsymbol{c}]$ and thus it must be the case that $I \cup I_p^b \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}]$. Therefore since the signature of I_p^b does not contain b, we conclude $I \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}]$.

(b$\Rightarrow$) Assume $J \models 1 \neq 0$ and $J \models \mathrm{SM}[(F \wedge BF_b)_p^b; p\boldsymbol{c}]$. Let $I = J_b^p$ where J_b^p denotes the interpretation of the signature of F obtained from J by replacing the predicate p with the boolean function b such that
$b^I(\xi_1, \ldots, \xi_k) = 1$ for all tuples such that $I \models p^I(\xi_1, \ldots, \xi_k)$,
$b^I(\xi_1, \ldots, \xi_k) = 0$ for all tuples such that $I \models \sim p^I(\xi_1, \ldots, \xi_k)$. Since $J \models (BF_b)_p^b$, this is a well-defined function.
Clearly, $J = I_p^b$ so it only remains to be shown that $I \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}]$.

Since I and J have the same universe and $J \models 1 \neq 0$, it follows that $I \cup J \models 1 \neq 0$. Also by the definition of J_b^p $I \cup J \models$ (6.9). Also, since $J \models (BF_b)_p^b$, it follows that $I \models BF_b$ and thus, $I \cup J \models BC_b$. Thus by Theorem 13, $I \cup J \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}] \leftrightarrow \mathrm{SM}[(F \wedge BF_b)_p^b \wedge CC_p; p\boldsymbol{c}]$

Since we assume $J \models \mathrm{SM}[(F \wedge BF_b)_p^b; p, \sim p, \boldsymbol{c}]$, it is the case that $I \cup J \models$

$\mathrm{SM}[(F \wedge BF_b)^b_p; p, \sim p, \boldsymbol{c}]$ and thus it must be the case that $I \cup J \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}]$. Now since the signature of J does not contain b, we conclude $I \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}]$.

(b$\Leftarrow$)Take any I such that $J = I^b_p$ and $I \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}]$. Since $J \models 1 \neq 0$ and I and J share the same universe, $I \cup J \models 1 \neq 0$. By definition of $J = I^b_p$, $I \cup J \models$ (6.9). Since we assume $I \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}]$, it follows that $I \models BF_b$ which further means that $I \models BC_b$ and so $I \cup I^b_p \models BF_b$. Thus by Theorem 13, $I \cup J \models \mathrm{SM}[F \wedge BF_b; b, \boldsymbol{c}] \leftrightarrow \mathrm{SM}[(F \wedge BF_b)^b_p; \sim p, p, \boldsymbol{c}]$

Since we assume $I \models \mathrm{SM}[F; b, \boldsymbol{c}]$, it is the case that $I \cup J \models \mathrm{SM}[F; b, \boldsymbol{c}]$ and thus it must be the case that $I \cup J \models \mathrm{SM}[F^b_{(p,\sim p)} \wedge CC_p; p, \sim p, \boldsymbol{c}]$. However since the signature of I does not contain p, we conclude $J \models \mathrm{SM}[F^b_{(p,\sim p)} \wedge CC_p; p, \sim p, \boldsymbol{c}]$. ∎

Chapter 7

ELIMINATING INTENSIONAL FUNCTIONS IN FAVOR OF INTENSIONAL PREDICATES

7.1 Multi-valued Propositional Formulas

We first consider the simpler task of turning multi-valued propositional formulas into propositional formulas. We show that multi-valued stable model semantics can be viewed as a special case of the propositional stable model semantics. Let σ be a multi-valued signature, and let σ^{prop} be the propositional signature consisting of all propositional atoms $c = v$ where $c \in \sigma$ and $v \in Dom(c)$. For example, for σ in Example 7, σ^{prop} is the set $\{Amount_0 = 0, \dots, Amount_0 = 10,\ Amount_0 = 1, \dots, Amount_1 = 10,\ \mathit{FillUp} = \boldsymbol{t},\ \mathit{FillUp} = \boldsymbol{f}\}$, where each element is understood as a propositional atom. [1] We identify a multi-valued interpretation of σ with the corresponding set of propositional atoms from σ^{prop}. It is clear that a multi-valued interpretation I of signature σ satisfies a multi-valued propositional formula F iff I satisfies F when F is viewed as a propositional formula of signature σ^{prop}. Also, it is not difficult to show that multi-valued formulas can be turned into standard propositional formulas having the same *classical models*. Less obvious is whether such a translation exists while keeping same *stable models*. Theorem 14 below shows such a translation.

Given a multi-valued signature σ, by UC_σ ("Uniqueness Constraint") we denote

[1] We could have included in σ^{prop} different expressions such as $c(v)$ in place of $c = v$. Viewing $c = v$ as both multi-valued atoms and propositional atoms under different signatures simplifies the formal statements.

the conjunction of

$$\bigwedge_{v \neq w \ | \ v,w \in Dom(c)} \neg(c = v \wedge c = w) \tag{7.1}$$

for all $c \in \sigma$, and by EC_σ ("Existence Constraint") we denote the conjunction of

$$\neg\neg \bigvee_{v \in Dom(c)} c = v \ , \tag{7.2}$$

for all $c \in \sigma$. By UEC_σ we denote the conjunction of (10.1) and (7.2) for all $c \in \sigma$.

The following theorem tells us that the functional stable model semantics for multi-valued propositional formulas can be reduced to the stable model semantics for classical propositional formulas in Ferraris (2005). In other words, checking the uniqueness of functions coincides with checking the minimality of propositional atoms under the stable model semantics.

Theorem 14 *Let F be a multi-valued propositional formula of signature σ, which can be also viewed as a propositional formula of signature σ^{prop}.*

(a) If an interpretation I of σ is a multi-valued stable model of F, then I can be viewed as an interpretation of σ^{prop} that is a propositional stable model of $F \wedge UEC_\sigma$ in the sense of Ferraris (2005).

(b) If an interpretation I of σ^{prop} is a propositional stable model of $F \wedge UEC_\sigma$ in the sense of Ferraris (2005), then I can be viewed as an interpretation of σ that is a multi-valued stable model of F.

Example 6 continued UEC_σ is

$$\neg\neg(c{=}1 \vee c{=}2 \vee c{=}3) \wedge \ \neg(c{=}1 \wedge c{=}2)$$
$$\wedge\neg(c{=}2 \wedge c{=}3) \wedge \neg(c{=}1 \wedge c{=}3) \ .$$

Note that the presence of $\neg\neg$ in (7.2) is essential for Theorem 14 to be valid. For instance, consider the signature containing only one constant d whose domain is $\{1, 2\}$ and F to be $\top$. F has no multi-valued stable models, but $F \wedge \neg(d\!=\!1 \wedge d\!=\!2) \wedge (d\!=\!1 \vee d\!=\!2)$ has two propositional stable models: $\{d\!=\!1\}$ and $\{d\!=\!2\}$.

7.2 Eliminating Intensional Functions from $\boldsymbol{c}$-Plain Formulas

We now show how to eliminate intensional functions in favor of intensional predicates. Doing so yields two useful results. First, we can compute models of a formula under the functional stable model semantics using state-of-the-art ASP solvers. Second, results established for the first-order stable model semantics Ferraris *et al.* (2011) can be established for the functional stable model semantics by eliminating the intensional functions.

Unlike the previous chapter, the result is first established for f-plain formulas, and then extended to allow "synonymity" rules.

Recall the definition of f-plain from Section 5.4

For a function constant f, a first-order formula is called *f-plain* if each atomic formula

- does not contain f, or
- is of the form $f(\boldsymbol{t}) = u$ where $\boldsymbol{t}$ is a tuple of terms not containing f, and u is a term not containing f.

For a list $\boldsymbol{f}$ of function constants, we say that F is $\boldsymbol{f}$-plain if F is f-plain for each member f of $\boldsymbol{f}$.

Let F be an f-plain formula, where f is an intensional function constant. Formula F^f_p is obtained from F as follows:

- in the signature of F, replace f with a new intensional predicate constant p of arity $n+1$, where n is the arity of f;

- replace each subformula $f(\boldsymbol{t}) = c$ in F with $p(\boldsymbol{t}, c)$.

By UEC_p we denote the following formulas that enforce the functional image on the predicates:

$$\begin{gathered} \forall \boldsymbol{x} yz(y \neq z \wedge p(\boldsymbol{x}, y) \wedge p(\boldsymbol{x}, z) \to \bot), \\ \neg\neg\forall \boldsymbol{x} \exists y\, p(\boldsymbol{x}, y), \end{gathered} \tag{7.3}$$

where $\boldsymbol{x}$ is a n-tuple of variables, and all variables in $\boldsymbol{x}$, y, and z are pairwise distinct. Note that each formula is a constraint. Clearly, UEC_p is strongly equivalent to

$$\neg\neg\forall \boldsymbol{x} \exists! y\, p(\boldsymbol{x}, y) \tag{7.4}$$

and also classically equivalent to

$$\forall \boldsymbol{x} \exists! y\, p(\boldsymbol{x}, y)\ . \tag{7.5}$$

Example 13 continued

Recall the example from Section 6.2 that describes the effect of a monkey moving. We eliminate the function Loc in favor of an intensional predicate Loc_p to obtain $F^{Loc}_{Loc_p} \wedge UEC_{Loc_p}$, which is the conjunction of the universal closures of the following formulas:

$$\begin{gathered} Loc_p(Monkey, 0, L1), \\ Loc_p(Monkey, 1, L2), \\ Move(Monkey, l, t) \to Loc_p(Monkey, t+1, l), \\ \forall wxyz(y \neq z \wedge Loc_p(w, x, y) \wedge Loc_p(w, x, z) \to \bot), \\ \neg\neg\forall wx \exists y(Loc_p(w, x, y)). \end{gathered}$$

Theorem 15 *For any f-plain formula F, formulas $\forall \boldsymbol{x}y(p(\boldsymbol{x},y) \leftrightarrow f(\boldsymbol{x}) = y)$, $\exists xy(x \neq y)$ entail*

$$SM[F; f\boldsymbol{c}] \leftrightarrow SM[F^f_p; p\boldsymbol{c}].$$

The following corollary shows that there is a simple 1–1 correspondence between the stable models of F and the stable models of $F^f_p \wedge UEC_p$. Recall that the signature of F^f_p is obtained from the signature of F by replacing f with p. For any interpretation I of the signature of F, by I^f_p we denote the interpretation of the signature of F^f_p obtained from I by replacing the function f^I with the set p^I that consists of the tuples

$$\langle \xi_1, \dots, \xi_n, f^I(\xi_1, \dots, \xi_n)\rangle$$

for all $\xi_1, \dots, \xi_n$ from the universe of I.

Corollary 4 *Let F be an f-plain sentence. (a) An interpretation I of the signature of F that satisfies $\exists xy(x \neq y)$ is a model of $SM[F; f\boldsymbol{c}]$ iff I^f_p is a model of $SM[F^f_p;\ p\boldsymbol{c}]$. (b) An interpretation J of the signature of F^f_p that satisfies $\exists xy(x \neq y)$ is a model of $SM[F^f_p \wedge UEC_p;\ p\boldsymbol{c}]$ iff $J = I^f_p$ for some model I of $SM[F; f\boldsymbol{c}]$.*

Theorem 15 and Corollary 4 are similar to Theorem 3 and Corollary 5 from Lifschitz and Yang (2011), which are about eliminating explainable functions in non-monotonic causal logic in favor of explainable predicates.

The method above eliminates only one intensional function constant at a time, but repeated applications can eliminate all intensional functions $\boldsymbol{f}$ from a given $\boldsymbol{f}$-plain formula. This allows us to represent the $\boldsymbol{f}$-plain formula by a logic program.

7.3 Non-$\boldsymbol{c}$-plain formulas

We expect that many domains can be described by $\boldsymbol{f}$-plain formulas, but we know of some concepts where $\boldsymbol{f}$-plain formulas are limited. One limitation is in capturing the many-sorted functional stable model semantics within the nonsorted functional stable model semantics, which will be described in detail in Section 8.4. Another is when we want to express "synonymity" rules Lee *et al.* (2010); Lifschitz and Yang (2011) that have the form

$$B \ \rightarrow \ f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2), \tag{7.6}$$

where f_1, f_2 are intensional function constants in $\boldsymbol{f}$, and $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ are tuples of terms not containing members of $\boldsymbol{f}$. This rule expresses that we believe $f_1(\boldsymbol{t}_1)$ to be "synonymous" to $f_2(\boldsymbol{t}_2)$ under condition B. We can eliminate f_1 and f_2 in favor of predicate constants p_1 and p_2 as follows.

We consider a more general case than an $\boldsymbol{f}$-plain formula. We define a new class of *$\boldsymbol{f}$-plain-syn formulas* in which every atomic formula

- does not contain any member of $\boldsymbol{f}$, or
- is of the form $f(\boldsymbol{t}) = u$ where f is in $\boldsymbol{f}$, symbol $\boldsymbol{t}$ is a tuple of terms not containing any member of $\boldsymbol{f}$, and u is a term not containing any member of $\boldsymbol{f}$, or
- is of the form $f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ where f_1, f_2 are in $\boldsymbol{f}$, symbols $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are tuples of terms not containing any member of $\boldsymbol{f}$.

Example 16 *Consider the Gears World domain in which there are two gears, and one is attached to a motor which turns the gear at 1 revolution per minute. If the gears are moved close together, the gears spin at the same rate. We can describe using* *the following $\boldsymbol{f}$-plain-syn formula.*

$$gear1speed(t) = 1$$

$$Choice(gear2speed(t) = 0)$$

$$Choice(moveGearsTogether(0))$$

$$moveGearsTogether(0) \rightarrow gearsConnected(1)$$

$$gearsConnected(t) \rightarrow gear1speed(t) = gear2speed(t)$$

One stable model of this is I where

$$\begin{array}{cc} gear1speed(0)^I = 1, & gear1speed(1)^I = 1, \\ gear2speed(0)^I = 0, & gear2speed(1)^I = 1, \\ moveGearsTogether(0)^I = \boldsymbol{t}, & moveGearsTogether(1)^I = \boldsymbol{f} \\ gearsConnected(0)^I = \boldsymbol{f}, & gearsConnected(1)^I = \boldsymbol{t}. \end{array}$$

Let F be an $\boldsymbol{f}$-plain-syn formula. The elimination is done by extending the previous method by turning atomic formulas of the form $f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ into

$$\forall y(p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y)),$$

where p_1, p_2 are new intensional predicate constants corresponding to f_1, f_2.

$F^{\boldsymbol{f}}_{\boldsymbol{p}}$ is defined similar to F^f_p except that it applies to the list of symbols.

Theorem 16 *For any $\boldsymbol{f}$-plain-syn formula F, the set of formulas $\forall \boldsymbol{x}y(p(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y)$ for each $f \in \boldsymbol{f}$ and the corresponding p, and $\exists xy(x \neq y)$ entail*

$$SM[F; \boldsymbol{f}\boldsymbol{q}] \leftrightarrow SM[F^{\boldsymbol{f}}_{\boldsymbol{p}}; \boldsymbol{p}\boldsymbol{q}].$$

Unlike in Theorem 15, the elimination in Theorem 16 applies to the list of intensional functions simultaneously. Applying the result of Theorem 10 to $F^{\boldsymbol{f}}_{\boldsymbol{p}}$ results in an $\boldsymbol{f}$-plain formula, and so the composition of these two translations reveals that $\boldsymbol{f}$-plain-syn formula F can actually be transformed into $\boldsymbol{f}$-plain formulas.

Example 16 continued *Using Theorem 16, we can eliminate the functions in the Gears World example in favor of predicates to get the formula F*

$$gear1speed(t, 1)$$

$$Choice(gear2speed(t, 0))$$

$$Choice(moveGearsTogether(0))$$

$$moveGearsTogether(0) \rightarrow gearsConnected(1)$$

$$gearsConnected(t) \rightarrow \forall x(gear1speed(t, x) \leftrightarrow gear2speed(t, x))$$

Then, the interpretation I with $|I| = \{0, 1\}$ satisfying

$$\forall tx(gear1speed(t) = x \leftrightarrow gear1speed(t, x))$$

$$\forall tx(gear2speed(t) = x \leftrightarrow gear2speed(t, x))$$

$$\exists xy(x \neq y)$$

such that both 0 *and* 1 *mapped to themselves and*

$$\begin{array}{cc} gear1speed(0,1)^I = \boldsymbol{t}, & gear1speed(1,1)^I = \boldsymbol{t}, \\ gear2speed(0,0)^I = \boldsymbol{t}, & gear2speed(1,1)^I = \boldsymbol{t}, \\ moveGearsTogether(0)^I = \boldsymbol{t}, & moveGearsTogether(1)^I = \boldsymbol{f} \\ gearsConnected(0)^I = \boldsymbol{f}, & gearsConnected(1)^I = \boldsymbol{t}. \end{array}$$

is a stable model of F.

7.4 Unfolding

In an attempt to relax the syntactic restrictions in the previous two sections, we have investigated transformations that turn non-$\boldsymbol{c}$-plain formulas into $\boldsymbol{c}$-plain formulas. In this section, we present one such method which we call "unfolding".

The process of *unfolding* F *w.r.t.* $\boldsymbol{c}$, denoted by $UF_{\boldsymbol{c}}(F)$, is formally defined as follows.

- If F is of the form $p(t_1, \dots, t_n)$ $(n \geq 0)$ such that $t_{k_1}, \dots, t_{k_j}$ are all the terms in $t_1, \dots, t_n$ that contain some members of $\boldsymbol{c}$, then $UF_{\boldsymbol{c}}(p(t_1, \dots, t_n))$ is

$$\exists x_1 \dots x_j \Big(p(t_1, \dots, t_n)'' \wedge \bigwedge_{1 \leq i \leq j} UF_{\boldsymbol{c}}(t_{k_i} = x_i)\Big)$$

  where $p(t_1, \dots, t_n)''$ is obtained from $p(t_1, \dots, t_n)$ by replacing each t_{k_i} with the variable x_i.

- If F is of the form $f(t_1, \dots, t_n) = t_0$ $(n \geq 0)$ such that $t_{k_1}, \dots, t_{k_j}$ are all the terms in $t_0, \dots, t_n$ that contain some members of $\boldsymbol{c}$, then $UF_{\boldsymbol{c}}(f(t_1, \dots, t_n) = t_0)$ is

$$\exists x_1 \dots x_j \Big((f(t_1, \dots, t_n) = t_0)'' \wedge \bigwedge_{0 \leq i \leq j} UF_{\boldsymbol{c}}(t_{k_i} = x_i)\Big)$$

  where $(f(t_1, \dots, t_n) = t_0)''$ is obtained from $f(t_1, \dots, t_n) = t_0$ by replacing each t_{k_i} with the variable x_i.

- $UF_{\boldsymbol{c}}(F \odot G)$ is $UF_{\boldsymbol{c}}(F) \odot UF_{\boldsymbol{c}}(G)$ where $\odot \in \{\wedge, \vee, \rightarrow\}$.

- $UF_{\boldsymbol{c}}(QxF)$ is $Qx\ UF_{\boldsymbol{c}}(F(x))$ where $Q \in \{\forall, \exists\}$.

It is clear that $UF_{\boldsymbol{c}}(F)$ is equivalent to F under classical logic. However, in general, $UF_{\boldsymbol{c}}(F)$ and F do not have the same stable models.

Example 17 *Consider when* F *is* $p(f) \wedge p(1) \wedge p(2)$. $UF_f(F)$ *is* $\exists x(p(x) \wedge f = x) \wedge p(1) \wedge p(2)$. *For an interpretation* I *such that* $f^I = 1$ *and the universe* $|I|$ *is* $\{1, 2\}$, I *is a stable model of* $UF_f(F)$ *but not of* F, *which we can easily see observing the reducts with respect to* I.
$F^{\underline{I}}$ *is* $p(f) \wedge p(1) \wedge p(2)$, *while*

$UF_f(F)^{\underline{I}}$ is equivalent to $f = 1 \wedge p(1) \wedge p(2)$.
Then, we can see that the interpretation J such that $f^J = 2$ and $|J| = |I| = \{1, 2\}$ satisfies $J <^{(f,g)} I$. Now, $J \models F^{\underline{I}}$ but $J \not\models UF_{(f,g)}(F)^{\underline{I}}$.

The following corollary shows that this method does preserve the stable models of formulas that are tight and in Clark Normal Form. And from this, combined with Theorem 15, we see another class of formulas for which we can eliminate intensional functions in favor of intensional predicates.

Corollary 5 *Let F be a formula in Clark Normal Form that is tight on $\boldsymbol{c}$. $SM[F; \boldsymbol{c}] \leftrightarrow SM[UF_{\boldsymbol{c}}(F); \boldsymbol{c}]$.*

7.5 Attempts at Generalizing Unfolding

While the syntactic restrictions in the previous sections–$\boldsymbol{f}$-plain, $\boldsymbol{f}$-plain-syn, and tight formulas in Clark Normal Form–are suitable for expressing many domains, it would be ideal to have a single general result that reveals how intensional functions may be eliminated in terms of intensional predicates. However, all attempts to convert non-$\boldsymbol{c}$-plain formulas into $\boldsymbol{c}$-plain formulas while preserving the stable models have proven fruitless.

In del Cerro *et al.* (2013), an attempt was made at establishing a Gentzen-style system for the functional stable model semantics. Theorem 2 in del Cerro *et al.* (2013) claimed that formulas that could be shown to be equivalent through the Gentzen-style system were strongly equivalent to each other. However, we were able to show that the formula $p(f)$ and $\exists x(p(x) \wedge f = x)$ could be shown to be equivalent in the Gentzen-style system. Example 17 in the previous section, demonstrates that these formulas are not strongly equivalent. The authors revised the Gentzen-style system in Cabalar *et al.* (2014) so that it no longer had this defect. Unfortunately, the new system no longer hinted at a way to convert non-$\boldsymbol{c}$-plain formulas into $\boldsymbol{c}$-plain formulas while preserving the stable models.

In fact, we are able to make a strong claim about the inability to find such a transformation. A *modular* translation is one that can be performed on conjunctive subformulas independently of each other. This is important for supporting elaboration-tolerance; if an elaboration E is introduced to a formula F, a modular translation T is one such that $T(F \wedge E)$ is strongly equivalent to $T(F) \wedge T(E)$.

Theorem 17 *There is no modular, signature-preserving translation that turns any sentence F into a $\mathbf{c}$-plain sentence F' such that $SM[F; \mathbf{c}]$ is equivalent to $SM[F'; \mathbf{c}]$ for any list $\mathbf{c}$ of constants.*

7.6 Proofs

7.6.1 Proof of Theorem 14

Lemma 16 *Assume that K and X are multi-valued interpretations of σ and Y is a propositional interpretation of σ^{prop} which is a subset of X such that*

$$K(c) = X(c) \textit{ iff } c = X(c) \in Y.$$

We have that $K \models F^X$ (when we view F as a multi-valued formula of σ) iff $Y \models F^X$ (when we view F as a propositional formula of σ^{prop}).

Proof. By induction on F. We show only the case of atoms. The other cases are straightforward.

Let F be an atom $c = v$. If $X \models c = v$, then F^X is F. The claim follows from the assumption since $K \models c = v$ iff $Y \models c = v$. If $X \not\models c = v$, then F^X is $\bot$, which neither K nor Y satisfies. ∎

Theorem 14 *Let F be a multi-valued propositional formula of signature σ, which can be also viewed as a propositional formula of signature σ^{prop}.*

(a) If an interpretation I of σ is a multi-valued stable model of F, then I can be viewed as an interpretation of σ^{prop} that is a propositional stable model of $F \wedge UEC_\sigma$ in the sense of Ferraris (2005).

(b) If an interpretation I of σ^{prop} is a propositional stable model of $F \wedge UEC_\sigma$ in the sense of Ferraris (2005), then I can be viewed as an interpretation of σ that is a multi-valued stable model of F.

Proof. (a) Assume X of signature σ is a stable model of F. This means $X \models F$ and no multi-valued interpretation K different from X satisfies F^X. Now since X is a multi-valued intepretation, $X \models UEC_\sigma$. Then clearly $X \models F$ when viewed as a propositional formula of signature σ^{prop}.

So, we wish to show that there is no interpretation Y of signature σ^{prop} such that $Y \subset X$ when X is viewed as a set of propositional atoms and $Y \models (F \wedge UEC_\sigma)^X$ when viewed as a propositional formula of signature σ^{prop}. To do so, we prove the contrapositive. We will show that if there is an interpretation Y of signature σ^{prop} such that $Y \subset X$ when X is viewed as a set of propositional atoms and $Y \models (F \wedge UEC_\sigma)^X$ when viewed as a propositional formula of signature σ^{prop}, then there is an interpretation K different from X that satisfies F^X when viewed as a multi-valued formula of signature σ.

Given such an interpretation Y, we create K as follows. For each $c \in \sigma$,

$$K(c) = \begin{cases} v & \text{if } c = v \in Y \\ m_c(v) & : \text{ if } c = v \in X \text{ and } c = v \notin Y \end{cases}$$

where m_c is any mapping from $m : Dom(c) \rightarrow Dom(c)$ such that $m(x) \neq x$. Note that this requires that every $Dom(c)$ have at least two elements. Note that since $Y \subset X$, there is at least one $c \in \sigma$ and $v \in Dom(c)$ such that $c = v \in X$ but $c = v \notin Y$. For this c, $K(c) = m(X(c)) \neq X(c)$ so K and X are different.

In addition, we have that $K(c) = X(c)$ iff $c = X(c) \in Y$. Now, since $Y \models (F \wedge UEC_\sigma)^X$, it follows that $Y \models F^X$. Thus, from Lemma 16 it follows that since $Y \models F^X$, then $K \models F^X$.

(b) Assume X of signature σ^{prop} is a stable model of $F \wedge UEC_\sigma$. This means that $X \models F \wedge UEC_\sigma$ and no interpretation Y such that $Y \subset X$ satisfies $(F \wedge UEC_\sigma)^X$. Since $X \models UEC_\sigma$, then X can be viewed as a multi-valued interpretation. Then clearly, $X \models F$.

Now, we wish to show that there is no interpretation K of signature σ that is different from X satisfying F^X. To do so, we prove the contrapositive. We will show that if there is an interpretation K of signature σ different from X and $K \models F^X$, then there is an interpretation Y such that $Y \subset X$ that satisfies $(F \wedge UEC_\sigma)^X$. Now since we already have seen that $X \models UEC_\sigma$, then $(UEC_\sigma)^X$ is equivalent to $\top$ so we need only show that there is an interpretation Y such that $Y \subset X$ that satisfies F^X.

Given such an interpretation K, we create Y as follows. Let us view K as a set of propositional atoms. We will take $Y = X \cap K$. Clearly $Y \subset X$. In addition, we have that $K(c) = X(c)$ iff $c = X(c) \in Y$. Thus, from Lemma 16 it follows that since $Y \models F^X$, then $K \models F^X$. ∎

7.6.2 Proof of Theorem 15

Theorem 15 *For any f-plain formula F,*

$$\forall \boldsymbol{x} y(p(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y) \tag{7.7}$$

and $\exists xy(x \neq y)$ entail

$$SM[F; f\boldsymbol{c}] \leftrightarrow SM[F^f_p; p\boldsymbol{c}].$$

Proof.

This is precisely the statement of Lemma 13. ∎

7.6.3 Proof of Corollary 4

Corollary 4 *Let F be an f-plain sentence. (a) An interpretation I of the signature of F that satisfies $\exists xy(x \neq y)$ is a model of SM$[F; f\boldsymbol{c}]$ iff I^f_p is a model of SM$[F^f_p; p\boldsymbol{c}]$. (b) An interpretation J of the signature of F^f_p that satisfies $\exists xy(x \neq y)$ is a model of SM$[F^f_p \wedge UEC_p; p\boldsymbol{c}]$ iff $J = I^f_p$ for some model I of SM$[F; f\boldsymbol{c}]$.*

Proof.

This is precisely the same statement as Lemma 14. ∎

7.6.4 Proof of Theorem 16

Lemma 17 *Given two lists of predicate and function constants $\boldsymbol{c}$ and $\boldsymbol{d}$ whose elements are in one-to-one correspondence, two lists of predicate constants $\boldsymbol{p}$ and $\boldsymbol{q}$ and two lists of function constants $\boldsymbol{f}$ and $\boldsymbol{g}$ all of the same length, a formula F of signature $\sigma \supseteq \boldsymbol{c} \cup \boldsymbol{f}$ that is $\boldsymbol{f}$-plain, and an interpretation I over a signature $\sigma' \supseteq \sigma \cup \boldsymbol{d} \cup \boldsymbol{p} \cup \boldsymbol{q} \cup \boldsymbol{g}$ that satisfies*

$$\forall \boldsymbol{x} y(p(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y) \tag{7.8}$$

for each corresponding p and f in $\boldsymbol{p}$ and $\boldsymbol{f}$ respectively,

$$\forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y) \tag{7.9}$$

for each corresponding q, f, and g in $\boldsymbol{q}$, $\boldsymbol{f}$, and $\boldsymbol{g}$ respectively, and

$$\forall \boldsymbol{x}\boldsymbol{y}(f_1(\boldsymbol{x}) = f_2(\boldsymbol{y}) \rightarrow g_1(\boldsymbol{x}) = g_2(\boldsymbol{y})) \tag{7.10}$$

for each corresponding f_1,f_2 and g_1,g_2 in $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively, if $I \models (F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{q}\boldsymbol{d})$, then $I \models F^*(\boldsymbol{g}\boldsymbol{d})$.

Proof. By induction on F.

Case 1: F is an atomic formula not containing any f from $\boldsymbol{f}$.

$F^{\boldsymbol{f}}_{\boldsymbol{p}}$ is exactly F thus $F^*(\boldsymbol{gd})$ is exactly $(F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{qd})$ so certainly the claim holds.

Case 2: F is $f(\boldsymbol{t}) = c$ where $f \in \boldsymbol{f}$ and $\boldsymbol{t}$ and c contain no elements from $\boldsymbol{f}$.

$F^*(\boldsymbol{gd})$ is $f(\boldsymbol{t}) = c \wedge g(\boldsymbol{t}) = c$.

$F^{\boldsymbol{f}}_{\boldsymbol{p}}$ is $p(\boldsymbol{t}, c)$.

$(F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{qd})$ is $q(\boldsymbol{t}, c)$.

Since $I \models$ (7.9) for every corresponding q, f, and g in $\boldsymbol{q}$, $\boldsymbol{f}$, and $\boldsymbol{g}$ respectively, it immediately follows that $I \models F^*(\boldsymbol{gd})$ iff $I \models (F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{qd})$.

Case 3: F is of the form $f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ where f_1 and f_2 are intensional and neither $\boldsymbol{t}_1$ nor $\boldsymbol{t}_2$ contains intensional constants.

$F^*(\boldsymbol{gd})$ is $f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2) \wedge g_1(\boldsymbol{t}_1) = g_2(\boldsymbol{t}_2)$.

$F^{\boldsymbol{f}}_{\boldsymbol{p}}$ is $\forall y(p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y))$.

$(F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{gd})$ is $\forall y((p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y)) \wedge (q_1(\boldsymbol{t}_1, y) \leftrightarrow q_2(\boldsymbol{t}_2, y)))$.

which is further equivalent to $\forall y(p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y)) \wedge \forall y(q_1(\boldsymbol{t}_1, y) \leftrightarrow q_2(\boldsymbol{t}_2, y))$.

Since $I \models$ (7.8) for every corresponding p and f in $\boldsymbol{p}$ and $\boldsymbol{f}$ respectively, it is clear that $I \models f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ iff $I \models \forall y((p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y))$. We consider two cases.

- If $I \not\models f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$, then clearly, $I \not\models F^*(\boldsymbol{gd})$ and by the previous observation $I \not\models (F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{gd})$ so the claim holds.

- Otherwise, $I \models f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$. Now, since $I \models$ (7.10) for each corresponding f_1,f_2 and g_1,g_2, we have that $I \models g_1(\boldsymbol{t}_1) = g_2(\boldsymbol{t}_2)$ and so $I \models F^*(\boldsymbol{gd})$. If $f_1(\boldsymbol{t}_1) = g_1(\boldsymbol{t}_1)$ (and thus $f_2(\boldsymbol{t}_2) = g_2(\boldsymbol{t}_2)$), then since $I \models$ (7.9) for each corresponding q,

f, and g in $\boldsymbol{q}$, $\boldsymbol{f}$, and $\boldsymbol{g}$ respectively, $I \models q_1(t_1, \xi)$ for $f_1(\boldsymbol{t}_1)^I = \xi$ and similarly $I \models q_2(t_2, \xi)$. If on the other hand $f_1(\boldsymbol{t}_1) \neq g_1(\boldsymbol{t}_1)$ (and thus $f_2(\boldsymbol{t}_2) \neq g_2(\boldsymbol{t}_2)$), then $I \not\models q_1(t_1, \xi)$ for any ξ. In either case, we conclude $I \models \forall y(q_1(\boldsymbol{t}_1, y) \leftrightarrow q_2(\boldsymbol{t}_2, y))$.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.
By I.H. on G and H.

Case 5: F is $G \rightarrow H$.
By I.H. on G and H.

Case 6: F is $Q\boldsymbol{x}G(\boldsymbol{x})$ where $Q \in \{\forall, \exists\}$.
By I.H. on G. ∎

Lemma 18 *Given two lists of predicate and function constants* $\boldsymbol{c}$ *and* $\boldsymbol{d}$ *whose elements are in one-to-one correspondence, two lists of predicate constants* $\boldsymbol{p}$ *and* $\boldsymbol{q}$ *and two lists of function constants* $\boldsymbol{f}$ *and* $\boldsymbol{g}$ *all of the same length, a formula* F *of signature* $\sigma \supseteq \boldsymbol{c} \cup \boldsymbol{f}$ *that is* $\boldsymbol{f}$*-plain, and an interpretation* I *over a signature* $\sigma' \supseteq \sigma \cup \boldsymbol{d} \cup \boldsymbol{p} \cup \boldsymbol{q} \cup \boldsymbol{g}$ *that satisfies*

$$\forall \boldsymbol{x} y(p(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y) \tag{7.11}$$

for each corresponding p *and* f *in* $\boldsymbol{p}$ *and* $\boldsymbol{f}$ *respectively,*

$$\forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y) \tag{7.12}$$

for each corresponding q, f, *and* g *in* $\boldsymbol{q}$, $\boldsymbol{f}$, *and* $\boldsymbol{g}$ *respectively, if* $I \models F^*(\boldsymbol{gd})$, *then* $I \models (F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{qd})$.

Proof. By induction on F.

Case 1: F is an atomic formula not containing any f from $\boldsymbol{f}$.

$F^{\boldsymbol{f}}_{\boldsymbol{p}}$ is exactly F thus $F^*(\boldsymbol{gd})$ is exactly $(F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{qd})$ so certainly the claim holds.

Case 2: F is $f(\boldsymbol{t}) = c$ where $f \in \boldsymbol{f}$ and $\boldsymbol{t}$ and c contain no elements from $\boldsymbol{f}$.

$F^*(\boldsymbol{gd})$ is $f(\boldsymbol{t}) = c \wedge g(\boldsymbol{t}) = c$.

$F^{\boldsymbol{f}}_{\boldsymbol{p}}$ is $p(\boldsymbol{t}, c)$.

$(F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{qd})$ is $q(\boldsymbol{t}, c)$.

Since $I \models$ (7.12) for every corresponding q, f, and g in $\boldsymbol{q}$, $\boldsymbol{f}$, and $\boldsymbol{g}$ respectively, it immediately follows that $I \models F^*(\boldsymbol{gd})$ iff $I \models (F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{qd})$.

Case 3: F is of the form $f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ where f_1 and f_2 are intensional and neither $\boldsymbol{t}_1$ nor $\boldsymbol{t}_2$ contains intensional constants.

$F^*(\boldsymbol{gd})$ is $f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2) \wedge g_1(\boldsymbol{t}_1) = g_2(\boldsymbol{t}_2)$.

$F^{\boldsymbol{f}}_{\boldsymbol{p}}$ is $\forall y(p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y))$.

$(F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{gd})$ is $\forall y((p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y)) \wedge (q_1(\boldsymbol{t}_1, y) \leftrightarrow q_2(\boldsymbol{t}_2, y)))$.

which is further equivalent to $\forall y(p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y)) \wedge \forall y(q_1(\boldsymbol{t}_1, y) \leftrightarrow q_2(\boldsymbol{t}_2, y))$.

Since $I \models$ (7.11) for every corresponding p and f in $\boldsymbol{p}$ and $\boldsymbol{f}$ respectively, it is clear that $I \models f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ iff $I \models \forall y((p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y))$. We consider two cases.

- If $I \not\models f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$, then clearly, $I \not\models F^*(\boldsymbol{gd})$ and by the previous observation $I \not\models (F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\boldsymbol{gd})$ so the claim holds.

- Otherwise, $I \models f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$. Now, if $I \models f_1(\boldsymbol{t}_1) = g_1(\boldsymbol{t}_1)$, then since we assume $I \models F^*(\boldsymbol{gd})$, we have $I \models f_2(\boldsymbol{t}_2) = g_2(\boldsymbol{t}_2)$. Thus, since $I \models$ (7.12), it follows that $I \models q_1(\boldsymbol{t}_1, \xi) \wedge q_2(\boldsymbol{t}_2, \xi))$ for $f_1(\boldsymbol{t}_1)^I = \xi$ and for all other $\xi' \neq \xi$,

$I \models \neg q_1(\boldsymbol{t}_1, \xi) \wedge \neg q_2(\boldsymbol{t}_2, \xi))$. If on the other hand $I \not\models f_1(\boldsymbol{t}_1) = g_1(\boldsymbol{t}_1)$, then since we assume $I \models F^*(\boldsymbol{gd})$, we have $I \not\models f_2(\boldsymbol{t}_2) = g_2(\boldsymbol{t}_2)$. Thus, since $I \models$ (7.12), it follows that $I \models \neg q_1(\boldsymbol{t}_1, \xi) \wedge q_2(\boldsymbol{t}_2, \xi))$ for all ξ. In either case we conclude $I \models \forall y(q_1(\boldsymbol{t}_1, y) \leftrightarrow q_2(\boldsymbol{t}_2, y))$ so the claim holds.

Case 4: F is $G \odot H$ where $\odot \in \{\wedge, \vee\}$.

By I.H. on G and H.

Case 5: F is $G \rightarrow H$.

By I.H. on G and H.

Case 6: F is $Q\boldsymbol{x}G(\boldsymbol{x})$ where $Q \in \{\forall, \exists\}$.

By I.H. on G. ∎

Lemma 19 *Given two lists of predicate and function constants $\boldsymbol{c}$ and $\boldsymbol{d}$ whose elements are in one-to-one correspondence, two lists of predicate constants $\boldsymbol{p}$ and $\boldsymbol{q}$ and two lists of function constants $\boldsymbol{f}$ and $\boldsymbol{g}$ all of the same length, a formula F of signature $\sigma \supseteq \boldsymbol{c} \cup \boldsymbol{f}$ that is $\boldsymbol{f}$-plain, and an interpretation I over a signature $\sigma' \supseteq \sigma \cup \boldsymbol{d} \cup \boldsymbol{p} \cup \boldsymbol{q} \cup \boldsymbol{g}$ that satisfies*

$$\forall \boldsymbol{x} y(p(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y) \tag{7.13}$$

for each corresponding p and f in $\boldsymbol{p}$ and $\boldsymbol{f}$ respectively and

$$\forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y) \tag{7.14}$$

for each corresponding q, f, and g in $\boldsymbol{q}$, $\boldsymbol{f}$, and $\boldsymbol{g}$ respectively, $I \models \boldsymbol{gd} < \boldsymbol{fc}$ iff $I \models \boldsymbol{qd} < \boldsymbol{pc}$.

Proof. ($\Rightarrow$) Assume $I \models \boldsymbol{gd} < \boldsymbol{fc}$. By definition, it follows that $I \models (\boldsymbol{gd})^{pred} \leq (\boldsymbol{fc})^{pred}$ and since $\boldsymbol{g}$ and $\boldsymbol{f}$ contain no predicates, we have $I \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$. The following arguments are made for corresponding tuples of p, q, f, and g from $\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{f}$, and $\boldsymbol{g}$ respectively. Since we assume $I \models$ (7.14), it follows that $I \models \forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \rightarrow f(\boldsymbol{x}) = y)$. Then from the assumption that $I \models$ (7.13), it follows that $I \models \forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \rightarrow p(\boldsymbol{x}, y))$ or simply that $I \models q \leq p$, from which it follows that $I \models (\boldsymbol{qd})^{pred} \leq (\boldsymbol{pc})^{pred}$.

Now since $I \models \boldsymbol{gd} < \boldsymbol{fc}$, it follows that $I \models \neg(\boldsymbol{gd} = \boldsymbol{fc})$. We consider two cases

- If $I \models \neg(d = c)$ for some corresponding d and c in $\boldsymbol{d}$ and $\boldsymbol{c}$ respectively, then we have $I \models \neg(\boldsymbol{d} = \boldsymbol{c})$ and further, $I \models \neg(\boldsymbol{qd} = \boldsymbol{pc})$.

- Otherwise, it must be that $I \models \neg(g = f)$ for some corresponding g and f in $\boldsymbol{g}$ and $\boldsymbol{f}$ respectively. That is, for some $\boldsymbol{\xi}$ and ξ, $I \not\models f(\boldsymbol{\xi}) = \xi \leftrightarrow g(\boldsymbol{\xi}) = \xi$. For a given $\boldsymbol{\xi}$, I maps $f(\boldsymbol{\xi})$ to exactly one ξ and similarly for $g(\boldsymbol{\xi})$ and so it follows that $I \not\models f(\boldsymbol{\xi}) = \xi \wedge g(\boldsymbol{\xi}) = \xi$ for every ξ. Since $I \models$ (7.14), $I \not\models q(\boldsymbol{\xi}, \xi)$ for every ξ. However, since $I \models f(\boldsymbol{\xi}) = \xi$ for some ξ, from $I \models$ (7.13), we know $I \models p(\boldsymbol{\xi}, \xi)$ for some ξ. Thus, $I \models \neg(q = p)$ and further $I \models \neg(\boldsymbol{qd} = \boldsymbol{pc})$.

From either case, we then conclude that $I \models q\boldsymbol{d} < p\boldsymbol{c}$.

($\Leftarrow$) Assume $I \models \boldsymbol{qd} < \boldsymbol{pc}$. By definition, it follows that $I \models (\boldsymbol{qd})^{pred} \leq (\boldsymbol{pc})^{pred}$ and further, we have $I \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$. Then, since $\boldsymbol{f}$ and $\boldsymbol{g}$ do not contain predicates, we have $I \models (\boldsymbol{gd})^{pred} \leq (\boldsymbol{fc})^{pred}$.

Now since $I \models \boldsymbol{qd} < \boldsymbol{pc}$, it follows that $I \models \neg(\boldsymbol{qd} = \boldsymbol{pc})$. We consider two cases

- If $I \models \neg(d = c)$ for some corresponding d and c in $\boldsymbol{d}$ and $\boldsymbol{c}$ respectively, then we have $I \models \neg(\boldsymbol{d} = \boldsymbol{c})$ and further, $I \models \neg(\boldsymbol{gd} = \boldsymbol{fc})$

- Otherwise, it must be that $I \models \neg(q = p)$ for some corresponding q and p from $\boldsymbol{q}$ and $\boldsymbol{p}$ respectively. That is, for some $\boldsymbol{\xi}$ and ξ, $I \not\models q(\boldsymbol{\xi}, \xi) \leftrightarrow p(\boldsymbol{\xi}, \xi)$. Since $I \models$ (7.13), there is exactly one $\boldsymbol{\xi}$ and ξ such that $I \models p(\boldsymbol{\xi}, \xi)$, which further means that $I \models f(\boldsymbol{\xi}) = \xi$. Thus since $I \models q < p$, it must be that $I \not\models q(\boldsymbol{\xi}, \xi)$, and since $I \models$ (7.14), it follows that $I \not\models g(\boldsymbol{\xi}) = \xi$. Thus, $I \models \neg(g = f)$ and further $I \models \neg(\boldsymbol{gd} = \boldsymbol{fc})$.

From either case, we then conclude that $I \models \boldsymbol{gd} < \boldsymbol{fc}$.

Theorem 16 *For any $\boldsymbol{f}$-plain-syn formula F, the set of formulas $\forall \boldsymbol{x} y(p(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y)$ for each $f \in \boldsymbol{f}$ and the corresponding p, and $\exists xy(x \neq y)$ entail*

$$SM[F; \boldsymbol{fq}] \leftrightarrow SM[F^{\boldsymbol{f}}_{\boldsymbol{p}}; \boldsymbol{pq}].$$

Proof. We will show that the conjunction over all pairs of corresponding f and p from $\boldsymbol{f}$ and $\boldsymbol{p}$ of

$$\forall \boldsymbol{x} y(p(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y) \tag{7.15}$$

entails

$$\mathrm{SM}[F; \boldsymbol{fc}] \leftrightarrow \mathrm{SM}[F^{\boldsymbol{f}}_{\boldsymbol{p}}; \boldsymbol{pc}].$$

Claim 1:For any interpretation $\mathcal{I} = \langle I, X \rangle$ of signature $\sigma \supseteq \{\boldsymbol{f}, \boldsymbol{p}, \boldsymbol{c}\}$ satisfying (7.15), $\mathcal{I} \models F$ iff $\mathcal{I} \models F^{\boldsymbol{f}}_{\boldsymbol{p}}$. We show this by showing that every atomic formula A in F is classically equivalent to the corresponding formula $A^{\boldsymbol{f}}_{\boldsymbol{p}}$ in $F^{\boldsymbol{f}}_{\boldsymbol{p}}$:

- A contains no intensional function constants. Then A is identical to $A^{\boldsymbol{f}}_{\boldsymbol{p}}$.

- A is of the form $f(\boldsymbol{t}) = c$ where f is an intensional function constant and neither t nor c contains intensional function constants. The corresponding formula $A^{\boldsymbol{f}}_{\boldsymbol{p}}$ is $p(\boldsymbol{t}, c)$ and under (7.15) it is clear that these are equivalent.

- A is of the form $f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ where f_1 and f_2 are intensional function constants and neither t_1 nor t_2 contains intensional function constants. The corresponding formula $A^{\boldsymbol{f}}_{\boldsymbol{p}}$ is $p_1(\boldsymbol{t}_1, y) \leftrightarrow p_2(\boldsymbol{t}_2, y)$ and under (7.15), this is equivalent to $f_1(\boldsymbol{t}_1) = y \leftrightarrow f_2(\boldsymbol{t}_2) = y$ which is equivalent to A.

Claim 2:

$$\mathcal{I} \models \neg\exists\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}} < \boldsymbol{f}\boldsymbol{c}) \wedge F^*(\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}}))$$

iff

$$\mathcal{I} \models \neg\exists\widehat{\boldsymbol{p}}\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{p}}\widehat{\boldsymbol{c}} < \boldsymbol{p}\boldsymbol{c}) \wedge (F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\widehat{\boldsymbol{p}}\widehat{\boldsymbol{c}}))$$

or equivalently,

$$\mathcal{I} \models \exists\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}} < \boldsymbol{f}\boldsymbol{c}) \wedge F^*(\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}}))$$

iff

$$\mathcal{I} \models \exists\widehat{\boldsymbol{p}}\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{p}}\widehat{\boldsymbol{c}} < \boldsymbol{p}\boldsymbol{c}) \wedge (F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\widehat{\boldsymbol{p}}\widehat{\boldsymbol{c}})).$$

($\Rightarrow$) Assume $\mathcal{I} \models \exists\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}} < \boldsymbol{f}\boldsymbol{c}) \wedge F^*(\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}}))$. We wish to show that $\mathcal{I} \models \exists\widehat{\boldsymbol{p}}\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{p}}\widehat{\boldsymbol{c}} < \boldsymbol{p}\boldsymbol{c}) \wedge (F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\widehat{\boldsymbol{p}}\widehat{\boldsymbol{c}}))$.

That is, take any list of functions $\boldsymbol{g}$ of the same arities as the corresponding functions in $\boldsymbol{f}$ and any list of predicates $\boldsymbol{d}$ of the same length $\boldsymbol{c}$. Now let $\mathcal{I}' = \langle I \cup J^{\boldsymbol{f}}_{\boldsymbol{g}}, X \cup Y^{\boldsymbol{c}}_{\boldsymbol{d}}\rangle$ be from an extended signature $\sigma' = \sigma \cup \{\boldsymbol{g}, \boldsymbol{d}\}$ where J is an interpretation of functions from the signature σ and I and J agree on all symbols not occurring in $\boldsymbol{f}$. $J^{\boldsymbol{f}}_{\boldsymbol{g}}$ denotes the interpretation from $\sigma^{\boldsymbol{f}}_{\boldsymbol{g}}$ (the signature obtained from σ by replacing $f \in \boldsymbol{f}$ with the corresponding $g \in \boldsymbol{g}$ and all elements of $\boldsymbol{c}$ with all elements of $\boldsymbol{d}$) obtained from the interpretation J by replacing $f \in \boldsymbol{f}$ with the corresponding $g \in \boldsymbol{g}$. Similarly, $Y^{\boldsymbol{c}}_{\boldsymbol{d}}$ is the interpretation from $\sigma^{\boldsymbol{c}}_{\boldsymbol{d}}$ obtained from the interpretation Y by replacing $c \in \boldsymbol{c}$ by the corresponding $d \in \boldsymbol{d}$. We assume

$$\mathcal{I}' \models (\boldsymbol{gd} < \boldsymbol{fc} \wedge F^*(\boldsymbol{gd}))$$

and wish to show that there is a list of predicates $\boldsymbol{q}$ of the same arities as the corresponding predicates in $\boldsymbol{p}$ such that

$$\mathcal{I}' \models (\boldsymbol{qd} < \boldsymbol{pc} \wedge (F_{\boldsymbol{p}}^{\boldsymbol{f}})^*(\boldsymbol{qd})).$$

We define each new predicate q in terms of the corresponding f and g as follows:

$$q^{\mathcal{I}'}(\vec{\xi}, \xi) = \begin{cases} 1 & \text{if } \mathcal{I}' \models f(\vec{\xi}) = \xi \wedge g(\vec{\xi}) = \xi \\ 0 & \text{otherwise} \end{cases}$$

We assume $\mathcal{I}' \models$ (7.15) for each corresponding f and p from $\boldsymbol{f}$ and $\boldsymbol{p}$ respectively. It is clear from the definition of q that $\mathcal{I}' \models \forall \boldsymbol{x} y(q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y)$ for each corresponding q, f, and g from $\boldsymbol{q}$, $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively. Thus, by Lemma 19, $\mathcal{I}' \models \boldsymbol{qd} < \boldsymbol{pc}$. From Lemma 18, we conclude $\mathcal{I}' \models F_{\boldsymbol{p}}^{\boldsymbol{f}})^*(\widehat{\boldsymbol{pc}})$.

($\Leftarrow$) Assume $\mathcal{I} \models \exists \widehat{\boldsymbol{pc}}((\widehat{\boldsymbol{pc}} < \boldsymbol{pc}) \wedge (F_{\boldsymbol{p}}^{\boldsymbol{f}})^*(\widehat{\boldsymbol{pc}}))$. We wish to show that $\mathcal{I} \models \exists \widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}}((\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}} < \boldsymbol{fc}) \wedge F^*(\widehat{\boldsymbol{f}}\widehat{\boldsymbol{c}}))$.

That is, take any list predicates $\boldsymbol{q}$ of the same arities as the corresponding predicates in $\boldsymbol{p}$ and any list of predicates $\boldsymbol{d}$ the same length as $\boldsymbol{c}$ such that

$$\mathcal{I}' \models (\boldsymbol{qd} < \boldsymbol{pc} \wedge (F_{\boldsymbol{p}}^{\boldsymbol{f}})^*(\boldsymbol{qd})).$$

We wish to show that there is a function g of the same arity as f such that

$$\mathcal{I}' \models (\boldsymbol{gd} < \boldsymbol{fc} \wedge F^*(\boldsymbol{gd}))$$

where $\mathcal{I}' = \langle I \cup J_{\boldsymbol{g}}^{\boldsymbol{f}}, X \cup Y_{\boldsymbol{d}}^{\boldsymbol{c}} \rangle$ is defined as before. Take any mapping $m : |\mathcal{I}'| \to |\mathcal{I}'|$ such that $\forall x(m(x) \neq x)$. We define each new function g in terms of the corresponding f, p, and q as follows:

$$g^{\mathcal{I}'}(\vec{\xi}) = \begin{cases} f^{\mathcal{I}'}(\vec{\xi}) & \text{if } \mathcal{I}' \models \exists y(p(\vec{\xi}, y) \wedge q(\vec{\xi}, y)) \\ m(f^{\mathcal{I}'}(\vec{\xi})) & \text{otherwise} \end{cases}$$

Note that the assumption that there are at least two elements in the universe is essential to this definition. We assume $\mathcal{I}' \models$ (7.15) for each corresponding f and p from $\boldsymbol{f}$ and $\boldsymbol{p}$ respectively. It is clear from the definition of g that $\mathcal{I}' \models \forall \boldsymbol{xy}(f_1(\boldsymbol{x}) = f_2(\boldsymbol{y}) \to g_1(\boldsymbol{x}) = g_2(\boldsymbol{y})$ for each corresponding f and g from $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively. We now show that $\mathcal{I}' \models \forall \boldsymbol{x}y(q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y)$ for each corresponding q, f, and g from $\boldsymbol{q}$, $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively.

Since we assume $\mathcal{I}' \models$ (7.7), it follows that for any given $\boldsymbol{\xi}$, there is only one ξ such that $\mathcal{I}' \models p(\boldsymbol{\xi}, \xi)$. Then, since we assume $\mathcal{I}' \models q \leq p$, we know $\mathcal{I}' \not\models q(\boldsymbol{\xi}, \xi')$ for any $\xi' \neq \xi$. If $\mathcal{I}' \models q(\boldsymbol{\xi}, \xi)$, then $\mathcal{I}' \models g(\boldsymbol{\xi}) = \xi$. Otherwise, $\mathcal{I}' \models g(\boldsymbol{\xi}) = \xi'$ for some $\xi' \neq \xi$. Since this is true for any $\boldsymbol{\xi}$, it follows that $\mathcal{I}' \models \forall \boldsymbol{x}y(q(\boldsymbol{x}, y) \leftrightarrow f(\boldsymbol{x}) = y \wedge g(\boldsymbol{x}) = y)$.

Thus, by Lemma 19, $\mathcal{I}' \models \boldsymbol{qd} < \boldsymbol{pc}$. From Lemma 17, we conclude $\mathcal{I}' \models F^{\boldsymbol{f}}_{\boldsymbol{p}})^*(\widehat{\boldsymbol{pc}})$.

∎

7.6.5 *Proof of Corollary 5*

Corollary 5 *Let F be a formula in Clark Normal Form that is tight on $\boldsymbol{c}$. SM$[F; \boldsymbol{c}] \leftrightarrow$ SM$[UF_{\boldsymbol{c}}(F); \boldsymbol{c}]$.*

Theorem 8 from section 5.5 tells us that the stable models relative to $\boldsymbol{c}$ of a formula F in Clark Normal Form that is tight on $\boldsymbol{c}$ are in one-to-one correspondence with the classical models of the completion of F relative to $\boldsymbol{c}$, which we will denote G. Further, since unfolding is a classically equivalent transformation, the stable models of F are in one-to-one correspondence with classical models of $UF_{\boldsymbol{c}}(G)$.

To apply Theorem 8 to $UF_{\boldsymbol{c}}(F)$, we must first establish that $UF_{\boldsymbol{c}}(F)$ is tight on $\boldsymbol{c}$. This follows immediately from the fact that $UF_{\boldsymbol{c}}$ does not affect strictly positively occurring atomic formulas since F is in Clark Normal Form and so these are of the form $p(\boldsymbol{x})$ or $f(\boldsymbol{x})=y$ and so are already $\boldsymbol{c}$ plain. Further, for any implication $B \rightarrow H$ of F, any atomic formula A occuring strictly positively in B, then since $UF_{\boldsymbol{c}}$ does not affect the number of implications that have A in the antecedent. Thus, $UF_{\boldsymbol{c}}(F)$ is tight on $\boldsymbol{c}$.

So by Theorem 8, the stable models of $UF_{\boldsymbol{c}}(F)$ are in one-to-one correspondence with the classical models of the completion of $UF_{\boldsymbol{c}}(F)$ relative to $\boldsymbol{c}$. All that remains to shown is that $UF_{\boldsymbol{c}}(G)$ is the same formula as the completion of $UF_{\boldsymbol{c}}(F)$ relative to $\boldsymbol{c}$.

Consider any implication $B \rightarrow H$ in F. The corresponding equivalence in the completion of F is $B \leftrightarrow H$. Now, applying unfolding here yields $UF_{\boldsymbol{c}}(B) \leftrightarrow H$ since as noted before, H is already $\boldsymbol{c}$-plain. On the other hand, the corresponding implication in G is $UF_{\boldsymbol{c}}(B) \rightarrow H$ and further, the corresponding equivalence in the completion of G is $UF_{\boldsymbol{c}}(B) \leftrightarrow H$ and so we conclude that $UF_{\boldsymbol{c}}(G)$ is the same formula as the completion of $UF_{\boldsymbol{c}}(F)$ relative to $\boldsymbol{c}$. ∎

7.6.6 Proof of Theorem 17

For a formula F involving function f and g, we call it fg-indistinguishable if every occurrence of f or g in F has the form $(f = t) \wedge (g = t)$, where t is a term. For 2 interpretations I and J of F, define the relation $R(I, J)$ as $R(I, J)$ if

- $I(f) \neq I(g)$;
- $J(f) \neq J(g)$;
- For every symbol s other than f or g, $I(s) = J(s)$.

Lemma 20 *If F is fg-indistinguishable, then for any I and J satisfying $R(I, J)$, $F^I = F^J$.*

Proof. By induction on F.

- F is an atom a (or $\bot$ or $\top$), where a does not involve f or f'. Obvious.
- F is $(f = t) \wedge (f' = t)$ for some t. Clearly since $I(f) \neq I(f')$ and $J(f) \neq J(f')$, $F^I = F^J = 0$.
- F is $\neg G$, where G is f-indistinguishable. For any I and J satisfying $R(I, J)$, by I.H., $G^I = G^J$, so $F^I = F^J$.
- F is $G \odot H$, where G and H are both f-indistinguishable and $\odot \in \{\wedge, \vee, \rightarrow\}$. For any I and J satisfying $R(I, J)$, by I.H, $G^I = G^J$ and $H^I = H^J$, so $F^I = F^J$.

Theorem 17 *There is no modular, signature-preserving translation that turns any sentence F into a $\boldsymbol{c}$-plain sentence F' such that SM$[F; \boldsymbol{c}]$ is equivalent to SM$[F'; \boldsymbol{c}]$ for any list $\boldsymbol{c}$ of constants.*

Proof.

Assume there is such a translation T that can turn a sentence F into a $\boldsymbol{c}$-plain sentence F' such that SM$[F; \boldsymbol{c}]$ is equivalent to SM$[F'; \boldsymbol{c}]$ for any list $\boldsymbol{c}$ of constants.

We first consider a formula G that is $p(f) \wedge p(1)$ and a list $\boldsymbol{c}$ that is $\{f\}$. It is not hard to verify that for $\mathcal{I} = \{p(1), f = 1\}$, we have $\mathcal{I} \models \mathrm{SM}[G; f]$. Thus we will have that $\mathcal{I} \models \mathrm{SM}[G'; f]$.

In particular, for a new function constant g and interpretation $K = \{p(1), f = 1, g = 2\}$, we have that $K \not\models G'^{*}(g)$.

Further, we consider a formula H that is $p(2)$. Now since T is modular, we have that $(G \wedge H)' = G' \wedge H'$. Further, since H is already f-plain, $H' = H$ and so $(G \wedge H)' = G' \wedge H$. We will proceed by referring to $G \wedge H$ as F.

We first note that by definition of SM, we will be showing that $F \wedge \neg\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F^{*}(\widehat{\boldsymbol{c}}))$ is equivalent to $F' \wedge \neg\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F'^{*}(\widehat{\boldsymbol{c}}))$.

Now, when we let $\boldsymbol{c}$ be empty, this is simply to show that T is such that F is equivalent to F'. Thus, we only need to show that $\neg\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F^{*}(\widehat{\boldsymbol{c}}))$ is equivalent to $\neg\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F'^{*}(\widehat{\boldsymbol{c}}))$ or equivalently, the contrapositive $\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F^{*}(\widehat{\boldsymbol{c}}))$ is equivalent to $\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F'^{*}(\widehat{\boldsymbol{c}}))$.

That is, we will show that for some list of predicates and function constants $\boldsymbol{d}$ corresponding to $\boldsymbol{c}$,

Now consider when F is the formula

$$p(f) \wedge p(1) \wedge p(2)$$

and where $\boldsymbol{c} = \{f\}$. Consider a new function constant g. $F^{*}(g)$ is

$$p(f) \wedge p(g) \wedge p(1) \wedge p(2).$$

Consider two interpretations $I = \{p(1), p(2), f = 1, g = 2\}$ and $J = \{p(1), p(2), f = 1, g = 3\}$. Clearly $I \models F^{*}(g)$ and $J \not\models F^{*}(g)$. Note that $R(I, J)$. Observe that I serves to show that for $\mathcal{J} = \{p(1), p(2), f = 1\}$, we have that $\mathcal{J} \not\models \mathrm{SM}[F; g]$. Now, since $I \models F$ and we already observed that T is such that F is equivalent to F', $I \models F'$.

Now, recall that F is $G \wedge H$ and that $F' = G' \wedge H$. Then, we have that $F'^*(g)$ is $G'^*(g) \wedge H$. Now since $K \not\models G'^*(g)$ certainly, $I \not\models G'^*(g)$ (recall $K = \{p(1), f = 1, g = 2\}$) and further $I \not\models G'^*(g) \wedge H$ [2]. Now since F' is $\boldsymbol{c}$-plain, it must be that every occurrence of f is in a term of the form $f = t$ where t does not contain f. Then, in $F'^*(g)$, every occurrence of f and g has the form $(f = t) \wedge (g = t)$. Therefore, we can apply Lemma 20 and conclude that $F'^*(g)^I = F'^*(g)^J$ for any interpretations such that $R(I, J)$. Thus, for every J that is fg-indistinguishable, $J \not\models G'^*(g) \wedge H$. So we conclude that $\mathcal{J} \models \mathrm{SM}[F'; g]$ but $\mathcal{J} \not\models \mathrm{SM}[F; g]$. Thus, no such translation can exist.
∎

[2]Splitting Theorem is the basis of this claim.

Chapter 8

MANY-SORTED FSM

Under the functional stable model semantics described in Chapter 4, in any interpretation I, each function f is understood as $f : |I| \times \cdots \times |I| \rightarrow |I|$. When describing real-world domain, this is unnatural; while $velocity(car)$ is certainly a value of interest, $velocity(17)$ is not. To allow functions to map to and from sets other than the universe, in this chapter we present the many-sorted functional stable model semantics.

8.1 Extending FSM to Many-sorted FSM

To extend FSM to many-sorted logic, we use the same definition of a signature as in many-sorted logic; a signature σ is comprised of a list of function and predicate constants and a list of sorts. To each function and predicate of arity n, we assign argument sorts $s_1, \ldots, s_n$ and to function constants of arity n, we assign a value sort s_{n+1}. We assume that we have an infinite number of variables for each sort. Atomic formulas are built similar to standard logic with the restriction that in $f(t_1, \ldots, t_n)$ ($p(t_1, \ldots, t_n)$), the sort of t_i must be a subsort of the ith argument of f (p). In addition $t_1 = t_2$ is an atomic formula if the sorts and t_1 and t_2 have a common supersort.

A many-sorted interpretation I has a non-empty universe $|I|^s$ for each sort s. When s_1 is a subsort of s_2, then an interpretation must satisfy $|I|^{s_1} \subseteq |I|^{s_2}$. The notion of satisfaction is similar to classical logic with the restriction that an interpretation map a term to an element in its associated sort.

For predicate symbols (constants or variables) u and c that have the same assigned argument and value sorts, we define $u \le c$ as $\forall \boldsymbol{x}(u(\boldsymbol{x}) \rightarrow c(\boldsymbol{x}))$ where each $x \in \boldsymbol{x}$ is of the appropriate sort. We define $u = c$ as $\forall \boldsymbol{x}(u(\boldsymbol{x}) \leftrightarrow c(\boldsymbol{x}))$ where each $x \in \boldsymbol{x}$ is of the appropriate sort if u and c are predicate symbols, and $\forall \boldsymbol{x}(u(\boldsymbol{x}) = c(\boldsymbol{x}))$ where each $x \in \boldsymbol{x}$ is of the appropriate sort if they are function symbols.

Let $\boldsymbol{c}$ be a list of distinct predicate and function constants and let $\widehat{\boldsymbol{c}}$ be a list of distinct predicate and function variables corresponding to $\boldsymbol{c}$ such that each $\widehat{c} \in \widehat{\boldsymbol{c}}$ and corresponding $c \in \boldsymbol{c}$ have the same assigned argument and value sorts.

By $\boldsymbol{c}^{pred}$ we mean the list of the predicate constants in $\boldsymbol{c}$, and by $\widehat{\boldsymbol{c}}^{pred}$ the list of the corresponding predicate variables in $\widehat{\boldsymbol{c}}$. We define $\widehat{\boldsymbol{c}} < \boldsymbol{c}$ as

$$(\widehat{\boldsymbol{c}}^{pred} \le \boldsymbol{c}^{pred}) \wedge \neg(\widehat{\boldsymbol{c}} = \boldsymbol{c})$$

and SM$[F;\ \boldsymbol{c}]$ as

$$F \wedge \neg \exists \widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F^*(\widehat{\boldsymbol{c}})),$$

where $F^*(\widehat{\boldsymbol{c}})$ is defined as follows.

- When F is an atomic formula, F^* is $F' \wedge F$, where F' is obtained from F by replacing all intensional (function and predicate) constants in it with the corresponding (function and predicate) variables;
- $(F \wedge G)^* = F^* \wedge G^*$; $\quad (F \vee G)^* = F^* \vee G^*$;
- $(F \rightarrow G)^* = (F^* \rightarrow G^*) \wedge (F \rightarrow G)$;
- $(\forall x F)^* = \forall x F^*$; $\quad (\exists x F)^* = \exists x F^*$.

When F is a many-sorted sentence, the many-sorted interpretations that are models of $\mathrm{SM}[F;\ \boldsymbol{c}]$ are called the $\boldsymbol{c}$-*stable* models of F. They are the models of F that are "stable" on $\boldsymbol{c}$.

Note that the second-order characterization of the many-sorted functional stable model semantics is only different from the definition in Chapter 4 in that the formula F and interpretation I are many-sorted and every variable $\widehat{c}$ in the list of variables $\widehat{\boldsymbol{c}}$ must have the same assigned sorts as the corresponding constant c in $\boldsymbol{c}$.

8.2 Reduct characterization of Many-sorted FSM

As was the case for the nonsorted functional stable model semantics, we can also define a characterization of the many-sorted functional stable model semantics in terms of grounding and reduct.

We first present a natural extension of the process of grounding with respect to many-sorted interpretations.

Let F be any first-order sentence of a many-sorted signature σ, and let I be a (many-sorted) interpretation of σ. By $gr_I[F]$ we denote the infinitary ground formula w.r.t. I that is obtained from F by the following process:

- If F is an atomic formula, $gr_I[F]$ is F;
- $gr_I[G \odot H] = gr_I[G] \odot gr_I[H] \quad (\odot \in \{\wedge, \vee, \rightarrow\})$;
- $gr_I[\exists x G(x)] = \{gr_I[G(\xi^\diamond)] \mid \xi \in |I|^s\}^\vee$ where s is the sort of the variable x;
- $gr_I[\forall x G(x)] = \{gr_I[G(\xi^\diamond)] \mid \xi \in |I|^s\}^\wedge$ where s is the sort of the variable

x.

We say for two interpretations I, J of multi-valued signature σ and a set of predicate and function constants $\boldsymbol{c}$ that $J <^{\mathbf{c}} I$ if

- $|I|^s = |J|^s$ for every sort $s \in \sigma$;
- I and J agree on all constants not in $\boldsymbol{c}$;
- $p^J \subseteq p^I$ for all predicates p in $\boldsymbol{c}$;
- I and J do not agree on $\boldsymbol{c}$.

Example 18 *Consider four interpretations I, J, K, L with universe $\{1, 2, 3\}$ and signature $\sigma = \{p, q, f, s\}$ where p and q are unary predicates, f is a unary function, and s is a sort. Let $\mathbf{c}$ be $\{p, f\}$. When*

$$
\begin{aligned}
I = &\quad \{p(1), q(1), f = 1, s = \{1, 2\}\} \\
J = &\quad \{p(1), q(1), f = 2, s = \{1, 2\}\} \\
K = &\quad \{q(1), f = 1, s = \{1, 2\}\} \\
L = &\quad \{q(1), f = 1, s = \{1, 2, 3\}\}
\end{aligned}
$$

we can see that $J <^{\mathbf{c}} I$ holds since $p^J \subseteq p^I$ (both have an extent of $\{1\}$) and J and I do not agree on $\mathbf{c}$ since $f^I \neq f^J$. Similarly, $K <^{\mathbf{c}} I$ holds since $p^K \subseteq p^I$ (the former has an empty extent while the latter has extent $\{1\}$) and K and I disagree on p. On the other hand, $L <^{\mathbf{c}} I$ does not hold since L and I do not agree on the sort s.

The *reduct* $F^{\underline{I}}$ of an infinitary ground formula F relative to an many-sorted interpretation I is defined as follows:

- For each atomic formula F, $F^{\underline{I}} = \bot$ if $I \not\models F$ and $F^{\underline{I}} = F$ otherwise;

- $(\mathcal{H}^\wedge)^{\underline{I}} = \bot$ if $I \not\models \mathcal{H}^\wedge$;   otherwise $(\mathcal{H}^\wedge)^{\underline{I}} = \{G^{\underline{I}} \mid G \in \mathcal{H}\}^\wedge$;

- $(\mathcal{H}^\vee)^{\underline{I}} = \bot$ if $I \not\models \mathcal{H}^\vee$;   otherwise $(\mathcal{H}^\vee)^{\underline{I}} = \{G^{\underline{I}} \mid G \in \mathcal{H}\}^\vee$;

- $(G \to H)^{\underline{I}} = \bot$ if $I \not\models G \to H$;   otherwise $(G \to H)^{\underline{I}} = G^{\underline{I}} \to H^{\underline{I}}$.

The following is the many-sorted counterpart to Theorem 1.

Theorem 18 *Let F be a first-order sentence of a many-sorted signature σ and let $\boldsymbol{c}$ be a list of intensional constants. For any interpretation I of σ, $I \models SM[F; \boldsymbol{c}]$ iff*

- *I satisfies F, and*

- *every interpretation J such that $J <^{\mathbf{c}} I$ does not satisfy $(gr_I[F])^{\underline{I}}$.*

8.3 Relation to Multi-valued Propositional Formulas

In this section, we show that multi-valued propositional formulas can be expressed naturally in terms of the many-sorted functional stable model semantics. Given a multi-valued signature σ, we construct the many-sorted signature σ' as follows:

- For every $c \in \sigma$, we have a sort $sort_c \in \sigma'$;

- For every $v \in Dom(c)$ for some $c \in \sigma$, we have a sort $sort_v \in \sigma'$ that is a subsort of every sort $sort_c$ for which $v \in Dom(c)$;

- For every $c \in \sigma$, we include an object constant $c \in \sigma'$ and associate it with the sort $sort_c$;

- For every $v \in Dom(c)$ for some $c \in \sigma$, we include an object constant $v \in \sigma'$ and associate it with the sort $sort_v$.

We define the universes of these sorts as $|I|^{sort_v} = \{v\}$ and $|I|^{sort_c} = Dom(c)$

We identify a multi-valued propositional interpretation X of signature σ with a many-sorted interpretation of signature σ' so that $X(c) = v$ iff $c^X = v$.

Theorem 19 *Let F be a multi-valued propositional formula. a) If X is a stable model of F viewed as a multi-valued formula of signature σ, then X is a stable model of F viewed as a many-sorted formula of signature σ'.*
b) If X is a stable model of F viewed as a many-sorted formula of signature σ', then X is a stable model of F viewed as a multi-valued formula of σ.

8.4 Reducing Many-sorted FSM to Nonsorted FSM

We can represent many-sorted FSM using nonsorted FSM as follows. Given a many-sorted signature σ, we define the signature σ^{ns} to contain every function and predicate constant from σ. In addition, for each sort $s \in \sigma$, we add a unary predicate s to σ^{ns}.

Given a formula F of many-sorted signature σ, we obtain the formula F^{ns} from nonsorted signature σ^{ns} as follows.

We replace every $\exists x\ F(x)$, where x is a sorted variable whose sort is s, with the formula

$$\exists y(F(y) \wedge \mathsf{s}(y))$$

where y is an nonsorted variable. Similarly, we replace every $\forall x\ F(x)$, where x is a sorted variable whose sort is s, with the formula

$$\forall y(\mathsf{s}(y) \rightarrow F(y))$$

where y is an nonsorted variable.

By SF_σ we denote the conjunction of

- the formulas $\forall y(\mathsf{s_i}(y) \rightarrow \mathsf{s_j}(y))$ for every two sorts s_i and s_j in σ such that s_i is a subsort of s_j,

- the formulas $\exists y\ \mathsf{s}(y)$ for every sort s in σ

- the formulas $\forall y_1 \dots y_k(\mathsf{args_1}(y_1) \wedge \dots \wedge \mathsf{args_k}(y_k) \rightarrow \mathsf{vals}(f(y_1, \dots, y_k)))$ for each function constant f in σ where the arity of f is k and the ith argument sort of f is $args_i$ and the value sort of f is $vals$.

- the formulas $\forall y_1 \dots y_{k+1}(\neg\mathsf{args_1}(y_1) \vee \dots \vee \neg\mathsf{args_k}(y_k) \rightarrow \{f(y_1, \dots, y_k) = y_{k+1}\})$ for each function constant f in σ where the arity of f is k and the ith argument sort of f is $args_i$.

- the formulas $\forall y_1 \dots y_k(\neg\mathsf{args_1}(y_1) \vee \dots \vee \neg\mathsf{args_k}(y_k) \rightarrow \{p(y_1, \dots, y_k)\})$ for each function constant f in σ where the arity of f is k and the ith argument sort of f is $args_i$.

Note that only the first 3 items are necessary to turn many-sorted formulas in classical logic to non-sorted formulas. Here, however, we need to add the fourth and fifth item for the FSM semantics so that any constant c and the corresponding variable $\widehat{c}$ in the formula SM for the nonsorted case can only disagree using values according to the many-sorted setting (which has arguments adhering to the argument sorts).

Example 19 *Consider* $\sigma = \{f, s_1, s_2\}$ *where the argument and value sort of* f *are both* s_1. *Take* F *to be* $f(1) = 1 \wedge f(2) = 2$. *The many-sorted interpretation* I *such that* $|I|^{s_1} = \{1, 2\}$, $|I|^{s_2} = \{3, 4\}$, $f(1)^I = 1$, *and* $f(2)^I = 2$ *is clearly a stable model of* F. *However, if we neglect the last two items of* SF_σ, F^{ns} *is*

$$f(1) = 1 \wedge f(2) = 2 \wedge$$

$$\exists y(s_1(y)) \wedge \exists y(s_2(y)) \wedge$$

$$\forall y_1(sort_1(y_1) \rightarrow sort_1(f(y_1)))$$

and K *is a nonsorted interpretation such that* $|K| = \{1, 2, 3, 4\}$, $(s_1)^K = \{1, 2\}$, $(s_2)^K = \{3, 4\}$, $f(1)^K = 1$, $f(2)^K = 2$, $f(3)^K = 3$, *and* $f(4)^K = 4$ *is not a stable model of* F^{ns} *since we can take* J *that is different from* K *only on* $f(4)$ *so that* $f(4)^J = 3$ *and* J *still satisfies the reduct.*

Also note that the formulas in item 3 are not $\boldsymbol{c}$-plain. This transformation illustrates one use of non-$\boldsymbol{c}$-plain formulas that is unable to be expressed as a $\boldsymbol{c}$-plain as far as we know.

Given an interpretation I of a many-sorted signature σ, we can identify this with the nonsorted signature I^{ns} by taking $|I^{ns}| = \bigcup_{s \text{ is a sort in } \sigma} |I|^s$. We specify that the sort predicates and sorts correspond by defining the sort predicate s for every sort $s \in \sigma$ as

$$\mathsf{s}^{I^{ns}} = |I|^s$$

For every function f in σ and every tuple $\boldsymbol{\xi}$ comprised of elements from $|I^{ns}|$, we take

$$f(\boldsymbol{\xi})^{I^{ns}} = \begin{cases} f(\boldsymbol{\xi})^I & \text{if } \xi_i \in |I|^{args_i} \text{ where } args_i \text{ is the } i\text{th argument sort of } f \\ |I^{ns}|_0 & \text{otherwise} \end{cases}$$

where $|I^{ns}|_0$ denotes some element in the universe (we use the same element for every situation this case holds).

For every predicate p in σ and every $\boldsymbol{\xi}$ we take

$$p(\boldsymbol{\xi})^{I^{ns}} = \begin{cases} p(\boldsymbol{\xi})^I & \text{if } \xi_i \in |I|^{args_i} \text{ where } args_i \text{ is the } i\text{th argument sort of } p \\ \boldsymbol{f} & \text{otherwise.} \end{cases}$$

Note that $\boldsymbol{f}$ was arbitrarily chosen.

The choice of I^{ns} mapping a function whose arguments are not of the intended sort to the value $|I^{ns}|_0$ is arbitrary and so there are many unsorted interpretations that correspond to the many-sorted interpretation. To characterize this many-to-one relationship, we say two unsorted interpretations I and J are *related* with relation R, denoted $R(I, J)$, if for every predicate or function constant c, we have $c(\xi_1, \ldots, \xi_k)^I = c(\xi_1, \ldots, \xi_k)^J$ whenever each $\xi_i \in args_i$ where $args_i$ is the ith argument sort of c.

Theorem 20 *Given a formula F of a many-sorted signature σ, and a set of function and predicate constants $\boldsymbol{c}$,*
a) An interpretation I of signature σ is a model of SM$[F; \boldsymbol{c}]$ iff I^{ns} is a model of SM$[F^{ns} \wedge SF_\sigma; \boldsymbol{c}]$.
b) An interpretation L_1 of signature σ^{ns} is a model of SM$[F^{ns} \wedge SF_\sigma; \boldsymbol{c}]$ iff there is some interpretation L of signature σ^{ns} such that $R(L, L_1)$ and $L = I^{ns}$ for some model I of SM$[F; \boldsymbol{c}]$.

8.5 ASPMT as a Special Case of Many-Sorted FSM

In this section, we present a special case of Many-Sorted FSM–answer set programming modulo theories (ASPMT). This is a framework which extends answer set programming analogously to how SMT extends SAT. We then present a prototype implementation of this framework in Section 9.2. We expect this framework to elicit similar benefits to those that SMT provided over SAT.

Formally, an SMT instance is a formula in many-sorted first-order logic, where some designated function and predicate constants are constrained by some fixed background interpretation. SMT is the problem of determining whether such a formula has a model that expands the background interpretation Barrett *et al.* (2009).

The syntax of ASPMT is the same as that of SMT. Let σ^{bg} be the (many-sorted) signature of the background theory bg. An interpretation of σ^{bg} is called a *background interpretation* if it satisfies the background theory. For instance, in the theory of reals, we assume that σ^{bg} contains the set $\mathcal{R}$ of symbols for all real numbers, the set of arithmetic functions over real numbers, and the set $\{<, >, \leq, \geq\}$ of binary predicates over real numbers. Background interpretations interpret these symbols in the standard way.

Let σ be a signature that is disjoint from σ^{bg}. We say that an interpretation I of σ satisfies F w.r.t. the background theory bg, denoted by $I \models_{bg} F$, if there is a background interpretation J of σ^{bg} that has the same universe as I, and $I \cup J$ satisfies F. For any ASPMT sentence F with background theory σ^{bg}, interpretation I is a stable model of F relative to $\boldsymbol{c}$ (w.r.t. background theory σ^{bg}) if $I \models_{bg} \mathrm{SM}[F; \boldsymbol{c}]$.

Example 7 continued *Formula F can be understood as an ASPMT formula with the theory of integers as the background theory. Arithmetic functions and comparison operators belong to the background signature. Let I be an interpretation of signature $\{Amount_0, Amount_1, FillUp\}$ such that $Amount_0^I = 6$, $Amount_1^I = 5$, $FillUp^I = \boldsymbol{f}$. We say that $I \models_{bg} SM[F; Amount_1]$.*

8.6 Proofs

8.6.1 Proof of Theorem 18

Given an interpretation J of many-sorted signature σ, a set of constants $\boldsymbol{c} \subseteq \sigma$, and a set of constants $\boldsymbol{d}$ that is disjoint from σ and is of the same length as $\boldsymbol{c}$ whose corresponding elements have the same argument and value sorts, $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ is the interpretation from $(\sigma \setminus \boldsymbol{c}) \cup \boldsymbol{d}$ obtained from J by replacing every constant from $\boldsymbol{c}$ with the corresponding constant from $\boldsymbol{d}$.

For two interpretations I and J of the same many-sorted signature σ, a set of constants $\boldsymbol{c} \subseteq \sigma$ such that I and J agree on constants in $\sigma \setminus \boldsymbol{c}$ and a set of constants $\boldsymbol{d}$ of the same length as $\boldsymbol{c}$ (whose constants have the same argument / value sorts) that is disjoint from σ, we define $I \cup J^{\boldsymbol{c}}_{\boldsymbol{d}}$ as the interpretation from the extended signature $\sigma \cup \boldsymbol{d}$ such that

- $I \cup J^{\boldsymbol{c}}_{\boldsymbol{d}}$ agrees with both I and J on constants in $\sigma \setminus \boldsymbol{c}$
- $I \cup J^{\boldsymbol{c}}_{\boldsymbol{d}}$ agrees with I for all constants in $\boldsymbol{c}$ and
- $I \cup J^{\boldsymbol{c}}_{\boldsymbol{d}}$ agrees with $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ for all constants in $\boldsymbol{d}$.

Lemma 21 *If F is a sentence, I and J are interpretations of the same many-sorted signature and $J <^{\boldsymbol{c}} I$, then $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $J \models gr_I(F)^{\underline{I}}$.*

Proof.

- Case 1: F is a variable-free atomic formula containing no intensional constants. In this case, $F^*(\boldsymbol{d}) = F$ so $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $I \models F$ iff $J \models F$ (recall I and J agree on all non-intensional constants). On the other hand, $gr_I(F)^{\underline{I}}$ is F if $I \models F$ and $\perp$ if $I \not\models F$. In both cases, since I and J agree on all non-intensional constants, $J \models gr_I(F)^{\underline{I}}$ iff $J \models F$ so the claim holds in this case.

- Case 2: F is a variable-free atomic formula containing an intensional constant. $F^*(\boldsymbol{d})$ is equivalent to $F \wedge F(\boldsymbol{d})$.

  Consider the following subcases:

  - Subcase 1: $I \models F$. In this case, $F^*(\boldsymbol{d}) = F \wedge F(\boldsymbol{d})$ so $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $J^{\boldsymbol{c}}_{\boldsymbol{d}} \models F(\boldsymbol{d})$ iff $J \models F$. On the other hand, $gr_I(F)^{\underline{I}} = F$ since there is no subformula that is unsatisfied and thus $J \models gr_I(F)^{\underline{I}}$ iff $J \models F$ so the claim holds in this case.

  - Subcase 2: $I \not\models F$. In this case $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \not\models F \wedge F(\boldsymbol{d}) = F^*(\boldsymbol{d})$. Also in this case, we have $gr_I(F)^{\underline{I}} = \bot$ since the entire formula is not satisfied by I. $J \not\models gr_I(F)^{\underline{I}}$ so the claim holds in this case.

- Case 3: F is $G \wedge H$. By I.H. on G and H.

- Case 4: F is $G \vee H$. By I.H. on G and H.

- Case 5: F is $G \rightarrow H$.

  In this case, $F^*(\boldsymbol{d}) = (G \rightarrow H) \wedge (G^*(\boldsymbol{d}) \rightarrow H^*(\boldsymbol{d}))$. Consider the following subcases:

  - Subcase 1: $I \models G$ and $I \models H$. In this case, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models G^*(\boldsymbol{d}) \rightarrow H^*(\boldsymbol{d})$. On the other hand, $gr_I(F)^{\underline{I}} = G \rightarrow H$ so this case holds by I.H. on G and H.

  - Subcase 2: $I \models G$ and $I \not\models H$. In this case $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \not\models F^*(\boldsymbol{d})$. Also in this case, we have $gr_I(F)^{\underline{I}} = \bot$ since the entire formula is not satisfied by I. $J \not\models gr_I(F)^{\underline{I}}$ so the claim holds in this case.

  - Subcase 3: $I \not\models G$. In this case, $gr_I(F)^{\underline{I}} = \bot \rightarrow H$ or $gr_I(F)^{\underline{I}} = \bot \rightarrow \bot$ depending on whether $I \models H$. In either case, $J \models gr_I(F)^{\underline{I}}$. On the other hand, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models G^*(\boldsymbol{d}) \rightarrow H^*(\boldsymbol{d})$. However, since $I \not\models G$, $gr_I(G)^{\underline{I}} = \bot$. By I.H. on G, we conclude that $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \not\models G^*(\boldsymbol{d})$ (since $J \not\models gr_I(G)^{\underline{I}} = \bot$). Thus $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models G^*(\boldsymbol{d}) \rightarrow H^*(\boldsymbol{d})$ and further $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ so the claim holds in this case.

- Case 6: F is $\exists x G(x)$. By I.H. on $G(\xi)$ for each $\xi \in |I|^s$ where s is the sort of x. By showing $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models G(\xi)^*(\boldsymbol{d})$ iff $J \models gr_I(G(\xi))^{\underline{I}}$ for each $\xi \in |I|^s$, we prove that $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $J \models gr_I(F)^{\underline{I}}$.

- Case 7: F is $\forall x G(x)$. By I.H. on $G(\xi)$ for each $\xi \in |I|^s$ where s is the sort of x. By showing $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models G(\xi)^*(\boldsymbol{d})$ iff $J \models gr_I(G(\xi))^{\underline{I}}$ for each $\xi \in |I|^s$, we prove that $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models F^*(\boldsymbol{d})$ iff $J \models gr_I(F)^{\underline{I}}$.

∎

Lemma 22 *Given two interpretations I and J of the same many-sorted signature σ, a set of constants $\boldsymbol{c} \subseteq \sigma$, and a set of constants $\mathbf{d}$ disjoint from σ of the same length as $\boldsymbol{c}$, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models \boldsymbol{d} < \boldsymbol{c}$ iff $J <^{\boldsymbol{c}} I$.*

Proof. By definition, $\boldsymbol{d} < \boldsymbol{c}$ is

$$\boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred} \wedge \neg(\boldsymbol{d} = \boldsymbol{c})$$

and by definition, $J <^{\boldsymbol{c}} I$ is

- $|J|^s$ and $|I|^s$ are the same for each sort $s \in \sigma$ and agree on all constants not in $\boldsymbol{c}$;

- $p^J \subseteq p^I$ for all predicate constants p in $\boldsymbol{c}$; and

- J and I do not agree on $\boldsymbol{c}$.

By definition of $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$, $|J|^s$ and $|I|^s$ are the same for each sort $s \in \sigma$ and agree on all constants not in $\boldsymbol{c}$.

By definition, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$ is true iff for every predicate p in $\boldsymbol{c}$

$$\forall \boldsymbol{x}(p(\boldsymbol{x})^{\boldsymbol{c}}_{\boldsymbol{d}} \rightarrow p(\boldsymbol{x}))$$

which is equivalent to saying $(p^{\boldsymbol{c}}_{\boldsymbol{d}})^{J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I} \subseteq p^{J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I}$. Since I does not interpret any constant from $\boldsymbol{d}$ and $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ does not interpret any constant from $\boldsymbol{c}$, this is equivalent to $(p^{\boldsymbol{c}}_{\boldsymbol{d}})^{J^{\boldsymbol{c}}_{\boldsymbol{d}}} \subseteq p^I$ and further to $p^J \subseteq p^I$.

Since I does not interpret any constant from $\boldsymbol{d}$ and $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ does not interpret any constant from $\boldsymbol{c}$, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models \neg(\boldsymbol{d} = \boldsymbol{c})$ is equivalent to saying J and I do not agree on $\boldsymbol{c}$.

Theorem 18 *Let F be a first-order sentence of a many-sorted signature σ and let $\boldsymbol{c}$ be a list of intensional constants. For any interpretation I of σ, $I \models SM[F; \boldsymbol{c}]$ iff*

- *I satisfies F, and*
- *every interpretation J such that $J <^{\mathbf{c}} I$ does not satisfy $(gr_I[F])^{\underline{I}}$.*

Proof.

$I \models \mathrm{SM}[F; \boldsymbol{c}]$ is by definition

$$I \models F \wedge \neg\exists \widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} < \boldsymbol{c} \wedge F^*(\widehat{\boldsymbol{c}})). \tag{8.1}$$

The first item, "I satifies F", is equivalent to the first conjunctive term of (8.1).

By Lemma 16 and Lemma 22, the second item, "no interpretation J of σ such that $J <^{\mathbf{c}} I$ satisfies $gr_I(F)^{\underline{I}}$", is equivalent to the second conjunctive term in (8.1). ∎

8.6.2 *Proof of Theorem 19*

The following lemma shows that X satisfies a multi-valued formula F iff X satisfies F when viewed as a many-sorted propositional formula of signature σ'.

Lemma 23 *Let F be a multi-valued formula of signature σ. A multi-valued interpretation X satisfies F iff X satisfies F when we view both to be from the many sorted signature σ'.*

Proof. By induction on F. We only show the base case of atomic formulas. When F an atomic formula $c = v$, X satisfies this in the multi-valued sense when $X(c) = v$. For X to satisfy F in the many-sorted sense, it must be that $c^X = v^X$. However by definition of σ', $|X|^v = \{v\}$ so $v^X = v$. Thus it follows that $c^X = v^X$ iff $c^X = v$. Then from the way we identify a multi-valued interpretation and a many-sorted interpretation–$X(c) = v$ iff $c^X = v$–the claim follows.

Lemma 24 *Assume that K and X are multi-valued interpretations of σ and Y is a many-sorted interpretation of σ' such that $Y <^{\mathbf{c}} X$ such that for every $c \in \mathbf{c}$, we have*

$$K(c) = c^Y.$$

We have that $K \models F^X$ (when F is viewed as a multi-valued formula of σ) iff $Y \models F^X$ (when F is viewed as a many-sorted formula of signature σ').

Proof. By induction on F. We show only the case of atoms. The other cases are straightforward.

Let F be an atom $c = v$. If $X \models c = v$, then F^X is F. The claim follows from the assumption since $K \models c = v$ iff $Y \models c = v$. If $X \not\models c = v$, then F^X is $\bot$, which neither K nor Y satisfies. ∎

Theorem 19 *a) If X is a stable model of F viewed as a multi-valued formula of signature σ, then X is a stable model of F viewed as a many-sorted formula of signature σ'.*

b) If X is a stable model of F viewed as a many-sorted formula of signature σ', then X is a stable model of F viewed as a multi-valued formula of σ.

Proof.

Let $\boldsymbol{c}$ denote the list of constants in σ and let $\boldsymbol{c}'$ denote the list of constants in σ'.

By Lemma 23 X satisfies a multi-valued formula F iff X satisfies F when viewed as a many-sorted propositional formula of signature σ'. We need only show that there is an interpretation Y that disagrees with X on σ that is a model of the reduct F^X (when viewed as a multi-valued formula) iff there is an interpretation Y such that $Y <^{\boldsymbol{c}} X$ that is a model of the reduct F^X (when viewed as a many-sorted formula).

The proof is by considering the same identification for Y.

Take any multi-valued interpretation Y of σ and note that we identify Y with a many-sorted interpretation of signature σ'. We consider the meaning of $Y <^{\boldsymbol{c}'} X$ in this context. Since there are no predicate constants in σ', this is equivalent to saying that Y and X have the same universes $|I|^s$ for every sort s, I and J agree on all constants not in $\boldsymbol{c}'$, and I and J do not agree on $\boldsymbol{c}'$. Further since the sorts for each constant $v \in Dom(c)$ for some $c \in \sigma$ contain only one element, it is impossible for Y to disagree with X on these constants. So in order for it to be the case that $Y <^{\boldsymbol{c}'} X$, it must be that Y and X disagree on some $c \in \boldsymbol{c}$. However, this is precisely the same as saying Y disagrees with X on σ when viewed as multi-valued interpetations.

So we have Y disagrees with X on σ iff $Y <^{\boldsymbol{c}'} X$.

It follows from Lemma 24 that X satisfies F^X when viewed as a multi-valued formula iff X satisfies F^X when viewed as a many-sorted propositional formula of signature σ'. ∎

8.6.3 Proof of Theorem 20

Lemma 25 *Given a formula F of many-sorted signature σ and an interpretation I of σ, $I \models gr_I[F]$ iff $I^{ns} \models gr_{I^{ns}}[F^{ns}]$.*

Proof. By induction on F.

- F is $p(\boldsymbol{t})$ where each t_i in $\boldsymbol{t}$ is comprised of ground terms from the extended signature σ^I. $gr_I[F]$ is also $p(\boldsymbol{t})$.
  F^{ns} is $p(\boldsymbol{t})$. $gr_{I^{ns}}[F^{ns}]$ is also $p(\boldsymbol{t})$. By the definition of I^{ns}, $p(\boldsymbol{t})^I = p(\boldsymbol{t})^{I^{ns}}$ since $\boldsymbol{t}$ must be comprised of terms from the corresponding argument sorts of p and so the claim holds.

- F is $t_1 = t_2$ where each t_i is comprised of ground terms from the extended signature σ^I. $gr_I[F]$ is also $t_1 = t_2$. F^{ns} is $t_1 = t_2$. $gr_{I^{ns}}[F^{ns}]$ is also $t_1 = t_2$. By the definition of I^{ns}, $t_1^I = t_1^{I^{ns}}$ and $t_2^I = t_2^{I^{ns}}$ since the subterms of t_1 and t_2 must be comprised of terms from the corresponding argument sorts and so the claim holds.

- F is $G \odot H$ where $\odot \in \{\wedge, \vee, \rightarrow\}$. $gr_I[F]$ is $gr_I[G] \odot gr_I[H]$. F^{ns} is $G^{ns} \odot H^{ns}$. $gr_{I^{ns}}[F^{ns}]$ is $gr_{I^{ns}}[G^{ns}] \odot gr_{I^{ns}}[H^{ns}]$ so the claim follows by induction on G and H.

- F is $\exists x G(x)$. $gr_I[F]$ is $\{gr_I[G(\xi^\diamond)] : \xi \in |I|^s\}^\vee$ where s is the sort of x.
  F^{ns} is $\exists y(G(y)^{ns} \wedge \mathsf{s}(y))$. $gr_{I^{ns}}[F^{ns}]$ is $\{gr_{I^{ns}}[G(\xi^\diamond)^{ns}] \wedge \mathsf{s}(\xi^\diamond) : \xi \in |I^{ns}|\}^\vee$.

(⇒) Assume $I \models gr_I[F]$. That is, assume there is some $\xi \in |I|^s$ where s is the sort of x such that $I \models gr_I[G(\xi^\diamond)]$. By definition of I^{ns}, since $\xi \in |I|^s$, then $I^{ns} \models \mathsf{s}(\xi^\diamond)$. So then, the claim follows by I.H. on $G(\xi^\diamond)$.

(⇐) Assume $I^{ns} \models gr_{I^{ns}}[F^{ns}]$. That is, assume there is some $\xi \in |I^{ns}|$ such that $I^{ns} \models gr_{I^{ns}}[G(\xi^\diamond)^{ns}] \wedge \mathsf{s}(\xi^\diamond)$. By definition of I^{ns}, since $I^{ns} \models \mathsf{s}(\xi^\diamond)$, then $\xi \in |I|^s$. SO then, the claim follows by I.H. on $G(\xi^\diamond)$.

- F is $\forall x G(x)$. $gr_I[F]$ is $\{gr_I[G(\xi^\diamond)] : \xi \in |I|^s\}^\wedge$ where s is the sort of x.
  F^{ns} is $\forall y(\mathsf{s}(y) \rightarrow G(y)^{ns})$. $gr_{I^{ns}}[F^{ns}]$ is $\{\mathsf{s}(\xi^\diamond) \rightarrow gr_{I^{ns}}[G(\xi^\diamond)^{ns}] : \xi \in |I^{ns}|\}^\wedge$.
  (⇒) Assume $I \models gr_I[F]$. That is, for every $\xi \in |I|^s$ where s is the sort of x, assume that $I \models gr_I[G(\xi^\diamond)]$. Note that for every $\xi \in |I^{ns}|$ such that $I^{ns} \not\models \mathsf{s}(\xi^\diamond)$, we have that I^{ns} vacuously satisfies $\mathsf{s}(\xi^\diamond) \rightarrow gr_{I^{ns}}[G(\xi^\diamond)^{ns}]$. By definition of I^{ns}, since $\xi \in |I|^s$ iff $I^{ns} \models \mathsf{s}(\xi^\diamond)$ the claim follows by I.H. on $G(\xi^\diamond)$ for every $\xi \in |I|^s$.

  (⇐) Assume $I^{ns} \models gr_{I^{ns}}[F^{ns}]$. That is, assume for every $\xi \in |I^{ns}|$ that $I^{ns} \models \mathsf{s}(\xi^\diamond) \rightarrow gr_{I^{ns}}[G(\xi^\diamond)^{ns}]$. This means that for every ξ such that $I^{ns} \models \mathsf{s}(\xi^\diamond)$, it must be that $I^{ns} \models gr_{I^{ns}}[G(\xi^\diamond)^{ns}]$.

  Now, by definition of I^{ns}, for any ξ such that $I^{ns} \models \mathsf{s}(\xi^\diamond)$, we have that $\xi \in |I|^s$. So then, the claim follows by I.H. on $G(\xi^\diamond)$ for every $\xi \in |I|^s$. ∎

∎

Lemma 26 *Given a formula F of many-sorted signature σ, interpretations I and J of σ and an interpretation K of σ^{ns} such that*

- *for every sort s in σ, $|I|^s = |J|^s = s^K$,*

- *for every predicate and function constant c and for every tuple $\boldsymbol{\xi}$ composed of elements from $|I^{ns}|$ such that $\xi_i \in |I|^{args_i}$ for every $\xi_i \in \boldsymbol{\xi}$, where $args_i$ is the ith argument sort of c, we have $c(\boldsymbol{\xi})^K = c(\boldsymbol{\xi})^J$,*

- *for every predicate and function constant c and for every tuple $\boldsymbol{\xi}$ composed of elements from $|I^{ns}|$ such that $\xi_i \notin |I|^{args_i}$ for some $\xi_i \in |I|^{args_i}$, where $args_i$ is the ith argument sort of c, we have $c(\boldsymbol{\xi})^K = c(\boldsymbol{\xi})^{I^{ns}}$,*

J is a model of $gr_I[F]^{\underline{I}}$ iff K is a model of $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$.

Proof. By induction on F.

- F is $p(\boldsymbol{t})$ where each t_i in $\boldsymbol{t}$ is comprised of ground terms from the extended signature σ^I.

  F^{ns} is $p(\boldsymbol{t})$.

  We consider two cases:

  - If $I \models p(\boldsymbol{t})$, then $gr_I[F]^{\underline{I}}$ is $p(\boldsymbol{t})$. By Lemma 25 , it follows that $I^{ns} \models p(\boldsymbol{t})$ and so $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ is $p(\boldsymbol{t})$. Thus, in this case, J is a model of $gr_I[F]^{\underline{I}}$ iff K is a model of $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$.
  - If $I \not\models p(\boldsymbol{t})$, then $gr_I[F]^{\underline{I}}$ is $\bot$. By Lemma 25 , it follows that $I^{ns} \not\models p(\boldsymbol{t})$ and so $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ is also $\bot$. Thus, in this case, J is not a model of $gr_I[F]^{\underline{I}}$ and K is not a model of $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ so the claim follows.

- F is $t_1 = t_2$ where each t_i is comprised of ground terms from the extended signature σ^I.

  F^{ns} is $t_1 = t_2$.

  We consider two cases:

– If $(t_1)^I = (t_2)^I$, then $gr_I[F]^{\underline{I}}$ is $t_1 = t_2$. By Lemma 25 , it follows that $(t_1)^{I^{ns}} = (t_2)^{I^{ns}}$ and so $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ is $t_1 = t_2$. Thus, in this case by the second item in the requirement of this lemma, J is a model of $gr_I[F]^{\underline{I}}$ iff K is a model of $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$.

– If $(t_1)^I \neq (t_2)^I$, then $gr_I[F]^{\underline{I}}$ is $\bot$. By Lemma 25 , it follows that $(t_1)^{I^{ns}} \neq (t_2)^{I^{ns}}$ and so $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ is also $\bot$. Thus, in this case, J is not a model of $gr_I[F]^{\underline{I}}$ and K is not a model of $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ so the claim follows.

- F is $G \odot H$ where $\odot \in \{\wedge, \vee, \rightarrow\}$.

  F^{ns} is $G^{ns} \odot H^{ns}$. We consider two cases:

  – If $I \models G \odot H$, then $gr_I[F]^{\underline{I}}$ is $gr_I[G]^{\underline{I}} \odot gr_I[H]^{\underline{I}}$. By Lemma 25, $I^{ns} \models G^{ns} \odot H^{ns}$ and so $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ is $gr_{I^{ns}}[G^{ns}]^{\underline{I}^{ns}} \odot gr_{I^{ns}}[H^{ns}]^{\underline{I}^{ns}}$ so the claim follows by induction on G and H.

  – If $I \not\models G \odot H$ then $gr_I[F]^{\underline{I}}$ is $\bot$. By Lemma 25, $I^{ns} \not\models G^{ns} \odot H^{ns}$ and so $(F^{ns})^{\underline{I}^{ns}}$ is $\bot$. Thus, in this case, J is not a model of $gr_I[F]^{\underline{I}}$ and K is not a model of $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ so the claim follows.

- F is $\exists x(G(x))$ where the sorted variable x has sort s.

  F^{ns} is $\exists y(G(y)^{ns} \wedge \mathsf{s}(y))$ (the variable here is unsorted).

  $gr_I[F]$ is $\{gr_I[G(\xi^\diamond)] : \xi \in |I|^{\mathsf{s}}\}^\vee$.

  $gr_{I^{ns}}[F^{ns}]$ is $\{gr_{I^{ns}}[G(\xi^\diamond)^{ns}] \wedge \mathsf{s}(\xi^\diamond) : \xi \in |I^{ns}|\}^\vee$.

  $gr_I[F]^{\underline{I}}$ is equivalent to

$$\{gr_I[G(\xi^\diamond)]^{\underline{I}} : \xi \in |I|^{\mathsf{s}} \text{ and } I \models gr_I[G(\xi^\diamond)]\}^\vee.$$

$gr_{I^{ns}}[F^{ns}]_{\underline{I}}^{I^{ns}}$ is equivalent to

$$\{gr_{I^{ns}}[G(\xi^\diamond)^{ns}]_{\underline{I}}^{I^{ns}} \wedge \mathsf{s}(\xi^\diamond) : \xi \in |I^{ns}| \text{ and } I^{ns} \models gr_{I^{ns}}[G(\xi^\diamond)^{ns}] \wedge \mathsf{s}(\xi^\diamond)\}^\vee.$$

Further, since $I^{ns} \models \mathsf{s}(\xi^\diamond)$ iff ξ is from $|I|^s$ and by the first item in the requirement of this lemma, $K \models gr_{I^{ns}}[F^{ns}]_{\underline{I}}^{I^{ns}}$ iff

$$K \models \{gr_{I^{ns}}[G(\xi^\diamond)^{ns}]_{\underline{I}}^{I^{ns}} : \xi \in |I|^{\mathsf{s}} \text{ and } I^{ns} \models gr_{I^{ns}}[G(\xi^\diamond)^{ns}]\}^\vee.$$

Then, by I.H. on each $G(\xi^\diamond)$ such that $\xi \in |I|^{\mathsf{s}}$ and $I \models G(\xi^\diamond)$, we have that $J \models gr_I[G(\xi^\diamond)]_{\underline{I}}^{\underline{I}}$ iff $K \models gr_{I^{ns}}[G(\xi^\diamond)^{ns}]_{\underline{I}}^{I^{ns}}$, from which the claim then follows.

- F is $\forall x(G(x))$ where the sorted variable x has sort s.

  F^{ns} is $\forall x(\mathsf{s}(y) \to G(y))$ (the variable here is unsorted).

  $gr_I[F]$ is $\{gr_I[G(\xi^\diamond)] : \xi \in |I|^{\mathsf{s}}\}^\wedge$.

  $gr_{I^{ns}}[F^{ns}]$ is $\{\mathsf{s}(\xi^\diamond) \to gr_{I^{ns}}[G(\xi^\diamond)^{ns}] : \xi \in |I^{ns}|\}^\wedge$.

  We consider two cases:

  – If $I \models G(\xi^\diamond)$ for every $\xi \in |I|^{\mathsf{s}}$, then $gr_I[F]_{\underline{I}}^{\underline{I}}$ is equivalent to

  $$\{gr_I[G(\xi^\diamond)]_{\underline{I}}^{\underline{I}} : \xi \in |I|^{\mathsf{s}}\}^\wedge.$$

  For every $\xi \notin |I|^{\mathsf{s}}$, $I^{ns} \not\models \mathsf{s}(\xi^\diamond)$ and so in $gr_{I^{ns}}[F^{ns}]_{\underline{I}}^{I^{ns}}$, the implications corresponding to such ξ are vacuously satisfied and so $gr_{I^{ns}}[F^{ns}]_{\underline{I}}^{I^{ns}}$ is equivalent to

  $$\{\mathsf{s}(\xi^\diamond)^{I^{ns}} \to gr_{I^{ns}}[G(\xi^\diamond)^{ns}]_{\underline{I}}^{I^{ns}} : \xi \in |I|^{\mathsf{s}} \text{ and } I^{ns} \models gr_{I^{ns}}[G(\xi^\diamond)^{ns}]\}^\wedge.$$

  Since $\xi \in |I|^{\mathsf{s}}$ iff $I^{ns} \models \mathsf{s}(\xi^\diamond)$ and since by Lemma 25, $I^{ns} \models gr_{I^{ns}}[G(\xi^\diamond)^{ns}]$ for every $\xi \in |I|^{\mathsf{s}}$, $K \models gr_{I^{ns}}[F^{ns}]_{\underline{I}}^{I^{ns}}$ iff

  $$K \models \{gr_{I^{ns}}[G(\xi^\diamond)^{ns}]^{I^{ns}} : \xi \in |I|^s\}^\wedge.$$

Then, by I.H. on each $G(\xi^\diamond)$ such that $\xi \in |I|^{\mathsf{s}}$, we have that $J \models gr_I[G(\xi^\diamond)]^{\underline{I}}$ iff $K \models gr_{I^{ns}}[G(\xi^\diamond)^{ns}]^{\underline{I}^{ns}}$, from which the claim then follows.

– If $I \not\models G(\xi^\diamond)$ for some $\xi \in |I|^{\mathsf{s}}$, then $gr_I[F]^{\underline{I}}$ is $\bot$. Since $\xi \in |I|^{\mathsf{s}}$, $I^{ns} \models \mathsf{s}(\xi^\diamond)$ but by Lemma 25, $I^{ns} \not\models gr_{I^{ns}}[G(\xi^\diamond)^{ns}]$ so $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ is $\bot$. In this case, J is not a model of $gr_I[F]^{\underline{I}}$ and K is not a model of $gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$ so the claim follows. ∎

Lemma 27 *Given a formula F of many-sorted signature σ and two interpretations L and L_1 of σ^{ns} such that $R(L, L_1)$, if $L \models F^{ns} \wedge SF_\sigma$, then $L_1 \models F^{ns} \wedge SF_\sigma$.*

Proof. We first show that $L_1 \models SF_\sigma$. Since $R(L, L_1)$, L and L_1 agree on all sort predicates s corresponding to sorts $s \in \sigma$. Thus, L_1 clearly satisfies the first two items of SF_σ. We now consider the third item of SF_σ. For tuples $\xi_1, \ldots, \xi_k$ such that each $\xi_i \in args_i$ where $args_i$ is the ith argument sort of f, since $R(L, L_1)$, L and L_1 agree on $f(\xi_1, \ldots, \xi_k)$ so L_1 satisfies the implication. For all other tuples, the implication is vacuously satisfied. Finally, the fourth and fifth items of SF_σ are tautologies in classical logic so we conclude that $L_1 \models SF_\sigma$.

We now show that $L_1 \models F^{ns}$ by induction on F^{ns}.

- F^{ns} is $p\boldsymbol{t}$ where $\boldsymbol{t}$ is a ground term from the extended signature σ^I. Since every $t_i \in \boldsymbol{t}$ must be from the ith argument sort of p, it follows from $R(L, L_1)$ that $L_1 \models F^{ns}$.

- F^{ns} is $t_1 = t_2$ where t_1 and t_2 are ground terms from the extended signature σ^I. Since every subterm of t_1 and t_2 must be from the the appropriate sort, it follows from $R(L, L_1)$ that $L_1 \models F^{ns}$.

- F^{ns} is $G^{ns} \odot H^{ns}$ where $\odot \in \{\wedge, \vee, \rightarrow\}$. The claim follows by I.H. on G^{ns} and H^{ns}.

- F^{ns} is $\exists y(G(y) \wedge \mathsf{s}(y))$. Since we assume there $L \models F^{ns}$, there is some $\xi \in |I^{ns}|$ such that $L \models G(\xi^\diamond) \wedge \mathsf{s}(\xi^\diamond)$. Further, since $L \models \mathsf{s}(\xi^\diamond)$ iff $\xi \in |I|^s$, the claim follows by I.H. on $G(\xi^\diamond)$.

- F^{ns} is $\forall y(\mathsf{s}(y) \rightarrow G(y))$. Since we assume there $L \models F^{ns}$, for every $\xi \in |I^{ns}|$ we have $L \models \mathsf{s}(\xi^\diamond) \rightarrow G(\xi^\diamond)$. For every $\xi \notin |I|^s$, L_1 vacuously satisfies $\mathsf{s}(\xi^\diamond) \rightarrow G(\xi^\diamond)$. For every $\xi \in |I|^s$, since $L_1 \models \mathsf{s}(\xi^\diamond)$ iff $\xi \in |I|^s$, the claim follows by I.H. on every $G(\xi^\diamond)$ such that $\xi \in |I|^s$.

Lemma 28 *Given a formula F of many-sorted signature σ, a set of function and predicate constants $\mathbf{c}$ from σ and two interpretations L and L_1 of σ^{ns} such that $R(L, L_1)$, if L is a stable model of $F^{ns} \wedge SF_\sigma$ w.r.t. $\mathbf{c}$, then L_1 is a stable model of $F^{ns} \wedge SF_\sigma$ w.r.t. $\mathbf{c}$.*

Proof. We first note $\mathbf{c}$ contains function and predicate constants from σ and thus contains none of the sort predicates introduced in σ^{ns}.

We assume that L is a stable model of $F^{ns} \wedge SF_\sigma$, and wish to show that L_1 is a stable model of $F^{ns} \wedge SF_\sigma$. That is, given that $L \models F^{ns} \wedge SF_\sigma$ and there is no interpretation K such that $K <^{\mathbf{c}} L$ and $K \models gr_L[F^{ns} \wedge SF_\sigma]^{\underline{L}}$, we wish to show that there is no interpretation K_1 such that $K_1 <^{\mathbf{c}} L_1$ and $K_1 \models gr_{L_1}[F^{ns} \wedge SF_\sigma]^{\underline{L_1}}$. Equivalently, we will show that if there is an interpretation K_1 such that $K_1 <^{\mathbf{c}} L_1$ and $K_1 \models gr_{L_1}[F^{ns} \wedge SF_\sigma]^{\underline{L_1}}$, then there is an interpretation K such that $K <^{\mathbf{c}} L$ and $K \models gr_L[F^{ns} \wedge SF_\sigma]^{\underline{L}}$.

Assume that there is an interpretation K_1 such that $K_1 <^{\mathbf{c}} L_1$ and $K_1 \models gr_{L_1}[F^{ns} \wedge SF_\sigma]^{\underline{L_1}}$, we construct K as follows.

- $|K| = |K_1|$,
- $\mathsf{s}^K = \mathsf{s}^{K_1}$ for every s corresponding to a sort $s \in \sigma$,
- $c(\xi_1, \ldots, \xi_k)^K = c(\xi_1, \ldots, \xi_k)^{K_1}$ for every tuple $\xi_1, \ldots, \xi_k$ such that $\xi_i \in s_i$ where s_i is the ith argument sort of c,
- $c(\xi_1, \ldots, \xi_k)^K = c(\xi_1, \ldots, \xi_k)^L$ for every tuple $\xi_1, \ldots, \xi_k$ such that $\xi_i \notin s_i$ for some i where s_i is the ith argument sort of c.

We first show that $K <^{\mathbf{c}} L$. By definition $|K| = |K_1|$. From $K_1 <^{\mathbf{c}} L_1$, it follows that $|K| = |L_1|$. Then since $R(L_1, L)$, it follows that $|K| = |L|$. By definition of K, it follows that $\mathsf{s}^K = \mathsf{s}^{K_1}$ for every s corresponding to a sort $s \in \sigma$. Then, since $K_1 <^{\mathbf{c}} L_1$ and since $R(L_1, L)$, it follows that $\mathsf{s}^K = \mathsf{s}^L$. Now, for any function or predicate c and any tuple $\xi_1, \ldots, \xi_k$ such that $\xi_i \notin s_i$ for some i where s_i is the ith argument sort of c, by definition, $c(\xi_1, \ldots, \xi_k)^K = c(\xi_1, \ldots, \xi_k)^L$. Finally, for every function or predicate c and every tuple $\xi_1, \ldots, \xi_k$ such that $\xi_i \in s_i$ where s_i is the ith argument sort of c, since $R(L, L_1)$, it is clear that $c(\xi_1, \ldots, \xi_k)^{L_1} = c(\xi_1, \ldots, \xi_k)^L$. We also have by definition, $c(\xi_1, \ldots, \xi_k)^K = c(\xi_1, \ldots, \xi_k)^{K_1}$ for such predicate (functions) and tuples.

Now since we assume that $K_1 <^{\mathbf{c}} L_1$, there must be some function or predicate constant c and some tuple $\xi_1, \ldots, \xi_k$ such that $c(\xi_1, \ldots, \xi_k)^{K_1} \neq c(\xi_1, \ldots, \xi_k)^{L_1}$. Now by definition of $K_1 <^{\mathbf{c}} L_1$, K_1 and L_1 agree on all of the sort predicates s coming from sorts $s \in \sigma$. Further, since $K_1 \models (SF_\sigma)^{\underline{L_1}}$, the fourth and fifth items of $(SF_\sigma)^{\underline{L_1}}$ force K_1 to agree with L_1 on all functions (predicates) and tuples such that some tuple is not of the correct sort. Thus, it must be that the tuple $\xi_1, \ldots, \xi_k$ such that $c(\xi_1, \ldots, \xi_k)^{K_1} \neq c(\xi_1, \ldots, \xi_k)^{L_1}$ has that every ξ_i belongs to the appropriate sort. Thus, by the observation before that $c(\xi_1, \ldots, \xi_k)^{L_1} = c(\xi_1, \ldots, \xi_k)^L$ and $c(\xi_1, \ldots, \xi_k)^K = c(\xi_1, \ldots, \xi_k)^{K_1}$, it follows that $K <^{\mathbf{c}} L$.

Now, we show that $K \models gr_L[SF_\sigma]^{\underline{L}}$ by considering each item of SF_σ. We first note that since $K_1 \models gr_{L_1}[SF_\sigma]^{\underline{L_1}}$, it must be that $L_1 \models gr_{L_1}[SF_\sigma]$. Thus by Lemma 27, we have that $L \models gr_L[SF_\sigma]$.

- Item 1: $\forall y(\mathsf{s}_i(y) \rightarrow \mathsf{s}_j(y))$ for every two sorts s_i and s_j in σ such that s_i is a subsort of s_j.

  From $K_1 <^{\boldsymbol{c}} L_1$, it follows that $\mathsf{s}_i(\xi)^{K_1} = \mathsf{s}_i(\xi)^{L_1}$ for every predicate s corresponding to a sort $s \in \sigma$ and for every ξ in $|L_1| = |K_1|$. By definition of K, and since $R(L, L_1)$, we then have that $\mathsf{s}_i(\xi)^K = \mathsf{s}_i(\xi)^{K_1} = \mathsf{s}_i(\xi)^{L_1} = \mathsf{s}_i(\xi)^L$ so clearly the claim holds for this item.

- Item 2: $\exists y(\mathsf{s}(y))$ for every sort s in σ.

  By the same argument in Item 1, $\mathsf{s}_i(\xi)^K = \mathsf{s}_i(\xi)^{K_1} = \mathsf{s}_i(\xi)^{L_1} = \mathsf{s}_i(\xi)^L$ so clearly the claim holds for this item.

- the formulas $\forall y_1 \dots y_k(\mathsf{a}rgs_1(y_1) \wedge \dots \wedge \mathsf{a}rgs_k(y_k) \rightarrow \mathsf{v}als(f(y_1, \dots, y_k)))$ for each function constant f in σ where the arity of f is k and the ith argument sort of f is $args_i$ and the value sort of f is $vals$.

  By $R(L, L_1)$, for every $\xi_1, \dots, \xi_k$ such that $\xi_i \in args_i$, we have that $f(\xi_1, \dots, \xi_k)^L = f(\xi_1, \dots, \xi_k)^{L_1})$. Then, by definition of K, $f(\xi_1, \dots, \xi_k)^K = f(\xi_1, \dots, \xi_k)^{K_1})$ so the claim holds for this item.

- the formulas $\forall y_1 \dots y_{k+1}(\neg \mathsf{a}rgs_1(y_1) \vee \dots \vee \neg \mathsf{a}rgs_k(y_k) \rightarrow \{f(y_1, \dots, y_k) = y_{k+1}\})$

  for each function constant f in σ where the arity of f is k and the ith argument sort of f is $args_i$.

By definition of K, $f(\xi_1, \ldots, \xi_k)^K = f(\xi_1, \ldots, \xi_k)^L$ and since the reduct of these formulas is only satisfied when K agrees with L for these tuples, the claim holds for this item.

- the formulas $\forall y_1 \ldots y_k(\neg \mathsf{a}rgs_1(y_1) \vee \cdots \vee \neg \mathsf{a}rgs_k(y_k) \rightarrow \{p(y_1, \ldots, y_k)\})$ for each function constant f in σ where the arity of f is k and the ith argument sort of f is $args_i$.

  By definition of K, $p(\xi_1, \ldots, \xi_k)^K = p(\xi_1, \ldots, \xi_k)^L$ and since the reduct of these formulas is only satisfied when K agrees with L for these tuples, the claim holds for this item.

Finally, we show that $K \models gr_L[F^{ns}]^{\underline{L}}$ iff $K_1 \models gr_{L_1}[F^{ns}]^{\underline{L_1}}$ by induction on F^{ns} and will conclude that since we assume $K_1 \models gr_{L_1}[F^{ns}]^{\underline{L_1}}$, that $K \models gr_L[F^{ns}]^{\underline{L}}$.

- F^{ns} is $p(\boldsymbol{t})$ where each element of $\boldsymbol{t}$ is a ground term from the extended signature σ^I and belongs to the corresponding argument sort of p.
  $gr_L[F^{ns}]^{\underline{L}}$ is the same as $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ by Lemma 27. If $L_1 \not\models p(\boldsymbol{t})$ then $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ is $\perp$ neither K nor K_1 satisfy this reduct so the claim holds. If instead $L_1 \models p(\boldsymbol{t})$ then $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ is $p(\boldsymbol{t})$.

  Then, by definition of K, since $p(\boldsymbol{t})^K = p(\boldsymbol{t})^{K_1}$, clearly the claim holds.

- F^{ns} is $f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ where each element of $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ is a ground term of the extended signature σ^I and belongs to the corresponding argument sort of f_1 and f_2 respectively.
  $gr_L[F^{ns}]^{\underline{L}}$ is the same as $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ by Lemma 27. If $L_1 \not\models f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ then $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ is $\perp$ neither K nor K_1 satisfy this reduct so the claim holds. If instead $L_1 \models f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$ then $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ is $f_1(\boldsymbol{t}_1) = f_2(\boldsymbol{t}_2)$.

Then, by definition of K, since $f_1(\boldsymbol{t}_1)^K = f_1(\boldsymbol{t}_1)^{K_1}$ and $f_2(\boldsymbol{t}_2)^K = f_2(\boldsymbol{t}_2)^{K_1}$, clearly the claim holds.

- F^{ns} is $G^{ns} \odot H^{ns}$ where $\odot \in \{\wedge, \vee, \rightarrow\}$. $gr_L[F^{ns}]^{\underline{L}}$ is $gr_L[G^{ns}]^{\underline{L}} \odot gr_L[H^{ns}]^{\underline{L}}$ and $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ is $gr_{L_1}[G^{ns}]^{\underline{L_1}} \odot gr_{L_1}[H^{ns}]^{\underline{L_1}}$ so the claim follows by I.H. on G^{ns} and H^{ns}.

- F^{ns} is $\exists y(G(y)^{ns} \wedge \mathsf{s}(y))$.
  $gr_L[F^{ns}]^{\underline{L}}$ is equivalent to $\{gr_L[G(\xi^\diamond)^{ns}]^{\underline{L}} : L \models \mathsf{s}(\xi^\diamond)\}^\vee$ and
  $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ is equivalent to $\{gr_{L_1}[G(\xi^\diamond)^{ns}]^{\underline{L_1}} : L_1 \models \mathsf{s}(\xi^\diamond)\}^\vee$. Since $R(L, L_1)$, we have that $\mathsf{s}^L = \mathsf{s}^{L_1}$ and so the claim follows by I.H. on each $G(\xi^\diamond)^{ns}$ such that $L \models \mathsf{s}(\xi^\diamond)$.

- F is $\forall y(\mathsf{s}(y) \rightarrow G(y)^{ns})$. We consider two cases:

  - If $L \not\models G(\xi^\diamond)^{ns}$ for some ξ such that $L \models \mathsf{s}(\xi^\diamond)$, then $gr_L[F^{ns}]^{\underline{L}}$ is $\bot$. By Lemma 27, we have that $L_1 \not\models G(\xi^\diamond)^{ns}$ and so $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ is $\bot$. Thus neither K nor K_1 satisfies the reduct and so the claim holds in this case.

  - Otherwise, $L \models G(\xi^\diamond)^{ns}$ for every ξ such that $L \models \mathsf{s}(\xi^\diamond)$.
    $gr_L[F^{ns}]^{\underline{L}}$ is equivalent to $\{gr_L[G(\xi^\diamond)^{ns}]^L : L \models \mathsf{s}(\xi^\diamond)\}^\wedge$ and
    $gr_{L_1}[F^{ns}]^{\underline{L_1}}$ is equivalent to $\{gr_{L_1}[G(\xi^\diamond)^{ns}]^{L_1} : L_1 \models \mathsf{s}(\xi^\diamond)\}^\wedge$. Since $R(L, L_1)$, we have that $\mathsf{s}^L = \mathsf{s}^{L_1}$ and so the claim follows by I.H. on each $G(\xi^\diamond)^{ns}$ such that $L \models \mathsf{s}(\xi^\diamond)$. ∎

Theorem 20 *Given a formula F of a many-sorted signature σ, and a set of function and predicate constants $\boldsymbol{c}$,*
a) If an interpretation I of signature σ is a model of SM$[F; \boldsymbol{c}]$, then I^{ns} is a model of SM$[F^{ns} \wedge SF_\sigma; \boldsymbol{c}]$.

b) If an interpretation L of signature σ^{ns} is a model of SM$[F^{ns} \wedge SF_\sigma; \boldsymbol{c}]$ then there is some interpretation I of signature σ such that I is a model of SM$[F; \boldsymbol{c}]$ and $R(L, I^{ns})$.

Proof.

a) Consider an interpretation I (of many-sorted signature σ) that is a stable model of F w.r.t. $\boldsymbol{c}$. This means that $I \models F$ and there is no interpretation J such that $J <^{\boldsymbol{c}} I$ and $J \models gr_I[F]^{\underline{I}}$. We wish to show that $I^{ns} \models F^{ns} \wedge SF_\sigma$ and there is no (unsorted) interpretation K such that $K <^{\boldsymbol{c}} I^{ns}$ and $K \models gr_{I^{ns}}[F^{ns} \wedge SF_\sigma]^{\underline{I^{ns}}}$. From Lemma 25, $I \models F$ iff $I^{ns} \models F^{ns}$. It follows from the definition of I^{ns} that $I^{ns} \models SF_\sigma$ so we conclude that $I \models F$ iff $I^{ns} \models F^{ns} \wedge SF_\sigma$. For the second item, we will prove the contrapositive; if there is an (unsorted) interpretation K such that $K <^{\boldsymbol{c}} I^{ns}$ and $K \models gr_{I^{ns}}[F^{ns} \wedge SF_\sigma]^{\underline{I^{ns}}}$, then there is a (many-sorted) interpretation J such that $J <^{\boldsymbol{c}} I$ and $J \models gr_I[F]^{\underline{I}}$.

Assume there is an interpretation K such that $K <^{\boldsymbol{c}} I^{ns}$ and $K \models gr_{I^{ns}}[F^{ns} \wedge SF_\sigma]^{\underline{I^{ns}}}$. We obtain the interpretation J as follows.

For every sort s in σ, $|J|^s = |I|^s$. For every predicate or function c in σ and every tuple $\vec{\xi}$ such that $\xi_i \in |I|^{s_i}$ where s_i is the sort of the ith argument of c, $c(\boldsymbol{\xi})^J = c(\boldsymbol{\xi})^K$. For predicates, it is not hard to see that this is a valid assignment as atoms are either true or false whether considering many-sorted or unsorted logic.

However, for functions, we argue that this assignment is valid. That is, K does not map a function f to a value outside of $|I|^s$ where s is the value sort of f. This follows from the fact that $I^{ns} \models SF_\sigma$ and in particular, the third item of SF_σ. Thus, since $K \models gr_{I^{ns}}[F^{ns} \wedge SF_\sigma]^{\underline{I^{ns}}}$, it follows that K too maps functions to elements of the appropriate sort.

We now show that $J <^{\boldsymbol{c}} I$. Since $K \models gr_{I^{ns}}[SF_\sigma]^{\underline{I^{ns}}}$, the fourth and fifth rules in SF_σ are choice formulas that force K to agree with I^{ns} on every predicate and function c for every tuple that has at least one element outside of the corresponding sort. For every predicate and function c and all tuples that have all elements in the appropriate sort, K and J agree. Further, since I and I^{ns} agree on these as well, it follows immediately since $K <^{\boldsymbol{c}} I^{ns}$, that $J <^{\boldsymbol{c}} I$.

To apply Lemma 26, it is clear that the second condition is satisfied. The first condition follows from the definition of $K <^{\boldsymbol{c}} I^{ns}$; since the sort predicates are not in $\boldsymbol{c}$, K and I^{ns} agree on these predicates. The third item follows from the fact that since $K \models gr_{I^{ns}}[F^{ns} \wedge SF_\sigma]^{\underline{I}^{ns}}$ it follows that $K \models gr_{I^{ns}}[SF_\sigma]^{\underline{I}^{ns}}$. The fourth and fifth rules in SF_σ are choice formulas that force K to agree with I^{ns} for every tuple that has at least one element outside of the corresponding sort. Thus, by Lemma 26, since $K \models gr_{I^{ns}}[F^{ns} \wedge SF_\sigma]^{\underline{I}^{ns}}$ and thus, $K \models gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$, it follows that $J \models gr_I[F]^{\underline{I}}$.

b) Given an interpretation L that is a stable model of $F^{ns} \wedge SF_\sigma$ w.r.t. $\boldsymbol{c}$, we first obtain the interpretation L_1 of σ^{ns} as follows.

- $|L_1| = |L|$,
- $\mathsf{s}^{L_1} = \mathsf{s}^L$ for every s corresponding to a sort s from σ,
- $c(\xi_1, \ldots, \xi_k)^{L_1} = c(\xi_1, \ldots, \xi_k)^L$ for every tuple $\xi_1, \ldots, \xi_k$ such that $\xi_i \in s_i$ where s_i is the ith argument sort of c,
- $c(\xi_1, \ldots, \xi_k)^{L_1} = |L_1|_0$ for every tuple $\xi_1, \ldots, \xi_k$ such that $\xi_i \notin s_i$ for some i where s_i is the ith argument sort of c.

It is easy to see that $R(L, L_1)$. By Lemma 28, L_1 is a stable model of $F^{ns} \wedge SF_\sigma$ w.r.t. $\boldsymbol{c}$. We then obtain the interpretation I of signature σ as follows.

For every sort s in σ, $|I|^s = s^{L_1}$. For every predicate or function c in σ and every tuple $\vec{\xi}$ such that $\xi_i \in |L|^{s_i}$ where s_i is the sort of the ith argument of c, $c(\boldsymbol{\xi})^I = c(\boldsymbol{\xi})^{L_1}$.

For predicates, it is not hard to see that this is a valid assignment as atoms are either true or false whether considering many-sorted or unsorted logic.

However, for functions, we argue that this assignment is valid. That is, I does not map a function f to a value outside of $|I|^s$ where s is the value sort of f. This follows from the fact that $L_1 \models SF_\sigma$ (by Lemma 27) and in particular, the third item of SF_σ. Thus, it follows that I too maps functions to elements of the appropriate sort.

Now it is clear that $L_1 = I^{ns}$ and so we have $R(L, I^{ns})$. We now show that I is a stable model of F.

We have an interpretation I (of many-sorted signature σ) such that I^{ns} is a stable model of $F^{ns} \wedge SF_\sigma$ w.r.t. $\boldsymbol{c}$. This means that $I^{ns} \models F^{ns} \wedge SF_\sigma$ and there is no interpretation K such that $K <^{\boldsymbol{c}} I^{ns}$ and $K \models gr_{I^{ns}}[F^{ns} \wedge SF_\sigma]^{\underline{I}^{ns}}$. We wish to show that $I \models F$ and there is no interpretation J such that $J <^{\boldsymbol{c}} I$ and $J \models gr_I[F]^{\underline{I}}$. From Lemma 25, $I \models F$ iff $I^{ns} \models F^{ns}$ so we conclude that $I \models F$. For the second item, we will prove the contrapositive; if there is a (many-sorted) interpretation J such that $J <^{\boldsymbol{c}} I$ and $J \models gr_I[F]^{\underline{I}}$, then there is an (unsorted) interpretation K such that $K <^{\boldsymbol{c}} I^{ns}$ and $K \models gr_{I^{ns}}[F^{ns} \wedge SF_\sigma]^{\underline{I}^{ns}}$.

Assume there is an interpretation J such that $J <^{\boldsymbol{c}} I$ and $J \models gr_I[F]^{\underline{I}}$. We obtain the interpretation K as follows. $K = J^{ns}$.

We now show that $K <^{\boldsymbol{c}} I^{ns}$. For every predicate and function c for every tuple that has at least one element outside of the corresponding sort, by definition of $K = J^{ns}$, $c^K = c^{I^{ns}} = |I^{ns}|_0$ if c is a function and $c^K = c^{I^{ns}} = 0$ if c is a predicate. That is, for every predicate and function c for every tuple that has at least one element outside of the corresponding sort, K and I^{ns} agree. For every predicate and function c and all tuples that have all elements in the appropriate sort, K and J

agree. Further, since I and I^{ns} agree on these as well, it follows immediately since $J <^{\mathbf{c}} I$, that $K <^{\mathbf{c}} I^{ns}$.

To apply Lemma 26, we must verify the conditions of the lemma. It is clear that the second condition is satisfied. The first condition follows from the definition of $K = J^{ns}$. The third item follows from the observation above; by definition of $K = J^{ns}$, $c^K = c^{I^{ns}} = |I^{ns}|_0$ if c is a function and $c^K = c^{I^{ns}} = 0$ if c is a predicate. Thus, by Lemma 26, since $J \models gr_I[F]^{\underline{I}}$, it follows that $K \models gr_{I^{ns}}[F^{ns}]^{\underline{I}^{ns}}$.

Then, it is easy to see that by definition of I^{ns}, $I^{ns} \models SF_\sigma$. Then, by definition of $K = J^{ns}$, it is clear that $K \models SF_\sigma$. We show that $K \models (SF_\sigma)^{\underline{I}^{ns}}$.

Since K and I^{ns} agree on all sort predicates, it is clear that K models the first two items of $(SF_\sigma)^{\underline{I}^{ns}}$.

Since K and I^{ns} agree on all functions f for tuples $\xi_i, \ldots, \xi_k$ such that each ξ_i is in $|I|^{s_i}$ where s_i is the ith argument sort of f, it is clear that K models the third item of $(SF_\sigma)^{\underline{I}^{ns}}$.

The last two items of $(SF_\sigma)^{\underline{I}^{ns}}$ are only satisfied if K agrees with I^{ns} on all predicates (functions) c and all tuples $\xi_1, \ldots, \xi_k$ such that some ξ_i is not in $|I|^{s_i}$ where s_i is the ith argument sort of c. However, by definition of $K = J^{ns}$ and I^{ns}, both K and I^{ns} map this to $|I^n s|_0$ if c is a function or 0 if c is a predicate so K satisfies these items. So we conclude that $K \models gr_{I^{ns}}[F^{ns} \wedge SF_\sigma]^{\underline{I}^{ns}}$. ∎

Chapter 9

IMPLEMENTATIONS

9.1 MVSM

System MVSM [1] is a prototype implementation of multi-valued propositional formulas under the stable model semantics. In fact, it is a script that invokes the following software: MVPF2LPCOMPILER, F2LP, GRINGO, CLASPD, and AS2TRANSITION. The component MVPF2LPCOMPILER is an implementation of the translations in Theorem 14 from Chapter 7 and Theorem 23 from Chapter 10, which translates total and partial multi-valued propositional formulas respectively into standard propositional formulas under the stable model semantics. As the theorems show, the translations are very similar, and the user can choose which translation to use. Then, F2LP transforms the propositional formula into an ASP program in the input language of GRINGO. Systems GRINGO, CLASPD ground and solve the ASP program respectively. Finally AS2TRANSITION processes the output of CLASPD and produces propositional atoms in the form of multi-valued atoms. The composition of these software is de-

[1] `http://reasoning.eas.asu.edu/mvsm/`

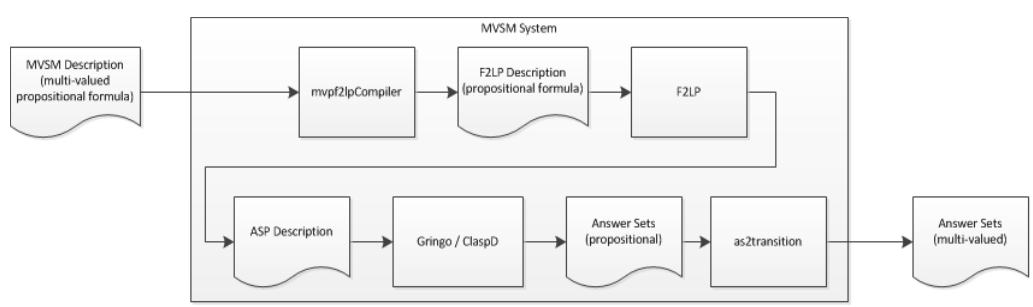

Figure 9.1: Architecture of MVSM

picted in Figure 9.1.

Shown below is a description of the blocks world domain in the language of MVSM assuming the functional stable model semantics. The syntax of declarations follows the one in the input language of the Causal Calculator V2 [2]. Compared to the usual ASP encoding, explicit declaration of sorts and type checking help reduce user error. The inertia and exogeneity assumptions in the last three rules have a simple reading, once we understand $\{F\}$ (the encoding of $Choice(F)$ in MVSM) as representing defaults. There is no need to use both strong negation and default negation.

```
% File 'bw': The blocks world


:- sorts
  step; astep;
  location >> block.


:- objects
  0..maxstep              :: step;
  0..maxstep-1            :: astep;
  1..6                    :: block;
  table                   :: location.


:- variables
  ST                      :: step;
  T                       :: astep;
  Bool                    :: boolean;
```

[2] `http://www.cs.utexas.edu/~tag/cc/`

```
  B,B1                      :: block;
  L                         :: location.

:- constants
  loc(block,step)                   :: location;
  move(block,location,astep)        :: boolean.

% two blocks can't be on the same block at the same time
<- loc(B1,ST)=B & loc(B2,ST)=B & B1!=B2.

% effect of moving a block
loc(B,T+1)=L <- move(B,L,T).

% a block can be moved only when it is clear
<- move(B,L,T) & loc(B1,T)=B.

% a block can't be moved onto a block that is being
% moved also
<- move(B,B1,T) & move(B1,L,T).

% initial location is exogenous
{loc(B,0)=L}.

% actions are exogenous
{move(B,L,T)=Bool}.
```

```
% fluents are inertial
{loc(B,T+1)=L} <- loc(B,T)=L.
```

9.2 ASPMT2SMT

9.2.1 Variable Elimination

Some SMT solvers do not support variables at all (e.g. iSAT) while others suffer in performance when handling variables (e.g. z3). While we can partially ground the input theories, some variables have large (or infinite) domains and should not (cannot) be grounded. Thus, we consider two types of variables; *ASP variables*–variables which should be grounded–and *SMT variables*–variables which should not be grounded. Eliminating ASP variables is simply done by grounding the original ASPMT theory. Then, we consider the problem of equivalently rewriting the completion of the partially ground ASPMT theory so that the result contains no variables.

To ensure that variable elimination can be performed, we impose some syntactic restrictions on ASPMT instances. We first impose that no SMT variable appears in the argument of an uninterpreted function.

We consider ASPMT2SMT programs comprised of rules of the form $H \leftarrow B$ where

- H is $\perp$ or an atom of the form $f(\boldsymbol{t}) = v$, where v is a variable;
- B is a conjunction of atomic formulas possibly preceded with $\neg$.

We define the variable dependency graph of a conjunction of possibly negated atomic formulas $C_1 \wedge \cdots \wedge C_n$ as follows. Nodes of the graph are variables occuring in $C_1 \wedge \cdots \wedge C_n$. There is a directed edge from v to u if there is a C_i that is $v = t$ or $t = v$ for some term t such that u appears in t. We say a variable v depends on a variable u if there is a directed path from v to u in the variable dependency graph. We say a rule $H \leftarrow B$ is *variable isolated* if every variable v in it occurs in an equality $t = v$ or $v = t$ that is positive in B and for the dependency graph of B, v does not depend on v.

Example 20 *The formula $f = X \leftarrow g = 2 * X$ is not variable isolated because X does not occur in an equality $X = t$ or $t = X$. Instead, we write this as $f = X \leftarrow g = Y \wedge Y = 2 * X$ which is variable isolated.*

Example 21 *The formula $f = X \leftarrow 2 * X = Y \wedge 2 * Y = X$ is not variable isolated; although Y occurs in an equality of the form $t = Y$, Y depends on Y (through X).*

The variable elimination is performed modularly so the process need only be described for a single equivalence. Any equivalence in the completion of an ASPMT program with no variables occurring in arguments of uninterpreted functions that is variable isolated will be of the form

$$\forall v(f = v \leftrightarrow \exists \boldsymbol{x}(B_1(v, \boldsymbol{x}) \vee \cdots \vee B_k(v, \boldsymbol{x})))$$

where each B_i is a conjunction of possibly negated literals and has $v = t$ as a non-negated subformula and within B_i, v does not depend on v. In the following, the notation F^v_t denotes the formula obtained from F by replacing every occurrence of the variable v with the term t. We define the process of eliminating variables from such an equivalence E as follows.

1. Given an equivalence $E = \forall v(f = v \leftrightarrow \exists \boldsymbol{x}(B_1(v, \boldsymbol{x}) \vee \cdots \vee B_k(v, \boldsymbol{x})))$

   $F := \forall v(f = v \rightarrow \exists \boldsymbol{x}(B_1(v, \boldsymbol{x}) \vee \cdots \vee B_k(v, \boldsymbol{x})))$

   $G := \forall v(\exists \boldsymbol{x}(B_1(v, \boldsymbol{x}) \vee \cdots \vee B_k(v, \boldsymbol{x})) \rightarrow f = v)$

2. Eliminate variables from F as follows

   (a) $F := \exists \boldsymbol{x}(B_1(v, \boldsymbol{x})^v_f \vee \cdots \vee B_k(v, \boldsymbol{x})^v_f)$ and then equivalently,

   $F := \exists \boldsymbol{x}(B_1(v, \boldsymbol{x})^v_f) \vee \cdots \vee \exists \boldsymbol{x}(B_k(v, \boldsymbol{x})^v_f)$

   (b) $F_i := \exists \boldsymbol{x}(B_i(v, \boldsymbol{x})^v_f)$

   (c) Eliminate variables from F_i as follows

      i. $D_i := B_i(v, \boldsymbol{x})^v_f$

      ii. While there is a variable x still in D_i, select a conjunctive term $x = t$ or $t = x$ (such that no variable in t depends on x) in D_i, then $D_i := (D_i)^x_t$.

      iii. $F_i = D_i$ (drop the existential quantifier since there are no variables in D_i).

   (d) $F := F_1 \vee \cdots \vee F_k$.

3. Eliminate variables from G as follows

   (a) $G := \forall v\boldsymbol{x}((B_1(v, \boldsymbol{x}) \vee \cdots \vee B_k(v, \boldsymbol{x})) \rightarrow f = v)$ and then equivalently,

   $G := \forall v\boldsymbol{x}(B_1(v, \boldsymbol{x}) \rightarrow f = v) \wedge \cdots \wedge \forall v\boldsymbol{x}(B_k(v, \boldsymbol{x}) \rightarrow f = v)$

   (b) $G_i := \forall v\boldsymbol{x}(B_i(v, \boldsymbol{x}) \rightarrow f = v)$

   (c) Eliminate variables from G_i as follows

      i. $D_i := B_i(v, \boldsymbol{x}) \rightarrow f = v$

      ii. While there is a variable x still in D_i, select a conjunctive term $x = t$ or $t = x$ (such that no variable in t depends on x) from the body of D_i, then $D_i := (D_i)^x_t$.

iii. $G_i = D_i$ (drop the universal quantifier since there are no variables in D_i).

(d) $G := G_1 \vee \cdots \vee G_k$.

4. $E := F \wedge G$.

The following proposition asserts the correctness of this method. Note that the absence of variables in arguments of uninterpreted functions can be achieved by grounding ASP variables and enforcing that no SMT variable occurs nested inside uninterpreted functions.

Proposition 1 *For any completion of a variable isolated ASPMT program with no variables in arguments of uninterpreted functions, applying variable elimination method repeatedly results in a classically equivalent formula that contains no variables.*

Example 1 continued *Recall the equivalence*

$$
\begin{aligned}
speed(1) = Y \ \leftrightarrow\ \exists XD(\quad & (accel(0)=1 \ \wedge\ speed(0)=X \ \wedge\ duration(0)=D \\
& \quad \wedge\ (Y = X + a \times D)) \\
& \vee (decel(0)=1 \ \wedge\ speed(0)=X \ \wedge\ duration(0)=D \\
& \quad \wedge\ (Y = X - a \times D)) \\
& \vee (speed(0) = Y)\ )
\end{aligned}
$$

Step 2a) turns the implication from left to right into the formula

$$
\begin{aligned}
\exists XD(\quad & (accel(0)=1 \ \wedge\ speed(0)=X \ \wedge\ duration(0)=D \ \wedge \\
& \quad (speed(1)=X + a \times D)) \\
& \vee (decel(0)=1 \ \wedge\ speed(0)=X \ \wedge\ duration(0)=D \wedge \\
& \quad (speed(1)=X - a \times D)) \\
& \vee (speed(0) = speed(1)))
\end{aligned}
$$

And then step 2d) produces

$$(accel(0)=1 \wedge speed(1)=speed(0)+a\times duration(0))\vee$$
$$(decel(0)=1 \wedge speed(1)=speed(0)-a\times duration(0))\vee$$
$$(speed(0)=speed(1)).$$

To see why variable isolation is required, consider the formula $f = X \leftrightarrow 2*X = Y \wedge 2*Y = X$. One step of 3c) produces the formula

$$2*2*Y = Y \wedge X = X \rightarrow f = 2*Y.$$

We can drop $X = X$ and then perform another step of 3c) to get

$$2*2*(2*2*Y) = 2*2*Y \rightarrow f = 2*(2*2*Y).$$

Then, at this point, no conjunctive term exists of the form $x = t$ so the procedure terminates but the variable Y still remains in the formula.

9.2.2 Syntax of Input Language

System ASPMT2SMT imposes three syntactic restrictions on input ASPMT2SMT theories comprised of rules of the form $H \leftarrow B$ where B is a conjunction of possibly negated literals and H is $\bot$ or $f(\boldsymbol{t}) = v$: they must be *variable isolated* (defined in Section 9.2.1), *av-separated*, and $\boldsymbol{f}$*-plain* (defined in Section 5.4). It should also be noted that the only background theories considered in this version of the implementation are arithmetic over reals and integers.

We require that input formulas be $\boldsymbol{c}$-plain. As Theorem 8 indicates, this condition can be relaxed if F is tight. Relaxing this restriction is left for a future version of this system.

We call a variable v in a rule an *argument variable* if it occurs in an argument $\boldsymbol{t}$ of some uninterpreted function $f(\boldsymbol{t})$ in the rule. We call a variable v in a rule a *value variable* if it occurs in

- $f(\boldsymbol{t}) = v$ for any term where f is an uninterpreted function, or
- $t_1 = t_2$ where t_1, t_2 are terms consisting of interpreted symbols (i.e., from σ^{bg}) and at least one other value variable (different from v) in the rule.

A rule is said to be *av-separated* (argument-value separated) if it contains no variable that is both an argument variable and a value variable. This is a stronger condition than the condition described in Section 9.2.1 concerning ASP and SMT variables and will be relaxed in future versions of the system.

Example 22 *In $f(x) = y \leftarrow y = m * m \wedge m = z + 1 \wedge g(x) = z$, x is an argument variable as it occurs as an argument of functions g and f. Both y and z occur as v in some $f(\boldsymbol{t}) = v$ so they are value variables. Since m occurs in $m = z + 1$ and z is a value variable, m is also a value variable. This rule is av-separated since no variable is both an argument variable and a value variable.*

Example 23 *In $f(x) = y \wedge y = x$, y is a value variable and since x appears in $y = x$, x is also a value variable. At the same time, x is also an argument variable. Consequently, this rule is not av-separated.*

Example 24 *To see why this condition is imposed, consider the formula*

$$f(x) = 1 \leftarrow g = y \wedge y = x.$$

The system sets the equality $y = x$ aside and grounds the formula and then replaced the equality $y = x$ to get

$$f(1) = 1 \leftarrow g = y \wedge y = x$$

$$f(2) = 1 \leftarrow g = y \wedge y = x$$

rather than the intended

$$f(1) = 1 \leftarrow g = y \wedge y = 1$$

$$f(2) = 1 \leftarrow g = y \wedge y = 2$$

System ASPMT2SMT uses a syntax similar to system CPLUS2ASP Babb and Lee (2013) for the declarations and a syntax similar to system F2LP Lee and Palla (2009) for the theory itself.

There are declarations of four kinds, sorts, objects, constants, and variables. The sort declarations specify user data types (note: these cannot be used for value sorts). The object declarations specify the elements of the user-declared data types. The constant declarations specify all of the (possibly boolean) function constants that appear in the theory. The variables declarations specify the user-declared data types associated with each variable. A declaration for the car example is shown below.

```
:-sorts
   step;astep.

:-objects
   0..st              ::  step;
   0..st-1            ::  astep.

:-constants
   time(step)         ::  real[0..t];
   duration(astep)    ::  real[0..t];
   accel(astep)       ::  boolean;
```

```
decel(astep)        :: boolean;
speed(step)         :: real[0..ms];
location(step)      :: real[0..l].


:-variables
 S                  :: astep;
 B                  :: boolean.
```

Only propositional connectives are supported in this version of ASPMT2SMT and these are represented in the system as follows:

$\wedge$	$\vee$	$\neg$	$\rightarrow$	$\leftarrow$
&	—	not	$->$	$<-$

Comparison and arithmetic [3] operators are represented as usual:

$<$	$\leq$	$\geq$	$>$	$=$	$\neq$	add	subtract	multiply	divide
$<$	$<=$	$>=$	$>$	$=$	$!=$	$+$	$-$	$*$	$/$

$a\ != b$ is understood as $\neg(a = b)$. To abbreviate the formula $A \vee \neg A$ (or $Choice(A)$), which is useful for expressing defaults and inertia, we write $\{A\}$. The rest of the car example is shown below.

```
% Actions and durations are exogenous
{accel(S)=B}.
{decel(S)=B}.
{duration(S)=X}.
```

[3]Note that the type of division (integer or real) is based on context; for atomic formulas not containing value variables, the division is understood as integer division whereas for atomic formulas containing value variables, the division is instead understood as real division.

```
% nonconcurrency of actions
<- accel(S)=true & decel(S)=true.

%effects of accel and decel
speed(S+1)=Y <- accel(S)=true & speed(S)=X & duration(S)=D
      & Y = X+ar*D.
speed(S+1)=Y <- decel(S)=true & speed(S)=X & duration(S)=D
      & Y = X-ar*D.

% preconditions of accel and decel
<- accel(S)=true & speed(S)=X & duration(S)=D
      & Y = X+ar*D & Y > ms.
<- decel(S)=true & speed(S)=X & duration(S)=D
      & Y = X-ar*D & Y < 0.

% inertia of speed
{speed(S+1)=X} <- speed(S)=X.

location(S+1)=Y <- location(S)=X & speed(S)=A &
      speed(S+1)=C & duration(S)=D & Y = X+(A+C)/2*D.

time(S+1)=Y <- time(S)=X & duration(S)=D & Y=X+D.

% problem instance
```

```
time(0)=0.
speed(0)=0.
location(0)=0.

<- location(st) = Z & Z != l.
<- speed(st) = Z & Z != 0.
<- time(st) = Z & Z != t.
```

This description can be run by the command

```
$aspmt2smt car -c st=3 -c t=4 -c ms=4 -c ar=3 -c l=10
```

which yields the output

```
  accel(0) = true  accel(1) = false  accel(2) = false
  decel(0) = false  decel(1) = false  decel(2) = true
  duration(0) = 1.1835034190  duration(1) = 1.6329931618
  duration(2) = 1.1835034190  location(0) = 0.0
  location(1) = 2.1010205144  location(2) = 7.8989794855
  location(3) = 10.0  speed(0) = 0.0
  speed(1) = 3.5505102572  speed(2) = 3.5505102572
  speed(3) = 0.0  time(0) = 0.0  time(1) = 1.1835034190
  time(2) = 2.8164965809  time(3) = 4.0
z3 time in milliseconds: 30
Total time in milliseconds: 71
```

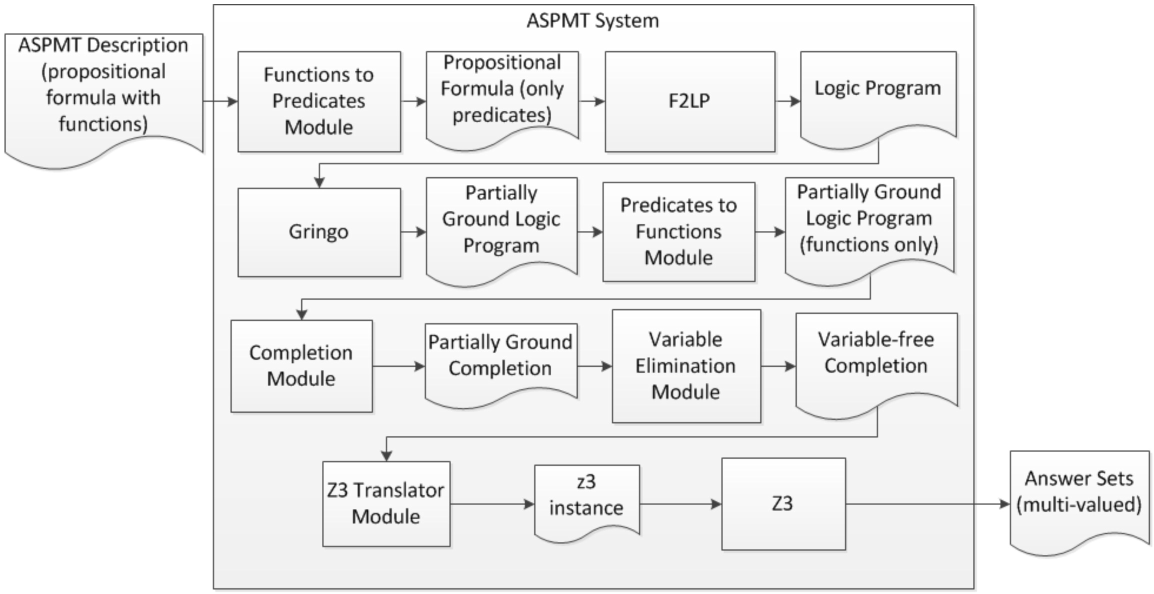

Figure 9.2: ASPMT2SMT System Architecture

9.2.3 Architecture

The architecture of the system is shown in Figure 9.2.3 The ASPMT2SMT system first converts the ASPMT description to a propositional formula containing only predicates. In addition, this step substitutes auxiliary constants for value variables and necessary preprocessing for F2LP and GRINGO to enable partial grounding of argument variables only. F2LP transforms the propositional formula into a logic program and then GRINGO performs partial grounding on only the argument variables. The ASPMT2SMT system then converts the predicates back to functions and replaces the auxiliary constants with the original expressions. Then the system computes the completion of this partially ground logic program and performs variable elimination on that completion. Finally, the system converts this variable-free description into the language of Z3 and then relies on Z3 to produce models which correspond to stable models of the original ASPMT description.

Example 1 continued *Consider the result of variable elimination on the portion of the completion related to* $speed(1)$ *of the running car example:*

$$\begin{aligned}
&(Accel(0)=1 \wedge Speed(1)=Speed(0) + A \times Duration(0))\vee \\
&(Decel(0)=1 \wedge Speed(1) = Speed(0) - A \times Duration(0))\vee \\
&(Speed(0) = Speed(1)).
\end{aligned}$$

In the language of $z3$*, this is*

```
(assert (or (or
    (and (= accel_0_ true) (= speed_1_
              (+ speed_0_ (* duration_0_ a))))
    (and (= decel_0_ true) (= speed_1_
              (- speed_0_ (* duration_0_ a)))))
    (= speed_1_ speed_0_)
))
```

The system is available at `http://reasoning.eas.asu.edu/aspmt/`.

9.2.4 Experiments

The following experiments demonstrate the capability of the ASPMT2SMT system to perform nonmonotonic reasoning about continuous changes. In addition, this shows a significant performance increase compared to ASP systems for domains in which only value variables have large domains. However, when argument variables have large domains, similar scalability issues arise as comparable grounding still occurs.

We also provide a comparison to system CLINGCON which loosely integrates logic programming and constraint satisfaction. While this performs well, these representations are either not elaboration tolerant or require new auxiliary abnormality symbols to represent the notions of inertia and default behaviors. Additionally, this system does not support continuous reasoning.

These experiments were performed on an Intel Core 2 Duo 3.00 GHZ CPU with 4 GB RAM running Ubuntu 13.10.

Leaking Bucket

Consider a leaking bucket with maximum capacity c that loses one unit of water every time step by default. The bucket can be refilled to its maximum capacity by the action $fill$. The initial capacity is 5 and the desired capacity is 10. Here, the argument variable corresponding to the length of the plan increases so both systems suffer scalability issues.

```
:- sorts
   atime;time.


:- objects
   0..c :: step;
   0..c-1  :: astep.


:- constants
   amt(step) :: int[0..c];
```

```
  fill(astep) :: boolean.

:-variables
  T :: step;
  ST :: astep;
  X :: int[0..c].

{amt(ST+1) = X-1} <- amt(ST) = X.
{fill(ST) = true}.
{fill(ST) = false}.

amt(ST+1) = X <- fill(ST) = true & X = c.
<- amt(T) = X & X < 2.

amt(0) = 5.
<- not (amt(c) = 10).
```

c	ASP (CLINGO 4.3.0) Run Time (Grounding + Solving)	ASPMT2SMT 1.0 Run Time (Preprocessing + Solving)	CLINGCON 2.0.3 Run Time (Preprocessing + Solving)
10	0s (0s+0s)	.037s (.027s + .01s)	0s(0s + 0s)
50	.03s (03s + 0s)	.089s (.079s + .01s)	0s(0s + 0s)
100	.15s (.15s + 0s)	.180s (.170s + .01s)	0.1s(0.1s + 0s)
500	3.95s (3.95s + 0s)	1.731s (1.661s + .07s)	0.3s(0.3s + 0s)
1000	19.01s (18.99s+ .02s)	35.326s (35.206s + .12s)	0.6s(0.6s + 0s)

We see that in this experiment, ASPMT2SMT does not yield better results than CLINGO. The reason for this is that the scaling of this domain takes place in the number of timesteps. Thus, since ASPMT2SMT uses GRINGO (the grounder for both CLINGO and ASPMT2SMT) to obtain functions for each of these timesteps, the ground descriptions given to CLASP (the solver in system CLINGO) and Z3 are of similar size. Consequently, we see that the majority of the time taken for ASPMT2SMT is in preprocessing.

Car Example

Recall the domain in Example 1.

The first half of the experiments are done with the values L $= 10k$, A $= 3k$, MS $=$ $4k$, T $= 4k$, which yields solutions with irrational values and so cannot be solved by systems CLINGO and CLINGCON. The second half of the experiments are done with the values L $= 4k$, A $= k$, MS $= 4k$, T $= 4k$, which yields solutions with integral values and so can be solved by systems CLINGO and CLINGCON. In this example, only the value variables have increasing domains but the argument variable domain remains the same. Consequently, the ASPMT2SMT system scales very well compared to the ASP system which can only complete the two smallest size domains.

k	ASP (clingo 4.3.0) Run Time (Grounding + Solving)	ASPMT2SMT 1.0 Run Time (Preprocessing + Solving)	clingcon 2.0.3 Run Time (Preprocessing + Solving)
1	n/a	.084s (.054s + .03s)	n/a
5	n/a	.085s (.055s + .03s)	n/a
10	n/a	.085s (.055s + .03s)	n/a
50	n/a	.087s (.047s + .04s)	n/a
100	n/a	.088s (.048s + .04s)	n/a
1	.22s (.22s + 0s)	.060s (.050s + .01s)	0s(0s + 0s)
2	62.11s (62.10s + .01s)	.07s (.050s + .02s)	0s(0s + 0s)
3	> 30 minutes	.072s (.052s + .02s)	0s(0s + 0s)
5	> 30 minutes	.068s (.048s + .02s)	0s(0s + 0s)
10	> 30 minutes	.068s (.048s + .02s)	0s(0s + 0s)
50	> 30 minutes	.068s (.048s + .02s)	0s(0s + 0s)
100	> 30 minutes	.072s (.052s + .02s)	0s(0s + 0s)

Here, the results for ASPMT2SMT are much more favorable. In this problem, the scaling lies only in the size of the value of the functions involved in the description. Consequently, we see no scaling issues in either ASPMT2SMT or clingcon. Neither clingcon nor clingo is able to handle the first set of configurations since these parameters yield non-integral solutions. On the other hand, ASPMT2SMT handles these configurations with comparable execution time to the performance in the second set of configurations.

Space Shuttle Example

The following example is from Lee and Lifschitz (2003), which represents cumulative effects on continuous changes. A spacecraft is not affected by any external forces. It has two jets and the force that can be applied by each jet along each axis is at most $4k$. The initial position of the rocket is (0,0,0) and its initial velocity is (0,1,1). How can it get to $(0,3k,2k)$ within 2 seconds? Assume the mass is 2.

```
:- sorts
     step;astep;axis.


:- objects
     0..st    :: step;
     0..st-1  :: astep;
     x,y,z    :: axis.


:- constants
     duration(astep)          :: real[0..2];
     time(step)               :: real[0..2];
     mass                     :: real[0..m];
     speed(axis,step)         :: real[0..v];
     pos(axis,step)           :: real[0..p];
     jet1fire(axis,astep)     :: real[0..f];
     jet2fire(axis,astep)     :: real[0..f].


:- variables
   S        :: step;
```

```
  AS       :: astep;
  AX,AX1   :: axis.

mass = m.

speed(x,0) = 0.
speed(y,0) = 1.
speed(z,0) = 1.

time(0) = 0.

pos(x,0) = 0.
pos(y,0) = 0.
pos(z,0) = 0.

{duration(AS) = X}.
{jet1fire(AX,AS) = X}.
{jet2fire(AX,AS) = X}.

<- jet1fire(AX,AS) = X & jet1fire(AX1,AS) = X1 &
     X != 0 & X1 != 0 & AX != AX1.
<- jet2fire(AX,AS) = X & jet2fire(AX1,AS) = X1 &
     X != 0 & X1 != 0 & AX != AX1.

pos(AX,AS+1) = Z <- pos(AX,AS) = X & duration(AS) = T &
```

```
    speed(AX,AS) = S0 & speed(AX,AS+1) = S1 &
            Z = X + T*(S0+S1)/2.
speed(AX,AS+1) = Z <- jet1fire(AX,AS) = X1 &
    jet2fire(AX,AS) = X2 & duration(AS) = T & mass = M &
            speed(AX,AS) = Y & Z = Y+T*(X1+X2)/M.

time(AS+1) = X <- time(AS) = Y & duration(AS) = T &
      X = Y+T.

<- pos(x,st) = X & X != 0.
<- pos(y,st) = X & X != 3*k.
<- pos(z,st) = X & X != 2*k.
```

k	ASP (CLINGO 4.3.0) Run Time (Grounding + Solving)	ASPMT2SMT 1.0 Run Time (Preprocessing + Solving)	CLINGCON 2.0.3 Run Time (Preprocessing + Solving)
1	0.01s (0.01s + 0s)	.048s (.038s + .01s)	0s(0s + 0s)
5	.08s (.06s + .02s)	.047s (.037s + .01s)	0s(0s + 0s)
10	.35s (.24s + .11s)	.053s (.043s + .01s)	0s(0s + 0s)
50	13.40s (6.64s + 6.76s)	.050s (.040s + .01s)	0s(0s + 0s)
100	39.17s (30.71s + 8.46s)	.051s (.041s + .01s)	0s(0s + 0s)

Again in this problem, the scaling lies only in the size of the value of the functions involved in the description. Consequently, we see no scaling issues in either ASPMT2SMT or CLINGCON.

Bouncing Ball Example

The following example is from Chintabathina (2008). Consider an agent acting in a domain consisting of a ball. The ball is held above the ground by the agent. The actions available to the agent are drop and catch. Dropping the ball causes the height of the ball to change continuously with time as defined by Newton's laws of motion. As the ball accelerates towards the ground it gains velocity. If the ball is not caught before it reaches the ground it hits the ground with speed s and bounces up into the air with speed $r * s$ where $r = .95$ is the rebound coefficient. The bouncing ball reaches a certain height and falls back towards the ground due to gravity. A robot is holding a ball at height $100k$. We want to have the ball hit the ground and caught at height 50.

```
:- sorts
   step;astep.


:- objects
   0..st     :: step;
   0..st-1   :: astep.


:- constants
   pos(step)      :: real[0..p];
   speed(step)    :: real[-5000..5000];
   drop(astep)    :: boolean;
   catch(astep)   :: boolean;
   duration(astep) :: real[0..1000];
   gravity        :: real[-50..50];
```

```
   coefficient :: real[0..1];
   holding(step) :: boolean.

:- variables
  S   :: step;
  AS :: astep.

coefficient = 95/100.
gravity = -98/10.

pos(0) = p.
holding(0) = true.
speed(0)  = 0.

{duration(AS) = X}.
{drop(AS) = true}.
{drop(AS) = false}.
{catch(AS) = true}.
{catch(AS) = false}.

<- drop(AS) = true & catch(AS) = true.
<- drop(AS) = true & holding(AS) = false.
<- catch(AS) = true & holding(AS) = true.

<- drop(AS) = true & duration(AS) = X & X != 0.
```

```
<- catch(AS) = true & duration(AS) = X & X != 0.

holding(AS+1) = true <- catch(AS) = true.
speed(AS+1) = 0 <- catch(AS) = true.
holding(AS+1) = false <- drop(AS) = true.

holding(AS+1) = true <- holding(AS) = true &
     drop(AS) = false.
holding(AS+1) = false <- holding(AS) = false &
     catch(AS) = false.

{speed(AS+1) = X} <- speed(AS) = Y & duration(AS) = T &
     gravity = G & X = Y+T*G & holding(AS) = false.
speed(AS+1) = X <- speed(AS) = X & holding(AS) = true.
speed(AS+1) = X <- speed(AS) = Y & coefficient = C &
     X = -1*Y*C & pos(AS) = 0 & holding(AS) = false.

<- pos(S) = X & X < 0.

pos(AS+1) = X <- pos(AS) = Y & duration(AS) = T &
     speed(AS+1) = S2 & speed(AS) = S1 &
                    X = Y + T*(S1+S2)/2 & ((catch(AS) = false &
                    holding(AS) = false) | drop(AS) = true).
pos(AS+1) = X <- pos(AS) = X &
     ((holding(AS) = true & drop(AS) = false) |
```

```
                    catch(AS) = true).

<- pos(st-2) = X & X != 0.
<- pos(st) = X & X != 50.
```

k	ASP (CLINGO 4.3.0) Run Time (Grounding + Solving)	ASPMT2SMT 1.0 Run Time (Preprocessing + Solving)	CLINGCON 2.0.3 Run Time (Preprocessing + Solving)
1	n/a	.072s (.062s + .01s)	n/a
10	n/a	.072s (.062s + .01s)	n/a
100	n/a	.071s (.061s + .01s)	n/a
1000	n/a	.075s (.065s + .01s)	n/a
10000	n/a	.082s (.062s + .02s)	n/a

Again, CLINGO and CLINGCON are unable to find solutions to this domain since solutions are not integral. Also, we see that ASPMT2SMT suffers no scaling issues here again due to the fact that in this problem the scaling lies only in the size of the value of the functions involved in the description.

9.3 Proofs

9.3.1 *Proof of Proposition 1*

The proof relies on the following lemmas.

Lemma 29

$$\forall v\boldsymbol{x}(t(\boldsymbol{x}) = v \wedge H(v\boldsymbol{x}) \rightarrow G(v\boldsymbol{x}))$$

is equivalent to

$$\forall \boldsymbol{x}(H(v\boldsymbol{x}) \rightarrow G(v\boldsymbol{x}))^{v}_{t(\boldsymbol{x})}.$$

Proof. Given an interpretation I

$I \models \forall v\boldsymbol{x}(t(\boldsymbol{x}) = v \wedge H(v\boldsymbol{x}) \rightarrow G(v\boldsymbol{x}))$ iff

$I \models t(\vec{\xi}) = \xi \wedge H(\xi\vec{\xi}) \rightarrow G(\xi\vec{\xi})$ for every $\xi\vec{\xi}$ from $|I|$ iff

$I \models H(\xi\vec{\xi}) \rightarrow G(\xi\vec{\xi})$ where $\xi = t(\vec{\xi})^I$ for every $\xi\vec{\xi}$ from $|I|$, (when $\xi \neq t(\vec{\xi})^I$, the implication is trivially satisfied) iff

$I \models \forall \boldsymbol{x}(H(t(\boldsymbol{x})\boldsymbol{x}) \rightarrow G(t(\boldsymbol{x})\boldsymbol{x}))$ iff

$I \models \forall \boldsymbol{x}(H(v\boldsymbol{x}) \rightarrow G(v\boldsymbol{x}))^{v}_{t(\boldsymbol{x})}$. ∎

Lemma 30

$$\exists z\boldsymbol{x}(D(z\boldsymbol{x}) \wedge z = t(\boldsymbol{x}))$$

is equivalent to

$$\exists \boldsymbol{x}(D(z\boldsymbol{x})^{z}_{t(\boldsymbol{x})}).$$

Proof. Given an interpretation I

$I \models \exists z\boldsymbol{x}(D(z\boldsymbol{x}) \wedge z = t)$ iff

$I \models D(\xi\vec{\xi}) \wedge \xi = t(\vec{\xi})$ for some $\xi\vec{\xi}$ from $|I|$ iff

$I \models D(\xi\vec{\xi})$ where $\xi = t(\vec{\xi})^I$ for some $\xi\vec{\xi}$ from $|I|$, (if $t(\vec{\xi})^I \neq \xi$, then clearly $I \not\models$

$D(\xi\vec{\xi}) \wedge \xi = t(\vec{\xi}))$ iff

$I \models \exists \boldsymbol{x}(D(t(\boldsymbol{x})\boldsymbol{x}))$ iff

$I \models \exists \boldsymbol{x}(D(z\boldsymbol{x})^z_{t(\boldsymbol{x})})$ ∎

Lemma 31 *Consider a conjunction of possibly negated atomic formulas $C_1 \wedge \cdots \wedge C_n$ such that for every variable v occurring in the conjunction, there is some C_i such that C_i is $v = t$ or $t = v$ for some term t such that within $C_1 \wedge \cdots \wedge C_n$, no variable in t depends on v. Given such a v and t, $(C_1 \wedge \cdots \wedge C_n)^v_t$, for every variable u in $(C_1 \wedge \cdots \wedge C_n)^v_t$, there is some C_i such that C_i is $u = t'$ or $t' = u$ for some term t' such that within $C_1 \wedge \cdots \wedge C_n$, no variable in t' depends on u.*

Proof. Consider any variables v and u in $C_1 \wedge \cdots \wedge C_n$. We start with the fact that there is some C_i that is $v = t_1$ or $t_1 = v$ for some term t_1 such that within $C_1 \wedge \cdots \wedge C_n$, no variable in t_1 depends on v and that there is some C_j that is $u = t_2$ or $t_2 = u$ for some term t_2 such that within $C_1 \wedge \cdots \wedge C_n$, no variable in t_2 depends on u. Now we consider the effect of replacing v with t_1 in $C_1 \wedge \cdots \wedge C_n$. This yields $(C_1 \wedge \cdots \wedge C_n)^v_{t_1}$. There are two possibilities for C_j.

- C_j does not contain v. Then $(C_j)^v_{t_1}$ is exactly C_j and so this still satisfies that no variable in t_2 depends on u.

- C_j does contain v. Then $(C_j)^v_{t_1}$ is $u = (t_2)^v_{t_1}$ or $(t_2)^v_{t_1} = u$ we must check that $(t_2)^v_{t_1}$ does not contain any variable that depends on u. However, in this case u depends on v and since we assumed that t_2 contained no variable that depends on u, we know v does not depend on u. Consequently, no variable in t_1 depends on u and so we conclude that $(t_2)^v_{t_1}$ does not contain any variable that depends on u.

Proposition 1 *For any completion of an av-separated, variable isolated ASPMT program, applying variable elimination method repeatedly results in a classically equivalent formula that contains no variables.*

Proof.

Consider the completion of an av-separated, variable isolated ASPMT program, which is a conjunction of equivalences of the form

$$E = \forall v(f = v \leftrightarrow \exists \boldsymbol{x}(B_1(\boldsymbol{x}) \vee \cdots \vee B_k(\boldsymbol{x})))$$

where each $B_i(\boldsymbol{x})$ is a conjunction of possibly negated atomic formulas and has $v = t$ or $t = v$ as a non-negated subformula for some term t such that within $B_i(\boldsymbol{x})$, no variable in t depends on v. The proof is by induction on each equivalence E and n, the number of variables in the E.

- E contains no variables. The variable elimination leaves E unchanged and since there are no variables in F, the claim holds.

- E is

  $$\forall v(f = v \leftrightarrow \exists \boldsymbol{x}(B_1(\boldsymbol{x}) \vee \cdots \vee B_k(\boldsymbol{x})))$$

  where each $B_i(\boldsymbol{x})$ is a conjunction of possibly negated atomic formulas and has $v = t$ or $t = v$ as a non-negated subformula for some term t such that within $B_i(\boldsymbol{x})$, no variable in t depends on v. Step 1 produces two formulas F and G where

  $F := \forall v(f = v \rightarrow \exists \boldsymbol{x}(B_1(\boldsymbol{x}) \vee \cdots \vee B_k(\boldsymbol{x})))$

  $G := \forall v(\exists \boldsymbol{x}(B_1(\boldsymbol{x}) \vee \cdots \vee B_k(\boldsymbol{x})) \rightarrow f = v)$ Clearly E is equivalent to $F \wedge G$ so the claim follows by induction on F and G.

- E is $\forall v(f = v \to \exists \boldsymbol{x}(B_1(\boldsymbol{x}) \vee \cdots \vee B_k(\boldsymbol{x})))$ where each $B_i(\boldsymbol{x})$ is a conjunction of possibly negated atomic formulas and has $v = t$ or $t = v$ as a non-negated subformula for some term t such that within $B_i(\boldsymbol{x})$, no variable in t depends on v. Step 2(a) of the variable elimination method produces the formula $F := \exists \boldsymbol{x}(B_1(\boldsymbol{x})^v_f) \vee \cdots \vee \exists \boldsymbol{x}(B_k(\boldsymbol{x})^v_f)$. F is equivalent to E by Lemma 29. F does not contain the variable v since v is replaced by f. The claim follows by induction on each $\exists \boldsymbol{x}(B_i(\boldsymbol{x})^v_f)$.

- E is $\exists y\boldsymbol{x}(B(y\boldsymbol{x}))$ where $B(y\boldsymbol{x})$ is a conjunction of possibly negated atomic formulas and has $y = t$ or $t = y$ as a non-negated subformula for some term t such that within $B(y\boldsymbol{x})$, no variable in t depends on y. One iteration of step 2(c) will produce the formula $F := \exists \boldsymbol{x}(B(\boldsymbol{x})^y_t)$. F is equivalent to E by Lemma 30. F does not contain the variable y since y is replaced by t and no variable in t depends on y. Further, by Lemma 31 F has the property that for every variable z in F, F has $z = t'$ or $t' = z$ as a non-negated subformula for some term t' such that within $B(\boldsymbol{x})^y_t$, no variable in t' depends on z. So, the claim follows by induction on F.

- E is $\forall y(\exists \boldsymbol{x}(B_1(\boldsymbol{x}) \vee \cdots \vee B_k(\boldsymbol{x})) \to f = v)$ (for $k \geq 2$). Step 3(a) will produce the formula $F := \forall v\boldsymbol{x}(B_1(\boldsymbol{x}) \to f = v) \wedge \cdots \wedge \forall v\boldsymbol{x}(B_k(\boldsymbol{x}) \to f = v)$. E is classically equivalent to F so the claim holds by induction on each $\forall v\boldsymbol{x}(B_i(\boldsymbol{x}) \to f = v)$.

- E is $\forall y\boldsymbol{x}(B(\boldsymbol{x}) \to f = v)$ (y and v may be the same) where $B(y\boldsymbol{x})$ is a conjunction of possibly negated atomic formulas and has $y = t$ or $t = y$ as a non-negated subformula for some term t such that within $B(y\boldsymbol{x})$, no variable in t depends on y. One iteration of step 3(c) produces the formula $F := \forall \boldsymbol{x}(B(\boldsymbol{x}) \to f = u)^y_t$.

F is equivalent to E by Lemma 29. F does not contain the variable y since y is replaced by t and no variable in t depends on y. Further, by Lemma 31 F has the property that for every variable z in F, F has $z = t'$ or $t' = z$ as a non-negated subformula for some term t' such that within $B(\boldsymbol{x})^y_t$, no variable in t' depends on z. So, the claim follows by induction on F.

Chapter 10

CABALAR SEMANTICS

10.1 Reduct Characterization

The Cabalar semantics reviewed in Chapter 3 can also be reformulated in terms of grounding and reduct. A theorem similar to Theorem 1 can be stated for the Cabalar semantics.

Theorem 21 *Let F be a first-order sentence of signature σ and let $\mathbf{c}$ be a list of intensional constants. For any partial interpretation I of σ, $\langle I, I\rangle$ is a partial equilibrium model of F iff*

- *$I \models_p F$, and*
- *for every partial interpretation J of σ such that $J \prec^{\mathbf{c}} I$, we have $J \not\models_p gr_I[F]^{\underline{I}}$.*

Example 3 continued *Recall the example that describes the inertia of the location of a box. The reduct F^{I_1} is*

$$\begin{array}{lcccr} & & at(box,0,l1)\wedge & & \\ ((at(box,1,l1) & \vee & \bot) & \leftarrow & at(box,0,l1))\wedge \\ ((\bot & \vee & \neg\bot) & \leftarrow & \bot) \end{array}$$

Recall the three partial interpretations J_1, J_2, J_3 that satisfy $J_i \prec^{at} I_1$ which agree with I_1 except that

- *$at(box,0,l1)^{J_1} = u$;*

- *$at(box, 1, l1)^{J_2} = u$;*

- *$at(box, 0, l1)^{J_3} = u$ and $at(box, 1, l1)^{J_3} = u$.*

Now it is easy to see that J_1 and J_3 fail to satisfy the first conjunction of the reduct while J_2 fails to satisfy the second conjunction of the reduct. Thus, this characterization corresponds to the equilibrium logic style definition for this case.

On the other hand, the reduct F^{I_2} is

$$
\begin{array}{lccccc}
 & & at(box,0,l1)\wedge & & & \\
((\bot & \vee & \neg\bot) & \leftarrow & at(box,0,l1))\wedge \\
((at(box,1,l2) & \vee & \bot) & \leftarrow & \bot)
\end{array}
$$

Consider again the partial interpretation J_4 that agrees with I_2 except that $at(box, 1, l2)^{J_4} = u$. We can see that J_4 satisfies this reduct. Thus, this characterization corresponds to the equilibrium logic style for this case.

Interestingly, this reformulation of the Cabalar semantics is closely related to the language ASP$\{f\}$ Balduccini (2012). We discuss the details in Section 11.2.

Comparing the reformulation of the Cabalar semantics in Theorem 21 and the reformulation of the functional stable model semantics semantics in Theorem 1 tells us that the reducts are defined in the same way, whereas interpretations we consider for stability checking and the notions of satisfaction are different. That is, if the intensional constants are function constants only, under the functional stable model semantics, the interpretations J we consider for stability checking are all other classical interpretations that are different from I, while under the Cabalar semantics, they are partial interpretations that are "smaller" than I. For instance, in Example 7, there are many such Js that are different from I_1 for the functional stable model semantics semantics depending on the size of the universe, while there are only 3 such J for the Cabalar semantics.

Later in this chapter, we present some syntactic classes of formulas on which the two semantics coincide despite these differences.

10.2 Second-Order Logic Characterization

The Cabalar semantics can also be formulated in the style of second-order logic. We extend the formulas to allow predicate and function variables as in the standard second-order logic, but consider partial interpretations in place of classical interpretations. Similar to the definition of $\widehat{\boldsymbol{c}} < \boldsymbol{c}$, we define $\widehat{\boldsymbol{c}} \preceq \boldsymbol{c}$ as

$$(\widehat{\boldsymbol{c}}^{pred} \leq \boldsymbol{c}^{pred}) \wedge (\widehat{\boldsymbol{c}}^{func} \leq \boldsymbol{c}^{func}) \ ,$$

where $\widehat{\boldsymbol{c}}^{pred} \leq \boldsymbol{c}^{pred}$ is as defined in Section 4.2–the conjunction of $\forall \boldsymbol{x}(\widehat{p}(\boldsymbol{x}) \rightarrow p(\boldsymbol{x}))$ for each predicate constant $p \in \boldsymbol{c}^{pred}$ and the corresponding predicate variable $\widehat{p} \in \widehat{\boldsymbol{c}}^{pred}$. $\widehat{\boldsymbol{c}}^{func} \leq \boldsymbol{c}^{func}$ is defined as the conjunction of

$$\forall \boldsymbol{x}((\widehat{f}(\boldsymbol{x}) \neq \widehat{f}(\boldsymbol{x})) \vee (\widehat{f}(\boldsymbol{x}) = f(\boldsymbol{x}))) \ .$$

for all function constants f in $\boldsymbol{c}^{func}$ and the corresponding function variables $\widehat{f}$ in $\widehat{\boldsymbol{c}}^{func}$. As explained earlier, the first disjunctive term is satisfiable under a partial interpretation, meaning that $\widehat{f}$ is undefined on $\boldsymbol{x}$; the second disjunctive term means that $\widehat{f}$ and f are both defined on $\boldsymbol{x}$ and map to the same element in the universe. We define $\widehat{\boldsymbol{c}} \prec \boldsymbol{c}$ as $(\widehat{\boldsymbol{c}} \preceq \boldsymbol{c}) \wedge \neg(\boldsymbol{c} \preceq \widehat{\boldsymbol{c}})$.

We reformulate the Cabalar semantics by using the expression CBL that looks similar to SM. It is defined as:

$$\mathrm{CBL}[F; \boldsymbol{c}] = F \wedge \neg \exists \widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} \prec \boldsymbol{c} \wedge F^{\dagger}(\widehat{\boldsymbol{c}})) \ ,$$

where $F^{\dagger}(\widehat{\boldsymbol{c}})$ is defined the same as $F^{*}(\widehat{\boldsymbol{c}})$ in Section 4.2 except for the base case:

- When F is an atomic formula, $F^{\dagger}(\widehat{\boldsymbol{c}})$ is $F(\widehat{\boldsymbol{c}})$ where $F(\widehat{\boldsymbol{c}})$ is the result of replacing each occurrence of each constant $c \in \boldsymbol{c}$ with the corresponding variable $\widehat{c} \in \widehat{\boldsymbol{c}}$. [1]

The following theorem states the correctness of the reformulation.

Theorem 22 *For any sentence F, a PHT-interpretation $\langle I, I\rangle$ is a partial equilibrium model of F relative to $\boldsymbol{c}$ iff $I \models_p CBL[F; \boldsymbol{c}]$.*

Note the similarity between this reformulation of the Cabalar semantics given in Theorem 22 and the definition of SM in Section 4.2. The differences are in the comparison operators $\prec$ vs. $<$, and whether to consider partial interpretations or classical interpretations.

Neither semantics is stronger than the other. The following example presents a formula that has a stable model under the Cabalar semantics, but not under the functional stable model semantics.

Example 25 *$SM[f = g;\ f, g]$ has no models if the universe contains more than one element. Take any I such that $I \models f = g$. The reduct of $f = g$ relative to I is $f = g$ itself, and there are other models of the reduct. Since I is not the unique model of the reduct, I is not a (f, g)-stable model of $f = g$. On the other hand, assuming that the universe is $\{1, 2, 3\}$, an interpretation I that assigns 1 to both f and g satisfies $CBL[f = g;\ f, g]$. The reduct is the same as before, but any interpretation J smaller than I maps either or both f and g to u, and hence does not satisfy the reduct. Similarly, there are two other models of $CBL[f = g;\ f, g]$ with the same universe.*

[1] In fact, $F^{*}(\widehat{\boldsymbol{c}})$ can be also used in place of $F^{\dagger}(\widehat{\boldsymbol{c}})$ for defining CBL$[F; \boldsymbol{c}]$ as well, without affecting the models.

On the other hand, in the following example, the formula has a stable model under the functional stable model semantics, but not under the Cabalar semantics.

Example 26 *Let F be the formula $f(1) = 1 \wedge f(2) = 1 \wedge (f(g) = 1 \rightarrow g = 1)$, and I be an interpretation such that the universe is $\{1, 2\}$, and $1^I = 1$, $2^I = 2$, $f(1)^I = 1$, $f(2)^I = 1$, $g^I = 1$. One can check that I is a model of SM$[F;\ f, g]$, but not a model of CBL$[F;\ f, g]$.*

10.3 Correspondence on Multi-valued Propositional Formulas

We first present the simpler relationship between the two semantics in the context of multi-valued propositional formulas.

Similar to Theorem 14, the following theorem tells us that the partial stable models of a multi-valued propositional formula can be identified with the stable models of a propositional formula.

Recall that given a multi-valued signature σ, by UC_σ ("Uniqueness Constraint") we denote the conjunction of

$$\bigwedge_{v \neq w \ \mid \ v, w \in Dom(c)} \neg(c = v \wedge c = w) \tag{10.1}$$

for all $c \in \sigma$. The only difference between the two transformations is that we only impose UC_σ and omit EC_σ (the existence constraint).

Theorem 23 *Let F be a multi-valued propositional formula of signature σ, which can be also viewed as a propositional formula of signature σ^{prop}.*

(a) *If a partial interpretation I of σ is a partial multi-valued stable model of F, then I can be viewed as an interpretation of σ^{prop} that is a propositional stable model of $F \wedge UC_\sigma$ (in the sense of Ferraris (2005)).*

(b) *If an interpretation I of σ^{prop} is a propositional stable model of $F \wedge UC_\sigma$ (in the sense of Ferraris (2005)), then I can be viewed as a partial interpretation of σ that is a partial multi-valued stable model of F.*

The following corollary immediately follows from Theorems 14 and 23. It tells us that the stable model semantics can be fully embedded into the partial multi-valued stable model semantics.

Corollary 6 *For any multi-valued propositional formula F of signature σ and any partial interpretation I, we have that I is a multi-valued stable model of F iff I is a partial multi-valued stable model of $F \wedge EC_\sigma$.*

Let σ be a multi-valued signature, and let σ^{none} be the signature that is the same as σ except that the domain of each constant has an additional new value NONE. Given a partial multi-valued interpretation I of σ, by I^{none} we denote a multi-valued interpretation of σ^{none} that agrees with I on all defined constants, and maps undefined constants to NONE.

Theorem 24 *Let F be a multi-valued propositional formula of signature σ.*

(a) *If an interpretation I of σ is a partial multi-valued stable model of F, then I^{none} is a multi-valued stable model of $F \wedge \bigwedge_{c\in\sigma} \Big(c = \text{NONE} \vee \neg(c = \text{NONE})\Big)$.*

(b) *If an interpretation J of σ^{none} is a stable model of $F \wedge \bigwedge_{c\in\sigma} \Big(c = \text{NONE} \vee \neg(c = \text{NONE})\Big)$ then $J = I^{none}$ for some partial multi-valued stable model I of F.*

10.4 Correspondence on $\boldsymbol{f}$-plain Sentences

This section presents the correspondence on $\boldsymbol{f}$-plain sentences between the functional stable model semantics and the Cabalar semantics coincide when we consider "total" interpretations only. Recall that a partial interpretation I is called *total* if I does not map any function constant to u. Obviously, a total interpretation can be identified with the classical interpretation.

Recall that for any function constant f, a first-order formula F is called *f-plain* if each atomic formula in F

- does not contain f, or
- is of the form $f(\boldsymbol{t}) = t_1$ where $\boldsymbol{t}$ is a list of terms not containing f, and t_1 is a term not containing f,

and for a list $\boldsymbol{c}$ of predicate and function constants, we say that F is *$\boldsymbol{c}$-plain* if F is f-plain for each function constant f in $\boldsymbol{c}$.

The following theorem states that the two semantics coincide on $\boldsymbol{c}$-plain formulas.

Theorem 25 *For any $\boldsymbol{c}$-plain sentence F of signature σ, any list $\mathbf{c}$ of intensional constants, and any total interpretation I of σ satisfying $\exists xy(x \neq y)$, $I \models SM[F; \boldsymbol{c}]$ iff $I \models_p CBL[F; \boldsymbol{c}]$.*

Examples 25 and 26 above demonstrate why the restriction to $\boldsymbol{c}$-plain formulas is necessary in Theorem 25.

The requirement in Theorem 25 that every occurrence of every atomic formula be $\boldsymbol{c}$-plain can be relaxed if the formula is tight and in Clark Normal Form. [2]

Theorem 26 *For any sentence F of signature σ in Clark Normal Form that is tight on $\boldsymbol{c}$, and any total interpretation I of σ satisfying $\exists xy(x \neq y)$, $I \models SM[F; \boldsymbol{c}]$ iff $I \models_p CBL[F; \boldsymbol{c}]$.*

10.4.1 Correspondence on non-$\boldsymbol{f}$-plain Sentences

Theorem 25 can be extended to non-$\boldsymbol{c}$-plain formulas by first unfolding F using the same process presented in Section 7.4 that we review here:

- If F is of the form $p(t_1, \dots, t_n)$ $(n \geq 0)$ such that $t_{k_1}, \dots, t_{k_j}$ are all the terms in $t_1, \dots, t_n$ that contain some members of $\boldsymbol{c}$, then $UF_{\boldsymbol{c}}(p(t_1, \dots, t_n))$ is

$$\exists x_1 \dots x_j \Big(p(t_1, \dots, t_n)'' \wedge \bigwedge_{1 \leq i \leq j} UF_{\boldsymbol{c}}(t_{k_i} = x_i)\Big)$$

  where $p(t_1, \dots, t_n)''$ is obtained from $p(t_1, \dots, t_n)$ by replacing each t_{k_i} with the variable x_i.

- If F is of the form $f(t_1, \dots, t_n) = t_0$ $(n \geq 0)$ such that $t_{k_1}, \dots, t_{k_j}$ are all the terms in $t_0, \dots, t_n$ that contain some members of $\boldsymbol{c}$, then $UF_{\boldsymbol{c}}(f(t_1, \dots, t_n) = t_0)$ is

$$\exists x_1 \dots x_j \Big((f(t_1, \dots, t_n) = t_0)'' \wedge \bigwedge_{0 \leq i \leq j} UF_{\boldsymbol{c}}(t_{k_i} = x_i)\Big)$$

  where $(f(t_1, \dots, t_n) = t_0)''$ is obtained from $f(t_1, \dots, t_n) = t_0$ by replacing each t_{k_i} with the variable x_i.

[2] Recall these definitions from Section 5.5.

- $UF_{\boldsymbol{c}}(F \odot G)$ is $UF_{\boldsymbol{c}}(F) \odot UF_{\boldsymbol{c}}(G)$ where $\odot \in \{\wedge, \vee, \rightarrow\}$.

- $UF_{\boldsymbol{c}}(QxF)$ is $Qx\ UF_{\boldsymbol{c}}(F(x))$ where $Q \in \{\forall, \exists\}$.

Recall that $UF_{\boldsymbol{c}}(F)$ is equivalent to F under classical logic. Similarly, Theorem 27 below shows that the Cabalar semantics preserves stable models when unfolding is applied. However, this is not the case under the functional stable model semantics.

Theorem 27 *For any sentence F, any list $\boldsymbol{c}$ of constants, and any partial interpretation I, we have $I \models_p CBL[F; \boldsymbol{c}]$ iff $I \models_p CBL[UF_{\boldsymbol{c}}(F); \boldsymbol{c}]$.*

This theorem generalizes Theorem 1 in Cabalar (2011) that turns programs with functions to programs without functions using a notion similar to unfolding.

Example 27 *Let F be $f = g$. Recall that $UF_{\boldsymbol{c}}(F)$ is $\exists xy(x = y \wedge f = x \wedge g = y)$. Let I_1, I_2, I_3 be interpretations whose universe is $\{1, 2, 3\}$, and each I_i maps f and g to i ($1 \leq i \leq 3$). Each of them satisfies $CBL[F;\ f, g]$ and $CBL[UF_{(f,g)}(F);\ f, g]$, but as we observed, none of them is a model of $SM[F;\ f, g]$.*

However, since $UF_{\boldsymbol{c}}(F)$ is $\boldsymbol{c}$-plain, the following corollary follows from Theorems 25 and 27.

Corollary 7 *For any sentence F, any list $\boldsymbol{c}$ of constants, and any total interpretation I satisfying $\exists xy(x \neq y)$, we have $I \models_p CBL[F; \boldsymbol{c}]$ iff $I \models_p CBL[UF_{\boldsymbol{c}}(F); \boldsymbol{c}]$ iff $I \models SM[UF_{\boldsymbol{c}}(F); \boldsymbol{c}]$.*

For example, $\text{SM}[UF_{(f,g)}(f = g);\ f, g]$ has the same models as $\text{CBL}[f = g;\ f, g]$.

These theorems have established several relationships between the two semantics for total interpretations but in the next sections we consider partial interpretations that may map functions to u.

Let F be a first-order sentence of signature σ. F^{none} is the formula of signature $\sigma \cup \{\text{NONE}\}$ (where NONE is a new object constant) that is obtained from F as follows.

- for any atomic formula F, $F^{none} = F$;
- $(G \odot H)^{none} = (G^{none} \odot H^{none})$ where $\odot \in \{\wedge, \vee, \rightarrow\}$;
- $\forall x G(x)^{none}$ is $\forall x(x \neq \text{NONE} \rightarrow G(x)^{none})$;
- $\exists x G(x)^{none}$ is $\exists x(G(x)^{none} \wedge x \neq \text{NONE})$.

Given a partial interpretation I, we define the total interpretation I^{none} as

- $|I^{none}| = |I| \cup \{\text{NONE}\}$;
- $\text{NONE}^{I^{none}} = \text{NONE}$;
- for every function constant $f \in \sigma$ and $\boldsymbol{\xi} \in |I^{none}|^n$ where n is the arity of f,

$$f^{I^{none}}(\boldsymbol{\xi}) = \begin{cases} f^I(\boldsymbol{\xi}) & \text{if } \boldsymbol{\xi} \text{ is in } |I|^n \text{ and } f^I(\boldsymbol{\xi}) \text{ is defined;} \\ \text{NONE} & \text{otherwise;} \end{cases}$$

- For every predicate $p \in \sigma$ and $\boldsymbol{\xi} \in |I^{none}|^n$ where n is the arity of p,

$$p^{I^{none}}(\boldsymbol{\xi}) = \begin{cases} p^I(\boldsymbol{\xi}) & \text{if } \boldsymbol{\xi} \text{ is in } |I|^n; \\ \boldsymbol{f} & \text{otherwise.} \end{cases}$$

Theorem 28 *For any sentence F of signature σ,* *(a) If I is a Cabalar stable model of F relative to $\mathbf{c}$, then I^{none} is a stable model of*

$$(UF_\sigma(F))^{none} \wedge \bigwedge_{f\in\mathbf{c}} \forall\boldsymbol{x}\langle f(\boldsymbol{x}) = \textsc{none}\rangle \tag{10.2}$$

relative to $\mathbf{c}$.

(b) If an interpretation J such that $\textsc{none}^J = \textsc{none}$ is a stable model of (10.2) *relative to $\mathbf{c}$, then $J = I^{none}$ for some Cabalar stable model I of F relative to $\mathbf{c}$.*

Example 28 *Let F be $f = f$, and let $\mathbf{c}$ be f. Assuming that the universe is* $\{1, 2\}$, *F has two Cabalar stable models: $\{f = 1\}$ and $\{f = 2\}$. The translation* (10.2) *yields the formula*

$$\exists x(f = x \wedge x \neq \textsc{none}) \wedge \{f = \textsc{none}\},$$

and, in accordance with Theorem 28, its stable models are the same as the Cabalar stable models.

For $\neg F$, set $\emptyset$ is the only Cabalar stable model. Accordingly, $(UF_\sigma(\neg F))^{none} \wedge \{f = \textsc{none}\}$ has only one stable model which maps f to NONE.

Theorem 28 becomes incorrect if we do not apply unfolding, i.e., if we replace $UF_\sigma(F)$ in the statement with F. Indeed, for formula $f = f$ above, the modification of (10.2) *yields $f = f \wedge \langle f = \textsc{none}\rangle$, which has $\{f = \textsc{none}\}$ as the only stable model.*

Also, Theorem 28 becomes incorrect if the unfolding is restricted to $\mathbf{c}$ only rather than to the whole σ, i.e., if we replace $UF_\sigma(F)$ with $UF_{\mathbf{c}}(F)$. Indeed, consider F to be $a = b$ where neither a nor b is intensional (i.e., $a, b \notin \mathbf{c}$). Formula (10.2) *is still $a = b$. $I = \emptyset$ is not a Cabalar stable model of $a = b$ relative to $\emptyset$, but $I^{none} = \{a = \textsc{none}, b = \textsc{none}\}$ is a stable model of $a = b$ relative to $\emptyset$.*

To see why we need the condition that $\textsc{none}^J = \textsc{none}$ in part (b), consider the following example.

Example 29 *Consider the formula* $\top$ *with signature* $\sigma = \{c\}$ *and the universe* $\{1\}$*. The only Cabalar stable model* I *is undefined on* c*. On the other hand, formula* (10.2) *yields* $f = \text{NONE} \vee f \neq \text{NONE}$*. Here, without the condition, we have a stable model* J *such that* $\text{NONE}^J = 1$ *and* $f^J = 1$ *but this does not correspond to the Cabalar stable model.*

10.6 Capturing FSM in the Cabalar Semantics

Theorem 25 tells us that for any $\boldsymbol{c}$-plain sentence F, the complete Cabalar stable models of F are precisely the stable models of F. The following corollary shows that the restriction to complete interpretations can instead be expressed in the sentence itself.

Corollary 8 *For any* $\boldsymbol{c}$*-plain sentence* F *and any partial interpretation* I *that satisfies* $\exists xy(x \neq y)$*,* I *is a stable model of* F *relative to* $\mathbf{c}$ *iff* I *is a Cabalar stable model of* $F \wedge \neg\neg \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x}(f(\boldsymbol{x}) = f(\boldsymbol{x}))$ *relative to* $\mathbf{c}$*.*

However, the restriction that the sentence is $\boldsymbol{c}$-plain remains. We consider two examples of non-$\boldsymbol{c}$-plain sentences below.

Example 30 *Consider the very simple problem of restricting the function* f *to a certain domain. To express that* f *is a member of* dom_1 *with the universe* $\{1, 2, 3\}$*, we can simply write* $dom_1(f)$ *where* $\mathbf{c} = \{f\}$ *(*dom_1 *is non-intensional) which alone has no stable models as long as* dom_1 *has more than one element. However, this has among its Cabalar stable models* $\{dom_1(1), dom_1(2), f = 1\}$ *and* $\{dom_1(1), dom_1(2), f = 2\}$*.*

We can try writing this as a constraint $\neg\neg dom_1(f)$ *and no longer are there any Cabalar stable models. However, this does not work in general.*

Example 31 *Consider the extension to the previous example in which we know that* f *belongs to two different domains. To express that* f *is a member of* dom_1 *and a member of* dom_2 *with universe* $\{1, 2, 3\}$*, we can simply write* $dom_1(f) \wedge dom_2(f)$ *where* $\mathbf{c} = \{f\}$ *(*dom_1 *and* dom_2 *are non-intensional) which has a stable model in the case that the intersection of* dom_1 *and* dom_2 *is of size 1; e.g.* $\{dom_1(1), dom_1(2), dom_2(2), dom_2(3), f = 2\}$ *is a stable model. Now the approach to capture this in the Cabalar semantics in the previous example would write this* $\neg\neg dom_1(f) \wedge \neg\neg dom_2(f)$ *which has no Cabalar stable models.*

It remains an open question whether this behavior can be captured in the Cabalar semantics.

10.7 Proofs

10.7.1 Proof of Theorem 21

Recall the definition: $J \preceq^{\boldsymbol{c}} I$ if

- J and I have the same universe and agree on all constants not in $\boldsymbol{c}$;
- $p^J \subseteq p^I$ for all predicate constants in $\boldsymbol{c}$; and
- $f^J(\boldsymbol{\xi}) = u$ or $f^J(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$ for all function constants in $\boldsymbol{c}$ and all lists $\boldsymbol{\xi}$ of elements in the universe.

As before, let $\boldsymbol{d}$ be a list of constants that is similar to $\boldsymbol{c}$ and is disjoint from σ. The notion of $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$ is straightforwardly extended to the case when J and I are partial interpretations.

Lemma 32 *For any partial interpretations I and J of signature σ, we have $J \preceq^{\boldsymbol{c}} I$ iff $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p \boldsymbol{d} \preceq \boldsymbol{c}$.*

Proof. By definition of $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$, J and I have the same universe and agree on all constants in $\sigma \setminus \boldsymbol{c}$, which is the first condition of $J \preceq^{\boldsymbol{c}} I$.

Recall the definition: $\boldsymbol{d} \preceq \boldsymbol{c}$ is

$$(\boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}) \wedge (\boldsymbol{d}^{func} \leq \boldsymbol{c}^{func}).$$

$J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p \boldsymbol{d}^{pred} \leq \boldsymbol{c}^{pred}$ iff, for every predicate constant p in $\boldsymbol{c}$,

$$J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \underset{p}{\models} \forall \boldsymbol{x}(p(\boldsymbol{x})^{\boldsymbol{c}}_{\boldsymbol{d}} \rightarrow p(\boldsymbol{x})),$$

which is equivalent to saying that $(p^{\boldsymbol{c}}_{\boldsymbol{d}})^{J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I} \subseteq p^{J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I}$. Since I does not interpret any constant from $\boldsymbol{d}$ and $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ does not interpret any constant from $\boldsymbol{c}$, this is equivalent to $(p^{\boldsymbol{c}}_{\boldsymbol{d}})^{J^{\boldsymbol{c}}_{\boldsymbol{d}}} \subseteq p^I$ and further to $p^J \subseteq p^I$, which is the second condition of $J \preceq^{\boldsymbol{c}} I$.

$J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p (\boldsymbol{d}^{func} \leq \boldsymbol{c}^{func})$ iff, for every function constant f in $\boldsymbol{c}$,

$$J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p \forall \boldsymbol{x}((f(\boldsymbol{x})^{\boldsymbol{c}}_{\boldsymbol{d}} \neq f(\boldsymbol{x})^{\boldsymbol{c}}_{\boldsymbol{d}}) \vee (f(\boldsymbol{x})^{\boldsymbol{c}}_{\boldsymbol{d}} = f(\boldsymbol{x}))),$$

which is equivalent to saying that $f^J(\boldsymbol{\xi}) = u$ or $f^J(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$ for all $\boldsymbol{\xi}$, the third condition of $J \preceq^{\boldsymbol{c}} I$. ∎

Lemma 33 *For any partial interpretations I and J of signature σ, we have $J \prec^{\boldsymbol{c}} I$ iff $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p \boldsymbol{d} \prec \boldsymbol{c}$.*

Proof. Immediate from Lemma 32 since

- $J \prec^{\boldsymbol{c}} I$ iff $J \preceq^{\boldsymbol{c}} I$ and not $I \preceq^{\boldsymbol{c}} J$, and
- $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p \boldsymbol{d} \prec \boldsymbol{c}$ iff $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p \boldsymbol{d} \preceq \boldsymbol{c}$ and $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \not\models_p \boldsymbol{c} \preceq \boldsymbol{d}$.

∎

Lemma 34 *For any sentence F of signature σ and any partial interpretations I and J of σ such that $J \preceq^{\boldsymbol{c}} I$,*

(a) if $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p F^{\dagger}(\boldsymbol{d})$, then $I \models_p F$.

(b) if $\langle J, I\rangle \models_{pht} F$, then $\langle I, I\rangle \models_{pht} F$.

Proof. Each of (a) and (b) can be proved by induction on F.

We will show only the case when F is an atomic sentence. The other cases are straightforward:

Part (a): Let F be an atomic sentence. Assume $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p F^{\dagger}(\boldsymbol{d})$, i.e., $J \models_p F$.

- *Subcase 1:* F is of the form $p(\boldsymbol{t})$. Since $J \preceq^{\boldsymbol{c}} I$, it follows that $I \models_p F$.

- *Subcase 2:* F is of the form $t_1 = t_2$. Since $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \models F(\boldsymbol{d})$, $t_1^J = t_2^J \neq u$. From $J \preceq^{\mathbf{c}} I$, it follows that $t_1^I = t_2^I \neq u$, i.e., $I \models_p F$.

Part (b): Let F be an atomic sentence. Assume $\langle J, I\rangle \models_{pht} F$, i.e., $\langle J, I\rangle, h \models_{pht} F$

- *Subcase 1:* F is of the form $p(\boldsymbol{t})$. Since $J \preceq^{\mathbf{c}} I$, it follows that $\langle J, I\rangle, t \models_{pht} F$.

- *Subcase 2:* F is of the form $t_1 = t_2$. Since $\langle J, I\rangle, h \models_{pht} F$, $t_1^J = t_2^J \neq u$. From $J \preceq^{\mathbf{c}} I$, it follows that $t_1^I = t_2^I \neq u$, i.e., $\langle J, I\rangle, t \models_{pht} F$.

∎

Lemma 35 *Let F be a sentence of signature σ, and let I and J be partial interpretations of σ such that $J \preceq^{\mathbf{c}} I$. We have $J \models_p gr_I[F]^{\underline{I}}$ iff $\langle J, I\rangle \models_{pht} F$.*

Proof. By induction on F.

Case 1: F is an atomic sentence. Clearly, $gr_I[F]$ is F.

- *Subcase 1:* $I \not\models_p F$. Then $gr_I[F]^{\underline{I}}$ is $\bot$, which J does not satisfy. Further, since $\langle I, I\rangle \not\models_{pht} F$, by Lemma 34 (b), it follows that $\langle J, I\rangle \not\models_{pht} F$.

- *Subcase 2:* $I \models_p F$. Then $gr_I[F]^{\underline{I}}$ is F. It is clear that $J \models_p F$ iff $\langle J, I\rangle \models_{pht} F$.

Case 2: F is $G \wedge H$ or $G \vee H$. The claim follows immediately from I.H. on G and H.

Case 3: F is $G \to H$. Consider the following subcases:

- *Subcase 1:* $I \not\models_p G \to H$. $gr_I[G \to H]^{\underline{I}}$ is $\bot$, which J does not satisfy. Further, $\langle I, I\rangle \not\models_p G \to H$. By Lemma 34 (b), $\langle J, I\rangle \not\models_p G \to H$.

- *Subcase 2:* $I \models_p G \to H$. $gr_I[G \to H]^{\underline{I}}$ is equivalent to $gr_I[G]^{\underline{I}} \to gr_I[H]^{\underline{I}}$. Further, $\langle J, I\rangle \models_{pht} G \to H$ is equivalent to $\langle J, I\rangle \not\models_{pht} G$ or $\langle J, I\rangle \models_{pht} H$. Then the claim follows from I.H. on G and H.

Case 4: F is $\forall x G(x)$, or $\exists x G(x)$. By induction on $G(\xi^\diamond)$ for each ξ in the universe.

∎

Theorem 21 *Let F be a first-order sentence of signature σ and let $\mathbf{c}$ be a list of intensional constants. For any partial interpretation I of σ, $\langle I, I\rangle$ is a partial equilibrium model of F iff*

- *$I \models_p F$, and*
- *for every partial interpretation J of σ such that $J \prec^{\mathbf{c}} I$, we have $J \not\models_p gr_I[F]^{\underline{I}}$.*

Proof. Clearly, $I \models_p F$ iff $\langle I, I\rangle \models_{pht} F$. By Lemma 35, for every partial interpretation J of σ such that $J \prec^{\mathbf{c}} I$, $J \not\models_p gr_I[F]^{\underline{I}}$ iff $\langle J, I\rangle \not\models_{pht} F$. ∎

10.7.2 Proof of Theorem 22

Lemma 36 *Let F be a sentence of signature σ, and let I and J be partial interpretations of σ. We have $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \models_p F^\dagger(\boldsymbol{d})$ iff $\langle J, I\rangle \models_{pht} F$.*

Proof. By induction on F.

Case 1: F is an atomic sentence. $F^\dagger(\boldsymbol{d})$ is $F(\boldsymbol{d})$. $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \models_p F(\boldsymbol{d})$ iff $J \models_p F$ iff $\langle J, I\rangle, h \models_{pht} F$ iff $\langle J, I\rangle \models_{pht} F$.

Case 2: F is $G \wedge H$ or $G \vee H$. Follows by I.H. on G and H.

Case 3: F is $G \rightarrow H$. Consider the following subcases:

- *Subcase 1:* $I \not\models_p G \rightarrow H$. Clearly, $J^{\mathbf{c}}_{\boldsymbol{d}} \cup I \not\models_p G \rightarrow H$ and $\langle J, I\rangle \not\models_{pht} G \rightarrow H$.

- *Subcase 2:* $I \models_p G \to H$. Then $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p (G \to H)^{\dagger}(\boldsymbol{d})$ iff $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models_p G^{\dagger}(\boldsymbol{d}) \to H^{\dagger}(\boldsymbol{d})$. Further, $\langle J, I\rangle \models_{pht} G \to H$ iff $\langle J, I\rangle \not\models_{pht} G$ or $\langle J, I\rangle \models_{pht} H$. Then the claim follows from I.H. on G and H.

Case 4: F is $\forall x G(x)$, or $\exists x G(x)$. By induction on $G(\xi^{\diamond})$ for each ξ in the universe. ∎

Theorem 22 *For any sentence F, a PHT-interpretation $\langle I, I\rangle$ is a partial equilibrium model of F relative to $\boldsymbol{c}$ iff $I \models_p CBL[F; \boldsymbol{c}]$.*

Proof. By definition, CBL$[F; \boldsymbol{c}]$ is

$$F \wedge \neg\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} \prec \boldsymbol{c} \wedge F^{\dagger}(\widehat{\boldsymbol{c}})).$$

Clearly, $I \models_p F$ iff $\langle I, I\rangle \models_{pht} F$. From Lemma 33 and Lemma 36, it follows that $I \models_p \neg\exists\widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} \prec \boldsymbol{c} \wedge F^{\dagger}(\widehat{\boldsymbol{c}}))$ iff there is no interpretation J of σ such that $J \prec^{\boldsymbol{c}} I$ and $\langle J, I\rangle \models_{pht} F$. ∎

10.7.3 Proof of Theorem 23

Lemma 37 *Assume that K and X are partial multi-valued interpretations of σ and Y is a propositional interpretation of σ^{prop} which is a subset of X such that*

$$K(c) = X(c) \text{ iff either } c = X(c) \in Y \text{ or } X(c) = undef.$$

We have that $K \models F^X$ (when we view F as a multi-valued formula of σ) iff $Y \models F^X$ (when we view F as a propositional formula of σ^{prop}).

Proof. By induction on F. We show only the case of atoms. The other cases are straightforward.

Let F be an atom $c = v$. If $X \models c = v$, then F^X is F and it cannot be that $X(c) = undef$. The claim follows from the assumption since $K \models c = v$ iff $Y \models c = v$. If $X \not\models c = v$, then F^X is $\perp$, which neither K nor Y satisfies. ∎

Theorem 23 *Let F be a multi-valued formula of signature σ, which can be also viewed as a propositional formula of signature σ^{prop}.*

(a) *If a partial interpretation I of σ is a partial multi-valued stable model of F, then I can be viewed as an interpretation of σ^{prop} that is a propositional stable model of $F \wedge UC_\sigma$ (in the sense of Ferraris (2005)).*

(b) *If an interpretation I of σ^{prop} is a propositional stable model of $F \wedge UC_\sigma$ (in the sense of Ferraris (2005)), then I can be viewed as a partial interpretation of σ that is a partial multi-valued stable model of F.*

Proof. (a) Assume X of signature σ is a partial multi-valued stable model of F. This means $X \models F$ and no partial multi-valued interpretation K that is a subset of X satisfies F^X. Now since X is a partial multi-valued intepretation, $X \models UC_\sigma$. Then clearly $X \models F$ when viewed as a propositional formula of signature σ^{prop}.

So, we wish to show that there is no interpretation Y of signature σ^{prop} such that $Y \subset X$ when X is viewed as a set of propositional atoms and $Y \models (F \wedge UC_\sigma)^X$ when viewed as a propositional formula of signature σ^{prop}. To do so, we prove the contrapositive. We will show that if there is an interpretation Y of signature σ^{prop} such that $Y \subset X$ when X is viewed as a set of propositional atoms and $Y \models (F \wedge UC_\sigma)^X$ when viewed as a propositional formula of signature σ^{prop}, then there is a partial interpretation K that is a subset of X that satisfies F^X when viewed as a multi-valued formula of signature σ.

Given such an interpretation Y, we create K as follows. For each $c \in \sigma$,

$$K(c) = \begin{cases} v & \text{if } c = v \in Y \\ undef & : \text{ if } c = v \notin Y \text{ for any } v \end{cases}$$

Note that this no longer requires there to be two explicit elements in $Dom(c)$

Note that since $Y \subset X$, there is at least one $c \in \sigma$ and $v \in Dom(c)$ such that $c = v \in X$ but $c = v \notin Y$. For this c, $K(c) = undef \neq X(c)$ so K and X are different. Further, when $c = v \in Y$, then $c = v \in X$ and so $X(c) = K(c)$, thus K is a subset of X.

In addition, we have that $K(c) = X(c)$ iff $c = X(c) \in Y$ or $X(c) = undef$. Now, since $Y \models (F \wedge UC_\sigma)^X$, it follows that $Y \models F^X$. Thus, from Lemma 37 it follows that since $Y \models F^X$, then $K \models F^X$.

(b) Assume X of signature σ^{prop} is a stable model of $F \wedge UC_\sigma$. This means that $X \models F \wedge UC_\sigma$ and no interpretation Y such that $Y \subset X$ satisfies $(F \wedge UC_\sigma)^X$. Since $X \models UC_\sigma$, then X can be viewed as a partial multi-valued interpretation. Then clearly, $X \models F$.

Now, we wish to show that there is no partial interpretation K of signature σ that is a subset of X satisfying F^X. To do so, we prove the contrapositive. We will show that if there is a partial interpretation K of signature σ that is a subset of X and $K \models F^X$, then there is an interpretation Y such that $Y \subset X$ that satisfies $(F \wedge UC_\sigma)^X$. Now since we already have seen that $X \models UC_\sigma$, then $(UC_\sigma)^X$ is equivalent to $\top$ so we need only show that there is an interpretation Y such that $Y \subset X$ that satisfies F^X.

Given such an interpretation K, we create Y as follows. Let us view K as a set of propositional atoms. We will take $Y = X \cap K$. Clearly $Y \subset X$. In addition, we have that $K(c) = X(c)$ iff $c = X(c) \in Y$ or $X(c) = undef$. Thus, from Lemma 16 it follows that since $Y \models F^X$, then $K \models F^X$. ∎

10.7.4 Proof of Corollary 6

Corollary 6 *For any multi-valued formula F of signature σ and any partial interpretation I, we have that I is a multi-valued stable model of F iff I is a partial multi-valued stable model of $F \wedge EC_\sigma$.*

Proof. By Theorem 14, we have that the multi-valued propositional stable models of F and the propositional stable models of $F \wedge UEC_\sigma$ are in one-to-one correspondence. On the other hand, Theorem 23 tells us that the partial multi-valued stable models of $F \wedge EC_\sigma$ and the propositional stable models of $F \wedge EC_\sigma \wedge UC_\sigma$ are in one-to-one correspondence. The corollary follows then by the fact that $F \wedge EC_\sigma \wedge UC_\sigma$ is exactly $F \wedge UEC_\sigma$.

10.7.5 Proof of Theorem 24

Given a multi-valued interpretation I, we define the partial first-order interpretation I^{pfo} as follows:

- $|I^{pfo}| = \{v \mid v \in Dom(c) \text{ for some } c \in \sigma\}$;

- $I^{pfo}(v) = v$ for each $v \in Dom(c)$ for each $c \in \sigma$;

- $I^{pfo}(c) = I(c)$ for each multi-valued constant $c \in \sigma$.

Lemma 38 *For any MVP-formula F of signature σ, and any partial MVP-interpretation I of σ whose multi-valued constants are $\boldsymbol{c}$, I is a partial multi-valued stable model of F iff I^{pfo} is a partial stable model of F with respect to $\boldsymbol{c}$ viewed as a first-order formula of signature σ^{pfo}.*

Proof. ($\Rightarrow$) Consider any partial multi-valued stable model I of F. This means that I satisfies F and no subset K of I satisfies $F^{\underline{I}}$. It is clear by induction that $I^{pfo} \models_p F$; the base case is when F is an atomic formula $c = v$ and clearly by definition of I^{pfo}, we have $I \models c = v$ iff $I^{pfo} \models c = v$.

Thus, we must show that there is no $J \prec^{bfc} I^{pfo}$ such that $J \models_p F^{\underline{I^{pfo}}}$. To do so, we will show that if there is such a J, then we can create a partial MVP-interpreation K such that $K \subset I$ and $K \models_p F^{\underline{I}}$.

Assume there is some J such that $J \prec^{bfc} I^{pfo}$ and $J \models_p F^{\underline{I^{pfo}}}$. We create K from J as follows. For each $c \in \sigma$

$$K(c) = \begin{cases} I(c) & \text{if } c^J = I(c) \\ u & \text{otherwise} \end{cases}$$

We first show that $K \subset I$. Since $J \prec^{bfc} I^{pfo}$, there must be some constant $c \in \sigma$ such that $c^J = u$ and $c^{I^{pfo}} \neq u$. However, since $c^{I^{pfo}} = I(c)$ by definition of I^{pfo}, we have that $c^J \neq I(c)$ and so $K(c) = u$ but $I(c) \neq u$. Thus, $K \subset I$.

We now show that $K \models_p F^{\underline{I}}$ iff $J \models_p F^{\underline{I^{pfo}}}$ by induction on F. From this, we will conclude that since we assume I is a partial multi-valued stable model, then no such K exists and so it follows that no such J exists, which means I^{pfo} is a partial stable model of F with respect to $\boldsymbol{c}$.

- Case 1: F is an MVP atom $c = v$. If $I \models_p F$ then by definition, $I^{pfo} \models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $c = v$. Then, by definition of J, we have $K \models_p F^{\underline{I}}$ iff $J \models_p F^{\underline{I^{pfo}}}$.

  On the other hand, if $I \not\models_p F$ then by definition, $I^{pfo} \not\models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

- Case 2: F is $G \wedge H$. If $I \models_p F$ then by definition, $I^{pfo} \models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are $G^{\underline{I^{pfo}}} \wedge H^{\underline{I^{pfo}}}$ and $G^{\underline{I}} \wedge H^{\underline{I}}$ so the claim follows by induction on $G^{\underline{I}}$, $G^{\underline{I^{pfo}}}$ and $H^{\underline{I^{pfo}}}$, $H^{\underline{I}}$.

  On the other hand, if $I \not\models_p F$ then by definition, $I^{pfo} \not\models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

- Case 3: F is $G \vee H$. If $I \models_p F$ then by definition, $I^{pfo} \models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are $G^{\underline{I^{pfo}}} \vee H^{\underline{I^{pfo}}}$ and $G^{\underline{I}} \vee H^{\underline{I}}$ so the claim follows by induction on $G^{\underline{I}}$, $G^{\underline{I^{pfo}}}$ and $H^{\underline{I^{pfo}}}$, $H^{\underline{I}}$.

  On the other hand, if $I \not\models_p F$ then by definition, $I^{pfo} \not\models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

- Case 4: F is $G \rightarrow H$. If $I \not\models_p G$ then by definition, $I^{pfo} \not\models_p G^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are $\bot \rightarrow H^{\underline{I^{pfo}}}$ and $\bot \rightarrow H^{\underline{I}}$. Then we have $K \models_p F^{\underline{I}}$ and $J \models_p F^{\underline{I^{pfo}}}$.

  If $I \models_p H$ and $I \models_p G$ then by definition, $I^{pfo} \models_p H^{pfo}$ and $I^{pfo} \models_p G^{pfo}$. Then we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are $G^{\underline{I^{pfo}}} \rightarrow H^{\underline{I^{pfo}}}$ and $G^{\underline{I}} \rightarrow H^{\underline{I}}$ so the claim follows by induction on $G^{\underline{I}}$, $G^{\underline{I^{pfo}}}$ and $H^{\underline{I^{pfo}}}$, $H^{\underline{I}}$.

If $I \not\models_p H$ and $I \models_p G$ then by definition, $I^{pfo} \not\models_p H^{pfo}$ and $I^{pfo} \models G^{pfo}$. Then we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

- Case 5: F is $\neg G$. If $I \models_p G$ then by definition, $I^{pfo} \models_p G^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

  On the other hand, if $I \not\models_p G$ then by definition, $I^{pfo} \not\models_p G^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\neg\bot$. Then we have $K \models_p F^{\underline{I}}$ and $J \models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

($\Leftarrow$) Consider any partial stable model I^{pfo} of F. This means that $I^{pfo} \models F$ and there is no interpretation J such that $J \prec^{\boldsymbol{c}} I^{pfo}$ and $J \models F^{\underline{I^{pfo}}}$. It is clear by induction that $I \models_p F$; the base case is when F is an atomic formula $c = v$ and clearly by definition of I^{pfo}, we have $I \models c = v$ iff $I^{pfo} \models c = v$.

Then it only remains to be shown no partial MVP-interpretation K that is a subset of I satisfies $F^{\underline{I}}$. To show this, we will show that if there is such a K, then we can create an interpretation J such that $J \prec^{bfc} I^{pfo}$ and $J \models F^{\underline{I^{pfo}}}$.

Assume such a K exists and let $J = K^{pfo}$.

We first show that $J \prec^{bfc} I^{pfo}$. Since K is a subset of I, there must be some constant $c \in \sigma$ such that $I(c) \neq u$ but $K(c) = u$. Then, by definition of $J = K^{pfo}$, we have that $c^J = u$ but $c^{I^{pfo}} \neq u$. Thus, $J \prec^{bfc} I^{pfo}$.

We now show that $K \models_p F^{\underline{I}}$ iff $J \models_p F^{\underline{I^{pfo}}}$ by induction on F. From this, we will conclude that since we assume I^{pfo} is a partial stable model with respect to $\boldsymbol{c}$, then no such J exists and so it follows that no such K exists, which means I is a partial multi-valued stable model of F.

- Case 1: F is an MVP atom $c = v$. If $I \models_p F$ then by definition, $I^{pfo} \models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $c = v$. Then, by definition of J, we have $K \models_p F^{\underline{I}}$ iff $J \models_p F^{\underline{I^{pfo}}}$.

  On the other hand, if $I \not\models_p F$ then by definition, $I^{pfo} \not\models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

- Case 2: F is $G \wedge H$. If $I \models_p F$ then by definition, $I^{pfo} \models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are $G^{\underline{I^{pfo}}} \wedge H^{\underline{I^{pfo}}}$ and $G^{\underline{I}} \wedge H^{\underline{I}}$ so the claim follows by induction on $G^{\underline{I}}$, $G^{\underline{I^{pfo}}}$ and $H^{\underline{I^{pfo}}}$, $H^{\underline{I}}$.

  On the other hand, if $I \not\models_p F$ then by definition, $I^{pfo} \not\models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

- Case 3: F is $G \vee H$. If $I \models_p F$ then by definition, $I^{pfo} \models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are $G^{\underline{I^{pfo}}} \vee H^{\underline{I^{pfo}}}$ and $G^{\underline{I}} \vee H^{\underline{I}}$ so the claim follows by induction on $G^{\underline{I}}$, $G^{\underline{I^{pfo}}}$ and $H^{\underline{I^{pfo}}}$, $H^{\underline{I}}$.

  On the other hand, if $I \not\models_p F$ then by definition, $I^{pfo} \not\models_p F^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

- Case 4: F is $G \rightarrow H$. If $I \not\models_p G$ then by definition, $I^{pfo} \not\models_p G^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are $\bot \rightarrow H^{\underline{I^{pfo}}}$ and $\bot \rightarrow H^{\underline{I}}$. Then we have $K \models_p F^{\underline{I}}$ and $J \models_p F^{\underline{I^{pfo}}}$.

If $I \models_p H$ and $I \models_p G$ then by definition, $I^{pfo} \models_p H^{pfo}$ and $I^{pfo} \models_p G^{pfo}$. Then we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are $G^{\underline{I^{pfo}}} \rightarrow H^{\underline{I^{pfo}}}$ and $G^{\underline{I}} \rightarrow H^{\underline{I}}$ so the claim follows by induction on $G^{\underline{I}}$, $G^{\underline{I^{pfo}}}$ and $H^{\underline{I^{pfo}}}$, $H^{\underline{I}}$.

If $I \not\models_p H$ and $I \models_p G$ then by definition, $I^{pfo} \not\models_p H^{pfo}$ and $I^{pfo} \models G^{pfo}$. Then we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

- Case 5: F is $\neg G$. If $I \models_p G$ then by definition, $I^{pfo} \models_p G^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\bot$. Then we have $K \not\models_p F^{\underline{I}}$ and $J \not\models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

  On the other hand, if $I \not\models_p G$ then by definition, $I^{pfo} \not\models_p G^{pfo}$ and so we have $F^{\underline{I^{pfo}}}$ and $F^{\underline{I}}$ are both $\neg\bot$. Then we have $K \models_p F^{\underline{I}}$ and $J \models_p F^{\underline{I^{pfo}}}$ so in this case, the claim holds.

Theorem 24 *Let F be a multi-valued formula of signature σ.*

(a) *If an interpretation I of σ is a partial multi-valued stable model of F, then I^{none} is a multi-valued stable model of $F \wedge \bigwedge_{c\in\sigma} Big(c{=}\text{NONE} \vee \neg(c{=}\text{NONE})\Big)$.*

(b) *If an interpretation J of σ^{none} is a stable model of $F \wedge \bigwedge_{c\in\sigma} Big(c{=}\text{NONE} \vee \neg(c{=}\text{NONE})\Big)$ then $J = I^{none}$ for some partial multi-valued stable model I of F.*

We first note that by Theorem 14, we can view $F \wedge \bigwedge_{c\in\sigma} \Big(c{=}\text{NONE} \vee \neg(c{=}\text{NONE})\Big)$ of signature σ^{none} as a first-order formula under the functional stable model semantics. Similarly by Proposition 38, we can view F as a first-order formula under the Cabalar semantics. Then, by Theorem 28 the claim follows.

Lemma 39 *Let F be a $\boldsymbol{c}$-plain sentence of signature σ, let I, K be total interpretations of σ, and let J be a partial interpretation of σ such that*

- $J \prec^{\boldsymbol{c}} I$ *and* $K <^{\boldsymbol{c}} I$;
- $p^J = p^K$ *for every predicate constant;*
- $f^J(\boldsymbol{\xi}) = u$ *iff* $f^K(\boldsymbol{\xi}) \neq f^I(\boldsymbol{\xi})$ *for every function constant* f *and every* $\boldsymbol{\xi} \in |I|^n$ *where* n *is the arity of* f.

We have $K \models gr_I[F]^{\underline{I}}$ *iff* $J \models_p gr_I[F]^{\underline{I}}$.

Proof.

Case 1: F is an atomic sentence of the form $p(\boldsymbol{t})$. Since F is $\boldsymbol{c}$-plain, $\boldsymbol{t}$ contains no constants from $\boldsymbol{c}$, and by the assumption $J \prec^{\boldsymbol{c}} I$ and $K <^{\boldsymbol{c}} I$, we have $\boldsymbol{t}^J = \boldsymbol{t}^K = \boldsymbol{t}^I$. Since J and K agree on p, the claim holds.

Case 2: F is an atomic sentence of the form $f(\boldsymbol{t}) = t_1$.

- *Subcase 1:* $I \not\models f(\boldsymbol{t}) = t_1$. Then $gr_I[F]^{\underline{I}}$ is $\bot$, so the claim holds.
- *Subcase 2:* $I \models f(\boldsymbol{t}) = t_1$. Then $gr_I[F]^{\underline{I}}$ is $f(\boldsymbol{t}) = t_1$. Further, from the assumption that F is $\boldsymbol{c}$-plain, $\boldsymbol{t}$ and t_1 contain no constants from $\boldsymbol{c}$, and by the assumptions that $J \prec^{\boldsymbol{c}} I$, $K <^{\boldsymbol{c}} I$ and that I is total, we have $\boldsymbol{t}^J = \boldsymbol{t}^K = \boldsymbol{t}^I \neq u$ and $t_1^J = t_1^K = t_1^I \neq u$.

  Either $f(\boldsymbol{t})^J \neq u$ or $f(\boldsymbol{t})^J = u$. In the first case, since $J \prec^{\boldsymbol{c}} I$, we have $f(\boldsymbol{t})^J = f(\boldsymbol{t})^I$. Also, by the assumption on K, $f(\boldsymbol{t})^K = f(\boldsymbol{t})^I$. Consequently, $J \models f(\boldsymbol{t}) = t_1$ and $K \models f(\boldsymbol{t}) = t_1$.

In the second case, $J \not\models f(\boldsymbol{t}) = t_1$. Also, by the assumption on K, $f(\boldsymbol{t})^K \neq f(\boldsymbol{t})^I = t_1^I = t_1^K$, so $K \not\models f(\boldsymbol{t}) = t_1$.

The other cases are straightforward. ∎

Recall the definitions: for two classical interpretations I, K of the same signature σ with the same universe and a list $\boldsymbol{c}$ of distinct predicate and function constants, we write $K <^{\mathbf{c}} I$ if

$$K \text{ and } I \text{ agree on all constants in } \sigma \setminus \boldsymbol{c}, \tag{10.3}$$

$$p^K \subseteq p^I \text{ for all predicates } p \text{ in } \boldsymbol{c}, \text{ and} \tag{10.4}$$

$$K \text{ and } I \text{ do not agree on } \boldsymbol{c}. \tag{10.5}$$

Similarly, for two partial interpretations J and I of the same signature σ over the same universe $|I|$, and a set of constants $\boldsymbol{c}$, $J \prec^{\mathbf{c}} I$ is equivalent to

$$J \text{ and } I \text{ agree on all constants in } \sigma \setminus \boldsymbol{c}, \tag{10.6}$$

$$p^J \subseteq p^I \text{ for all predicates } p \text{ in } \boldsymbol{c}, \text{ and} \tag{10.7}$$

$$J \text{ and } I \text{ do not agree on } \boldsymbol{c}. \tag{10.8}$$

with the additional requirement that

$$\begin{array}{l}\text{for every function constant } f \in \boldsymbol{c}, \text{ and every } \boldsymbol{\xi} \in |I|^n \text{ where } n \\ \text{is the arity of } f,\ f^I(\boldsymbol{\xi}) = f^J(\boldsymbol{\xi}) \text{ or } f^J(\boldsymbol{\xi}) = u.\end{array} \tag{10.9}$$

If we drop (10.8), this is equivalent to $J \preceq^{\mathbf{c}} I$.

Lemma 40 *Let F be a $\mathbf{c}$-plain sentence of signature σ, and let I be total interpretation of σ that satisfies $\exists xy(x \neq y)$. There is a partial interpretation J such that $J \prec^{\mathbf{c}} I$ and $J \models_p gr_I[F]^{\underline{I}}$ iff there is a total interpretation K such that $K <^{\mathbf{c}} I$ and $K \models gr_I[F]^{\underline{I}}$.*

Proof. *Left-to-right:* Let J be a partial interpretation such that $J \prec^{\boldsymbol{c}} I$ and $J \models gr_I[F]^{\underline{I}}$. We construct the total interpretation K as follows. For each constant d not in $\boldsymbol{c}$, $d^K = d^J = d^I$. For each predicate constant p in $\boldsymbol{c}$ and each $\boldsymbol{\xi} \in |I|^n$ where n is the arity of p,

$$p^K(\boldsymbol{\xi}) = p^J(\boldsymbol{\xi}) \ ,$$

and, for each function constant f in $\boldsymbol{c}$ and each $\boldsymbol{\xi} \in |I|^n$ where n is the arity of f,

$$f^K(\boldsymbol{\xi}) = \begin{cases} f^I(\boldsymbol{\xi}) & \text{if } f^J(\boldsymbol{\xi}) \neq u; \\ m(f^I(\boldsymbol{\xi})) & \text{otherwise} \end{cases}$$

where m is a mapping $m : |I| \to |I|$ such that $\forall x(m(x) \neq x)$ (note that such a mapping requires $I \models \exists xy(x \neq y)$).

We now show that $K <^{\boldsymbol{c}} I$. It is immediate from the assumption $J \prec^{\boldsymbol{c}} I$ and by definition that (10.3) and (10.4) hold. Consider the following cases.

- Case 1: For every function constant $f \in \boldsymbol{c}$ and every $\boldsymbol{\xi} \in |I|^n$ where n is the arity of f, $f^J(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$ (note that since I is total, these cannot be u). From (10.8), it follows that there is at least one predicate constant p in $\boldsymbol{c}$ such that $p^J \subset p^I$. However, by definition of K, $p^K \subset p^I$ and so (10.5) holds.

- Case 2: There is some function constant $f \in \boldsymbol{c}$ and some $\boldsymbol{\xi} \in |I|^n$ where n is the arity of f such that $f^J(\boldsymbol{\xi}) \neq f^I(\boldsymbol{\xi})$. From (10.9), it follows that $f^J(\boldsymbol{\xi}) = u$ and thus by definition of K, $f^K(\boldsymbol{\xi}) = m(f^I(\boldsymbol{\xi})) \neq f^I(\boldsymbol{\xi})$ and so (10.5) holds.

By Lemma 39, the fact $K \models gr_I[F]^{\underline{I}}$ follows from the assumption $J \models_p gr_I[F]^{\underline{I}}$.

Right-to-left: Let K be a total interpretation such that $K <^{\boldsymbol{c}} I$ and $K \models gr_I[F]^{\underline{I}}$. We construct the partial interpretation J as follows. For each constant d not in $\boldsymbol{c}$,

$d^K = d^J = d^I$. For each predicate constant p in $\boldsymbol{c}$ and each $\boldsymbol{\xi} \in |I|^n$ where n is the arity of p,

$$p^J(\boldsymbol{\xi}) = p^K(\boldsymbol{\xi}) \ ,$$

and, for each function constant f in $\boldsymbol{c}$ and each $\boldsymbol{\xi} \in |I|^n$ where n is the arity of f,

$$f^J(\boldsymbol{\xi}) = \begin{cases} f^I(\boldsymbol{\xi}) & \text{if } f^K(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi}); \\ u & \text{otherwise.} \end{cases}$$

We now show that $J \prec^{\mathbf{c}} I$. It is immediate from the assumption that $K <^{\mathbf{c}} I$ and by definition that (10.6) and (10.7) hold. Consider the following cases.

- Case 1: For every function constant $f \in \boldsymbol{c}$ and every $\boldsymbol{\xi} \in |I|^n$ where n is the arity of f, $f^K(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$. By definition of J, $f^J(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$ and so (10.9) holds. Now since (10.5) holds, there is at least one predicate constant p such that $p^K \subset p^I$. However, by definition of J, $p^J \subset p^I$ and so (10.8) holds.

- Case 2: There is some function constant $f \in \boldsymbol{c}$ and some $\boldsymbol{\xi} \in |I|^n$ where n is the arity of f such that $f^K(\boldsymbol{\xi}) \neq f^I(\boldsymbol{\xi})$. By definition of J, it must be that $f^J(\boldsymbol{\xi}) = u$ and thus (10.9) and (10.8) both hold.

By Lemma 39, the fact $J \models_p gr_I[F]^{\underline{I}}$ follows from the assumption $K \models gr_I[F]^{\underline{I}}$.

Theorem 25 *For any $\boldsymbol{c}$-plain formula F of signature σ, any list $\boldsymbol{c}$ of intensional constants, and any total interpretation I of σ satisfying $\exists xy(x \neq y)$, $I \models SM[F; \boldsymbol{c}]$ iff $I \models_p CBL[F; \boldsymbol{c}]$.*

Proof. We use Theorem 1 and Theorem 21 to refer to the grounding and reduct based definitions rather than the second-order logic based definitions. The claim follows from Lemma 40. ∎

10.7.7 Proof of Theorem 26

Theorem 26 *For any sentence F of signature σ in Clark Normal Form that is tight on $\boldsymbol{c}$, and any total interpretation I of σ satisfying $\exists xy(x \neq y)$, $I \models SM[F;\boldsymbol{c}]$ iff $I \models_p CBL[F;\boldsymbol{c}]$.*

Proof.

By Corollary 7, $I \models_p \mathrm{CBL}[F;\boldsymbol{c}]$ iff $I \models \mathrm{SM}[UF_{\boldsymbol{c}}(F);\boldsymbol{c}]$, so it remains to check that $I \models \mathrm{SM}[UF_{\boldsymbol{c}}(F);\boldsymbol{c}]$ iff $I \models \mathrm{SM}[F;\boldsymbol{c}]$.

It is easy to check that the completion of $UF_{\boldsymbol{c}}(F)$ relative to $\boldsymbol{c}$ is equivalent to the completion of F relative to $\boldsymbol{c}$. By Theorem 2 from Bartholomew and Lee (2013a), we conclude that $\mathrm{SM}[UF_{\boldsymbol{c}}(F);\boldsymbol{c}]$ is equivalent to $\mathrm{SM}[F;\boldsymbol{c}]$. ∎

10.7.8 Proof of Theorem 27

Lemma 41 *For any partial interpretation I and any atomic sentence $p(t_1,\dots,t_k)$ and $f(t_1,\dots,t_{k-1}) = t_k$,*

(a) $I \models_p p(t_1,\dots,t_k)$ *iff*

$$I \underset{p}{\models} \exists x_{n_1} \dots x_{n_j}(p(t_1,\dots,t_k)'' \wedge t_{n_1} = x_{n_1} \wedge \dots \wedge t_{n_j} = x_{n_j})$$

where $\{n_1,\dots,n_j\} \subseteq \{1,\dots,k\}$ and $p(t_1,\dots,t_k)''$ is obtained from $p(t_1,\dots,t_k)$ by replacing each t_{n_i} in $p(t_1,\dots,t_k)$ with x_{n_i}.

(b) $I \models_p f(t_1, \ldots, t_{k-1}) = t_k$ *iff*

$$I \models_p \exists x_{n_1} \ldots x_{n_j}((f(t_1, \ldots, t_{k-1}) = t_k)'' \wedge t_{n_1} = x_{n_1} \wedge \cdots \wedge t_{n_j} = x_{n_j})$$

where $\{n_1, \ldots, n_j\} \subseteq \{1, \ldots, k\}$ *and* $(f(t_1, \ldots, t_{k-1}) = t_k)''$ *is obtained from* $f(t_1, \ldots, t_{k-1}) = t_k$ *by replacing each* t_{n_i} *in* $f(t_1, \ldots, t_{k-1}) = t_k$ *with* x_{n_i}.

Proof. Consider the following cases.

Case 1: $t_i^I = u$ for some $i \in \{n_1, \ldots, n_j\}$. Clearly, $I \not\models_p p(t_1, \ldots, t_k)$ and $I \not\models_p f(t_1, \ldots, t_{k-1}) = t_k$. It is also the case that $I \not\models_p t_i = \boldsymbol{\xi}^\diamond$ for any $\boldsymbol{\xi} \in |I|$ so we have

$$I \not\models_p \exists x_{n_1} \ldots x_{n_j}(p(t_1, \ldots, t_k)'' \wedge t_{n_1} = x_{n_1} \wedge \cdots \wedge t_{n_j} = x_{n_j}) \tag{10.10}$$

and

$$I \not\models_p \exists x_{n_1} \ldots x_{n_j}((f(t_1, \ldots, t_{k-1}) = t_k)'' \wedge t_{n_1} = x_{n_1} \wedge \cdots \wedge t_{n_j} = x_{n_j}) \ . \tag{10.11}$$

Case 2: $t_i^I = u$ for some $i \in \{1, \ldots, k\} \setminus \{n_1, \ldots, n_j\}$. Clearly, $I \not\models_p p(t_1, \ldots, t_k)$ and $I \not\models_p f(t_1, \ldots, t_{k-1}) = t_k$. Also, since t_i remains in $p(t_1, \ldots, t_k)''$ and $(f(t_1, \ldots, t_k) = t)''$, we have $I \not\models_p p(t_1, \ldots, t_k)''$ and $I \not\models_p (f(t_1, \ldots, t_k) = t)''$, from which (10.10) and (10.11) follow.

Case 3: $t_i^I \neq u$ for all $i \in \{1, \ldots, k\}$. Condition (a) clearly holds because it coincides with classical equivalence. For Condition (b), consider two subcases:

- *Subcase 1:* $f(t_1, \ldots, t_{k-1})^I \neq u$. Clearly, Condition (b) coincides with classical equivalence.

- *Subcase 2:* $f(t_1, \ldots, t_{k-1})^I = u$. Clearly, $I \not\models_p f(t_1, \ldots, t_{k-1}) = t_k$. Now in

$$\exists x_{n_1} \ldots x_{n_j}((f(t_1, \ldots, t_{k-1}) = t_k)'' \wedge t_{n_1} = x_{n_1} \wedge \cdots \wedge t_{n_j} = x_{n_j}),$$

there is only one set of values for $x_{n_1} \ldots x_{n_j}$ that satisfies the last j conjunctive terms—when x_{n_i} is mapped to $t_{n_i}^I$. However, for this set of values, $((f(t_1, \ldots, t_{k-1}))'')^I = f(t_1, \ldots, t_{k-1})^I = u$ (where $(f(t_1, \ldots, t_{k-1}))''$ is obtained from $f(t_1, \ldots, t_{k-1})$ by replacing each t_{n_i} with x_{n_i}) so that

$I \not\models_p \exists x_{n_1} \ldots x_{n_j}((f(t_1, \ldots, t_{k-1}) = t_k)'' \wedge t_{n_1} = x_{n_1} \wedge \cdots \wedge t_{n_k} = x_{n_k})$.

∎

Lemma 42 *Given a sentence F, a set of constants $\boldsymbol{c}$, and a partial interpretation I, we have $I \models_p F$ iff $I \models_p UF_{\boldsymbol{c}}(F)$.*

Proof. The proof is by induction on the number of unfolding that needs to be done. More precisely, for any formula F, we define $NU_{\boldsymbol{c}}(F)$ ("Needed Unfolding") as follows.

- $NU_{\boldsymbol{c}}(p(t_1, \ldots, t_k)) =$
$$\begin{cases} 0 & \text{if } p(t_1, \ldots, t_k) \text{ is } \boldsymbol{c}\text{-plain;} \\ max(NU_{\boldsymbol{c}}(t_1 = x), \ldots, NU_{\boldsymbol{c}}(t_k = x)) + 1 & \text{otherwise.} \end{cases}$$
- $NU_{\boldsymbol{c}}(f(t_1, \ldots, t_{k-1}) = t_k) =$
$$\begin{cases} 0 & \text{if } f(t_1, \ldots, t_{k-1}) = t_k \text{ is } \boldsymbol{c}\text{-plain;} \\ max(NU_{\boldsymbol{c}}(t_1 = x), \ldots, NU_{\boldsymbol{c}}(t_k = x)) + 1 & \text{otherwise.} \end{cases}$$
- $NU_{\boldsymbol{c}}(G \odot H) = max(NU_{\boldsymbol{c}}(G), NU_{\boldsymbol{c}}(H)) + 1$, where $\odot \in \{\wedge, \vee, \rightarrow\}$.
- $NU_{\boldsymbol{c}}(QxG) = NU_{\boldsymbol{c}}(G) + 1$, where $Q \in \{\forall, \exists\}$.

Case 1: F is a $\boldsymbol{c}$-plain atomic sentence. F is identical to $UF_{\boldsymbol{c}}(F)$ so the claim holds.

Case 2: F is $p(\boldsymbol{t})$ where $\boldsymbol{t}$ contains at least one constant from $\boldsymbol{c}$. Let $t_{n_1} \ldots t_{n_j}$ be the j terms in $\boldsymbol{t}$ containing at least one constant from $\boldsymbol{c}$. Now $UF_{\boldsymbol{c}}(F)$ is

$\exists x_{n_1} \ldots x_{n_j}(p(t_1, \ldots, t_k)'' \wedge UF_{\boldsymbol{c}}(t_{n_1} = x_{n_1}) \wedge \cdots \wedge UF_{\boldsymbol{c}}(t_{n_j} = x_{n_j}))$ where $p(t_1, \ldots, t_k)''$

is obtained from $p(t_1, \dots, t_k)$ by replacing each t_{n_i} in $p(t_1, \dots, t_k)$ with x_{n_i}. Since $NU_{\boldsymbol{c}}(F) > NU_{\boldsymbol{c}}(t_{n_i} = \boldsymbol{\xi}^\diamond)$ for each $\boldsymbol{\xi} \in |I|$ and each $i \in \{1, \dots, j\}$, by I.H. on $t_{n_i} = \boldsymbol{\xi}^\diamond$, $UF_{\boldsymbol{c}}(t_{n_i} = x_{n_i})$ can be replaced by $t_{n_i} = x_{n_i}$ so that $I \models_p UF_{\boldsymbol{c}}(F)$ iff $I \models_p \exists x_{n_1} \dots x_{n_j}(p(t_1, \dots, t_k)'' \wedge t_{n_1} = x_{n_1} \wedge \cdots \wedge t_{n_j} = x_{n_j})$. By Lemma 41 the latter is equivalent to $I \models_p F$.

Case 3: F is $f(\boldsymbol{t}) = t_1$ where at least one of $\boldsymbol{t}$ and t_1 contain at least one constant from $\boldsymbol{c}$. Let $t_{n_1} \dots t_{n_j}$ be the j terms in $\boldsymbol{t}$ and t_1 containing at least one constant from $\boldsymbol{c}$. Now $UF_{\boldsymbol{c}}(F)$ is $\exists x_{n_1} \dots x_{n_j}((f(\boldsymbol{t}) = t_1)'' \wedge UF_{\boldsymbol{c}}(t_{n_1} = x_{n_1}) \wedge \cdots \wedge UF_{\boldsymbol{c}}(t_{n_j} = x_{n_j}))$, where $(f(\boldsymbol{t}) = t_1)''$ is obtained from $f(\boldsymbol{t}) = t_1$ by replacing each t_{n_i} in $f(\boldsymbol{t}) = t_1$ with x_{n_i}. Since $NU_{\boldsymbol{c}}(F) > NU_{\boldsymbol{c}}(t_{n_i} = \boldsymbol{\xi}^\diamond)$ for each $\boldsymbol{\xi} \in |I|$ and each $i \in \{1, \dots, j\}$, by I.H. on $t_{n_i} = \boldsymbol{\xi}^\diamond$, $UF_{\boldsymbol{c}}(t_{n_i} = x_{n_i})$ can be replaced by $t_{n_i} = x_{n_i}$ so that $I \models_p UF_{\boldsymbol{c}}(F)$ iff $I \models_p \exists x_{n_1} \dots x_{n_j}((f(\boldsymbol{t}) = t_1)'' \wedge t_{n_1} = x_{n_1} \wedge \cdots \wedge t_{n_j} = x_{n_j})$. By Lemma 41 the latter is equivalent to $I \models_p F$.

Case 4: F is $G \odot H$ for $\odot \in \{\wedge, \vee, \rightarrow\}$. By I.H. on G and H.

Case 5: F is $QxF(x)$ for $Q \in \{\forall, \exists\}$. By I.H. on $F(\boldsymbol{\xi}^\diamond)$ for each $\boldsymbol{\xi} \in |I|$. ∎

Theorem 27 *For any sentence F, any list $\boldsymbol{c}$ of constants, and any partial interpretation I, we have $I \models_p CBL[F; \boldsymbol{c}]$ iff $I \models_p CBL[UF_{\boldsymbol{c}}(F); \boldsymbol{c}]$.*

Proof. By definition, CBL$[F; \boldsymbol{c}]$ is

$$F \wedge \neg \exists \widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} \prec \boldsymbol{c} \wedge F^\dagger(\widehat{\boldsymbol{c}}))$$

and CBL$[UF_{\boldsymbol{c}}(F); \boldsymbol{c}]$ is by definition

$$UF_{\boldsymbol{c}}(F) \wedge \neg \exists \widehat{\boldsymbol{c}}(\widehat{\boldsymbol{c}} \prec \boldsymbol{c} \wedge (UF_{\boldsymbol{c}}(F))^\dagger(\widehat{\boldsymbol{c}})).$$

Now, for any partial interpretation I of signature $\sigma \supseteq \boldsymbol{c}$, by Lemma 42, $I \models_p F$ iff

$I \models_p UF_{\boldsymbol{c}}(F)$. It is sufficient to show that, for any partial interpretation J, $J_{\boldsymbol{d}}^{\boldsymbol{c}} \cup I \models_p \boldsymbol{d} \prec \boldsymbol{c} \wedge F^{\dagger}(\boldsymbol{d})$ iff $J_{\boldsymbol{d}}^{\boldsymbol{c}} \cup I \models_p \boldsymbol{d} \prec \boldsymbol{c} \wedge (UF_{\boldsymbol{c}}(F))^{\dagger}(\boldsymbol{d})$.

Case 1: F is an atomic sentence. $F^{\dagger}(\boldsymbol{d})$ is $F(\boldsymbol{d})$, and $UF_{\boldsymbol{c}}(F)^{\dagger}(\boldsymbol{d})$ is $UF_{\boldsymbol{c}}(F)(\boldsymbol{d})$. $J_{\boldsymbol{d}}^{\boldsymbol{c}} \cup I \models_p F(\boldsymbol{d})$ iff $J \models_p F$. Similarly, $J_{\boldsymbol{d}}^{\boldsymbol{c}} \cup I \models_p UF_{\boldsymbol{c}}(F)(\boldsymbol{d})$ iff $J \models_p UF_{\boldsymbol{c}}(F)$. By Lemma 41, $J \models_p F$ iff $J \models_p UF_{\boldsymbol{c}}(F)$, so the claim follows.

Case 2: F is $G \odot H$ for $\odot \in \{\wedge, \vee\}$. By induction on G and H.

Case 3: F is $G \rightarrow H$. $F^{\dagger}(\boldsymbol{d})$ is $(G^{\dagger}(\boldsymbol{d}) \rightarrow H^{\dagger}(\boldsymbol{d})) \wedge (G \rightarrow H)$ and $(UF_{\boldsymbol{c}}(F))^{\dagger}(\boldsymbol{d})$ is $(UF_{\boldsymbol{c}}(G))^{\dagger}(\boldsymbol{d}) \rightarrow (UF_{\boldsymbol{c}}(H))^{\dagger}(\boldsymbol{d})) \wedge (UF_{\boldsymbol{c}}(G) \rightarrow UF_{\boldsymbol{c}}(H))$. The equivalence between the first conjunctive terms (under partial satisfaction) is by I.H. on G and H, and the equivalence between the second conjunctive terms (under partial satisfaction) is by Lemma 42.

Case 4: F is $QxG(x)$ for $Q \in \{\forall, \exists\}$. By I.H. on $F(\boldsymbol{\xi}^{\diamond})$ for each $\boldsymbol{\xi} \in |I|$. ∎

10.7.9 Proof of Corollary 7

Corollary 7 *For any sentence F, any list $\boldsymbol{c}$ of constants, and any total interpretation I satisfying $\exists xy(x \neq y)$, we have $I \models_p CBL[F; \boldsymbol{c}]$ iff $I \models_p CBL[UF_{\boldsymbol{c}}(F); \boldsymbol{c}]$ iff $I \models SM[UF_{\boldsymbol{c}}(F); \boldsymbol{c}]$.*

Proof. The equivalence between the first and the second conditions is by Theorem 27. The equivalence between the second and the third conditions is by Theorem 25 since $UF_{\boldsymbol{c}}(F)$ is $\boldsymbol{c}$-plain. ∎

Lemma 43 *Given a σ-plain formula G of signature σ, a partial interpretation I satisfies G iff I^{none} satisfies G^{none}.*

Proof. By induction on G.

- Case 1: G is a (σ-plain) ground atomic formula of signature σ^* which is σ extended with object names from $|I|$ (not including NONE). G^{none} is the same as G in this case.

  - G is $p(\boldsymbol{\xi}^*)$. Note that since $\boldsymbol{\xi}^*$ are object names, I does not map any of these to u. Thus, by definition of I^{none}, $p(\boldsymbol{\xi}^*)^{\underline{I^{none}}} = p(\boldsymbol{\xi}^*)^I$ so certainly the claim holds.
  - G is $\xi_1^* = \xi_2^*$. The claim follows immediately from the fact that $(\xi_1^*)^I = (\xi_1^*)^{\underline{I^{none}}} = \xi_1$ and $(\xi_2^*)^I = (\xi_2^*)^{\underline{I^{none}}} = \xi_2$.
  - G is $f(\boldsymbol{\xi}^*) = \xi^*$. Note that since $\boldsymbol{\xi}^*$ and ξ^* are object names, I does not map any of these to u. Now if $f(\boldsymbol{\xi}^*)^I = \text{u}$, then by definition of I^{none}, $f(\boldsymbol{\xi}^*)^{\underline{I^{none}}} = \text{NONE}$. In this case, neither I nor I^{none} satisfy G. On the other hand, if $f(\boldsymbol{\xi}^*)^I \neq \text{u}$, then by definition of I^{none}, $f(\boldsymbol{\xi}^*)^{\underline{I^{none}}} = f(\boldsymbol{\xi}^*)^I$ so certainly the claim follows.

- Case 2: G is $H_1 \odot H_2$ where $\odot \in \{\wedge, \vee, \rightarrow\}$. G^{none} is $(H_1)^{none} \odot (H_2)^{none}$. By I.H. on H_1 and H_2, the claim follows.

- Case 3: G is $\exists x H(x)$. G^{none} is $\exists x(H(x)^{none} \wedge x \neq \text{NONE})$.

  ($\Rightarrow$) Assume $I \models_p G$. That means there is some $\xi \in |I|$ such that $I \models_p H(\xi)$. By I.H. on $H(\xi^\diamond)$ for every $\xi \in |I|$, we then have that there is some $\xi \in |I|$ such that $I^{none} \models H(\xi)^{none}$. Since $\xi \neq \textsc{none}$ for all $\xi \in |I|$, we have that there is some $\xi \in |I|$ such that $I^{none} \models H(\xi)^{none} \wedge \xi \neq \textsc{none}$. Finally, since $|I| \subseteq |I^{none}|$, we further have that there is some $\xi \in |I^{none}|$ such that $I^{none} \models H(\xi)^{none} \wedge \xi \neq \textsc{none}$, which is the definition of $I^{none} \models G^{none}$.

($\Leftarrow$) Assume $I^{none} \models G^{none}$. That means there is some $\xi \in |I^{none}|$ such that $I^{none} \models H(\xi)^{none} \wedge \xi \neq \textsc{none}$. It then follows that there is some $\xi \in |I|$ such that $I^{none} \models H(\xi)^{none}$. By I.H. on $H(\xi^{\diamond})$ for every $\xi \in |I|$, it then follows that there is some $\xi \in |I|$ such that $I \models_p H(\xi)$, which is the definition of $I \models_p G$.

- Case 4: G is $\forall x H(x)$. G^{none} is $\forall x(x \neq \textsc{none} \to H(x)^{none})$.

  ($\Rightarrow$) Assume $I \models_p G$. That means for every $\xi \in |I|$, we have $I \models_p H(\xi)$. By I.H. on $H(\xi^{\diamond})$ for every $\xi \in |I|$, it follows that for every $\xi \in |I|$ we have $I^{none} \models H(\xi)^{none}$. Since $\xi \neq \textsc{none}$ for all $\xi \in |I|$, we have that there is some $\xi \in |I|$ such that $I^{none} \models \xi \neq \textsc{none} \to H(\xi)^{none}$. Finally, since $|I| \subseteq |I^{none}|$ and since the implication $\xi \neq \textsc{none} \to H(\xi)^{none}$ is trivially satisfied when $\xi = \textsc{none}$, it further follows that for every $\xi \in |I^{none}|$ we have $I^{none} \models \xi \neq \textsc{none} \to H(\xi)^{none}$, which is the definition of $I^{none} \models G^{none}$.

  ($\Leftarrow$) Assume $I^{none} \models G^{none}$. That means for every $\xi \in |I^{none}|$ we have $I^{none} \models \xi \neq \textsc{none} \to H(\xi)^{none}$. Since $|I| \subseteq |I^{none}|$, it certainly follows that for every $\xi \in |I|$ we have $I^{none} \models \xi \neq \textsc{none} \to H(\xi)^{none}$. Then, since $\xi \neq \textsc{none}$ is true for every $\xi \in |I|$, it follows that for every $\xi \in |I|$ we have $I^{none} \models H(\xi)^{none}$. Then by I.H. on $H(\xi^{\diamond})$ for every $\xi \in |I|$ it follows that for every $\xi \in |I|$ we have $I \models_p H(\xi)$, which is the definition of $I \models_p G$

Note: the σ-plain assumption is only used for the atomic formulas $t_1 = t_2$ not $p(t)$.

Theorem 28 *For any sentence F of signature σ,*

(a) If I is a Cabalar stable model of F relative to $\boldsymbol{c}$, then I^{none} is a stable model of (10.2), recalling that this is

$$(UF_\sigma(F))^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle,$$

relative to $\boldsymbol{c}$.

(b) If an interpretation J such that $\text{NONE}^J = \text{NONE}$ is a stable model of (10.2) relative to $\boldsymbol{c}$, then $J = I^{none}$ for some Cabalar stable model I of F relative to $\boldsymbol{c}$.

We first note that by Theorem 7 in Bartholomew and Lee (2013b), I is a Cabalar stable model of F relative to $\boldsymbol{c}$ iff I is a Cabalar stable model of $UF_\sigma(F)$ relative to $\boldsymbol{c}$ (the theorem is about $UF_{\boldsymbol{c}}(F)$ but the same proof should hold for $UF_\sigma(F)$).

For notational simplicity, let $G = UF_\sigma(F)$. We will prove the theorem in terms of G. Further, we note that the Cabalar stable models of G are precisely the Cabalar stable models of F by Theorem 7 in Bartholomew and Lee (2013b). That is, we will show

(a) If I is a Cabalar stable model of G relative to $\boldsymbol{c}$, then I^{none} is a stable model of

$$G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle$$

relative to $\boldsymbol{c}$.

(b) If an interpretation J such that $\text{NONE}^J = \text{NONE}$ is a stable model of

$$G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle$$

relative to $\boldsymbol{c}$ then $J = I^{none}$ for some Cabalar stable model I of G relative to $\boldsymbol{c}$.

Proof. (a) Assume that I is a Cabalar stable model of G relative to $\boldsymbol{c}$. That is, $I \models_p G$ and for every partial interpretation K such that $K \prec^{\boldsymbol{c}} I$, we have $K \not\models_p gr_I[G]^{\underline{I}}$. We wish to show that I^{none} is a stable model of $G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle$ relative to $\boldsymbol{c}$. That is, we wish to show that $I^{none} \models G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle$ and for every interpretation L such that $L <^{\boldsymbol{c}} I^{none}$, we have $L \not\models gr_{I^{none}}(G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle)^{\underline{I^{none}}}$.

By Lemma 43, we have that since we assume $I \models_p G$, we conclude that $I^{none} \models G^{none}$. Then, since $\bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle$ is a tautology in classical logic, we have $I^{none} \models G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle$.

We now show that if for every partial interpretation K such that $K \prec^{\boldsymbol{c}} I$, we have $K \not\models_p gr_I[G]^{\underline{I}}$ then for any L such that $L <^{\boldsymbol{c}} I^{none}$, we have $L \not\models gr_{I^{none}}(G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle)^{\underline{I^{none}}}$. To do so, we prove the contrapositive; if there is some L such that $L <^{\boldsymbol{c}} I^{none}$ and $L \models gr_{I^{none}}(G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \text{NONE} \rangle)^{\underline{I^{none}}}$, then there is some partial interpretation K such that $K \prec^{\boldsymbol{c}} I$ and $K \not\models_p gr_I[G]^{\underline{I}}$. Given such an L, we construct such a K as follows. First, let $|K| = |I|$. For every predicate $p \in \sigma$, we define $p^K = p^L$. For every function $f \in \sigma$ of arity n and every tuple of objects $\boldsymbol{\xi}$ from $(|I| \cup \{\text{u}\})^n$, we define

$$f^K(\boldsymbol{\xi}) = \begin{cases} f^L(\boldsymbol{\xi}) & \text{if } f^L(\boldsymbol{\xi}) \neq \text{NONE},\ f^L(\boldsymbol{\xi}) = f^{I^{none}}(\boldsymbol{\xi}), \\ & \text{and } \boldsymbol{\xi} \in |I|^n; \\ \text{u} & \text{otherwise.} \end{cases}$$

Assuming $L <^{\boldsymbol{c}} I^{none}$, we show that $K \prec^{\boldsymbol{c}} I$. We first show that $K \preceq^{\boldsymbol{c}} I$.

- By definition, K and I both have the same universe.

- Since L and I^{none} agree on all constants not in $\boldsymbol{c}$, it is easy to see by definition of K and I^{none} that K and I agree on all constants not in $\boldsymbol{c}$.

- Consider any predicate constant $p \in \boldsymbol{c}$ and any tuple $\boldsymbol{\xi}$ from $|I|$. If $p(\boldsymbol{\xi})^K = 1$, then by definition of K, it must be that $p(\boldsymbol{\xi})^L = 1$. Then, since $p^L \subseteq p^{\underline{I^{none}}}$, it must be that $p(\boldsymbol{\xi})^{\underline{I^{none}}} = 1$. Finally, by definition of I^{none}, it follows that $p(\boldsymbol{\xi})^I = 1$. Thus it holds that $p^K \subseteq p^I$.

- Consider any function constant $f \in \boldsymbol{c}$ of arity n and any tuple $\boldsymbol{\xi}$ from $|I|^n$. We have three cases and wish to show that $f^k(\boldsymbol{\xi}) = \mathrm{u}$ or $f^K(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$.

  - If $f^L(\boldsymbol{\xi}) = \textsc{none}$, then by definition of K, $f^K(\boldsymbol{\xi}) = \mathrm{u}$ so in this case, the claim follows.

  - If $f^L(\boldsymbol{\xi}) \neq \textsc{none}$ and $f^L(\boldsymbol{\xi}) = f^{I^{none}}$. Then by definition of K, $f^K(\boldsymbol{\xi}) = f^L(\boldsymbol{\xi}) = f^{I^{none}}(\boldsymbol{\xi})$. Then, by definition of I^{none}, we have $f^{I^{none}}(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$ and so $f^K(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$ so in this case, the claim follows.

  - If $f^L(\boldsymbol{\xi}) \neq \textsc{none}$ and $f^L(\boldsymbol{\xi}) \neq f^{I^{none}}$. Then by definition of K, $f^K(\boldsymbol{\xi}) = \mathrm{u}$ so in this case, the claim follows.

  If $f^L(\boldsymbol{\xi}) = f^{I^{none}}(\boldsymbol{\xi})$, then by definition of K, $f^K(\boldsymbol{\xi}) = f^L(\boldsymbol{\xi}) = f^{I^{none}}(\boldsymbol{\xi})$. Then, by definition of I^{none}, we have $f^K(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$. On the other hand, if $f^L(\boldsymbol{\xi}) \neq f^{I^{none}}(\boldsymbol{\xi})$, then by definition of K, we have $f^K(\boldsymbol{\xi}) = \mathrm{u}$. Therefore, it holds for every function constant $f \in \sigma$ and every list $\boldsymbol{\xi}$ of elements from $|I|$ that $f^k(\boldsymbol{\xi}) = \mathrm{u}$ or $f^K(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$.

Thus, we have $K \preceq^{\boldsymbol{c}} I$.

We now show $(I \preceq^{\boldsymbol{c}} K)$ does not hold and conclude that $K \prec^{\boldsymbol{c}} I$. Since $L <^{\boldsymbol{c}} I^{\text{NONE}}$, we consider two cases:

- Case 1: There is some predicate $p \in \boldsymbol{c}$ of arity n and tuple $\boldsymbol{\xi}$ of objects from $|I^{none}|$ such that $p(\boldsymbol{\xi})^L = 0$ but $p(\boldsymbol{\xi})^{\underline{I^{none}}} = 1$. We first note that by definition of I^{none} that if $\boldsymbol{\xi}$ is not in $|I|^n$, then $p(\boldsymbol{\xi})^{\underline{I^{none}}} = 0$ so it must be that $\boldsymbol{\xi}$ is in $|I|^n$. Then by definition of K, we have that $p(\boldsymbol{\xi})^K = 0$ and by definition of I^{none}, it follows that $p(\boldsymbol{\xi})^I = 1$ so in this case $(I \preceq^{\boldsymbol{c}} K)$ does not hold.

- Case 2: There is some function $f \in \boldsymbol{c}$ of arity n and tuple $\boldsymbol{\xi}$ of objects from $|I^{none}|$ such that $f(\boldsymbol{\xi})^L \neq f(\boldsymbol{\xi})^{I^{none}}$. We need to show that $f(\boldsymbol{\xi})^I \neq \mathrm{u}$ and $f(\boldsymbol{\xi})^K \neq f(\boldsymbol{\xi})^I$.

  We show that for this to be the case, it must be that $f(\boldsymbol{\xi})^{I^{none}} \neq \textsc{none}$. Assume to the contrary that $f(\boldsymbol{\xi})^{I^{none}} = \textsc{none}$, then since $L \models gr_{I^{none}}(G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \textsc{none} \rangle)^{\underline{I^{none}}}$, and in particular $L \models gr_{I^{none}}(\bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \textsc{none} \rangle)^{\underline{I^{none}}}$, which contains a conjunctive term equivalent to $f(\boldsymbol{\xi}) = \textsc{none} \vee \bot$ and so it must be that $f(\boldsymbol{\xi})^L = \textsc{none}$ which contradicts the assumption that $f(\boldsymbol{\xi})^L \neq f(\boldsymbol{\xi})^{I^{none}}$.

  Thus it must be that $f(\boldsymbol{\xi})^{I^{none}} \neq \textsc{none}$. By definition of I^{none}, this means that $\boldsymbol{\xi}$ is in $|I|^n$ and $f^I \neq u$. However, by definition of K, since $f^L(\boldsymbol{\xi}) = f^{I^{none}}(\boldsymbol{\xi})$ does not hold, $f^K(\boldsymbol{\xi}) = \mathrm{u}$ and so we have $f^K(\boldsymbol{\xi}) \neq f^I(\boldsymbol{\xi})$. Thus in this case $(I \preceq^{\boldsymbol{c}} K)$ does not hold.

We now show by induction on G that $K \models_p gr_I[G]^{\underline{I}}$ iff $L \models gr_{I^{none}}(G^{none})^{\underline{I^{none}}}$ and then since we assume $L \models gr_{I^{none}}(G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) = \textsc{none} \rangle)^{\underline{I^{none}}}$ then certainly $L \models gr_{I^{none}}(G^{none})^{\underline{I^{none}}}$, and then we will conclude that $K \models_p gr_I[G]^{\underline{I}}$.

- Case 1: G is a (σ-plain) ground atomic formula of extended signature σ^* which is σ extended with object names from $|I|$ (not including NONE).

  If $I \not\models_p G$, then $I^{none} \not\models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and $gr_{I^{none}}(G^{none})^{\underline{I^{none}}}$ are both $\bot$, which neither L nor K satisfy so in this case, the claim holds.

  If instead $I \models_p G$, then $I^{none} \models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and $gr_{I^{none}}(G^{none})^{\underline{I^{none}}}$ are both G, so by definition of K, $K \models_p G$ iff $L \models G$.

- Case 2: G is $H_1 \odot H_2$ where $\odot \in \{\wedge, \vee, \rightarrow\}$ and so $gr_I[G]^{\underline{I}}$ is $gr_I[H_1]^{\underline{I}} \odot gr_I[H_2]^{\underline{I}}$. G^{none} is $(H_1)^{none} \odot (H_2)^{none}$.

  If $I \not\models_p G$, then $I^{none} \not\models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and $gr_{I^{none}}(G^{none})^{\underline{I^{none}}}$ are both $\bot$, which neither L nor K satisfy so in this case, the claim holds.

  On the other hand, if $I \models_p G$, then $I^{none} \models G^{none}$ by Lemma 43 and so $gr_{I^{none}}[G^{none}]^{\underline{I^{none}}}$ is

$$gr_{I^{none}}[(H_1)^{none}]^{\underline{I^{none}}} \odot gr_{I^{none}}[(H_2)^{none}]^{\underline{I^{none}}}.$$

  By I.H. on $gr_I[H_1]^{\underline{I}}$ and $gr_{I^{none}}[(H_1)^{none}]^{\underline{I^{none}}}$ and $gr_I[H_2]^{\underline{I}}$ and $gr_{I^{none}}[(H_2)^{none}]^{\underline{I^{none}}}$.

- Case 3: G is $\exists x H(x)$. G^{none} is $\exists x (H(x)^{none} \wedge x \neq \text{NONE})$.

  If $I \not\models_p G$, then $I^{none} \not\models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and $gr_{I^{none}}(G^{none})^{\underline{I^{none}}}$ are both $\bot$, which neither L nor K satisfy so in this case, the claim holds.

  On the other hand, if $I \models_p G$, then $I^{none} \models G^{none}$ by Lemma 43 and so $gr_I[G]^{\underline{I}}$

is

$$\{gr_I[H(\xi^\diamond)]^{\underline{I}} : \xi \in |I|\}^\vee$$

and $gr_{I^{none}}[G^{none}]^{\underline{I^{none}}}$ is

$$\{gr_{I^{none}}[H(\xi^\diamond)^{none}]^{\underline{I^{none}}} \wedge \xi^\diamond \neq \textsc{none} : \xi \in |(I^{none})|\}^\vee.$$

Now we note that $|(I^{none})| = |I| \cup \{\textsc{none}\}$. Further, we note that since $\xi^\diamond \neq \textsc{none}$ is not satisfied when $\xi = \textsc{none}$, the latter reduct is equivalent to

$$\{gr_{I^{none}}[H(\xi^\diamond)^{none}]^{\underline{I^{none}}} \wedge \xi^\diamond \neq \textsc{none} : \xi \in |I|\}^\vee.$$

The further, we note that for all $\xi \in |I|$, $\xi^\diamond \neq \textsc{none}$ is satisfied so that this is further equivalent to

$$\{gr_{I^{none}}[H(\xi^\diamond)^{none}]^{\underline{I^{none}}} : \xi \in |I|\}^\vee.$$

Thus the claim follows by induction on $gr_I[H(\xi^\diamond)]$ and $gr_{I^{none}}[H(\xi^\diamond)^{none}]$ for every $\xi \in |I|$.

- Case 4: G is $\forall x H(x)$. G^{none} is $\forall x(x \neq \textsc{none} \to H(x)^{none})$.

  If $I \not\models_p G$, then $I^{none} \not\models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and $gr_{I^{none}}(G^{none})^{\underline{I^{none}}}$ are both $\bot$, which neither L nor K satisfy so in this case, the claim holds.

  On the other hand, if $I \models_p G$, then $I^{none} \models G^{none}$ by Lemma 43 and so $gr_I[G]^{\underline{I}}$ is

  $$\{gr_I[H(\xi^\diamond)]^{\underline{I}} : \xi \in |I|\}^\wedge$$

  and $gr_{I^{none}}[H^{none}]^{\underline{I^{none}}}$ is

  $$\{\xi^\diamond \neq \textsc{none} \to gr_{I^{none}}[H(\xi^\diamond)^{none}]^{\underline{I^{none}}} : \xi \in |(I^{none})|\}^\wedge.$$

Now we note that $|(I^{none})| = |I| \cup \{\text{NONE}\}$. Further, we note that since $\xi^{\diamond} \neq$ NONE is not satisfied when $\xi = \text{NONE}$, the latter reduct is equivalent to

$$\{\xi^{\diamond} \neq \text{NONE} \rightarrow gr_{I^{none}}[H(\xi^{\diamond})^{none}]^{\underline{I^{none}}} : \xi \in |I|\}^{\wedge}.$$

The further, we note that for all $\xi \in |I|$, $\xi^{\diamond} \neq$ NONE is satisfied so that this is further equivalent to

$$\{gr_{I^{none}}[H(\xi^{\diamond})^{none}]^{\underline{I^{none}}} : \xi \in |I|\}^{\wedge}.$$

Thus the claim follows by induction on $gr_I[H(\xi^{\diamond})]$ and $gr_{I^{none}}[H(\xi^{\diamond})^{none}]$ for every $\xi \in |I|$.

(b) We assume that J is an interpretation such that $\text{NONE}^J = \text{NONE}$ and J is a stable model of $G^{none} \wedge \forall \boldsymbol{x}\langle f(\boldsymbol{x}) = \text{NONE}\rangle$. We wish to show that $J = I^{none}$ for some Cabalar stable model I of G relative to $\boldsymbol{c}$.

We will show this by constructing such an I from J. Let $I = J^{invnone}$ where $J^{invnone}$ is the partial interpretation obtained from J as follows:

- $|J^{invnone}| = |J| \setminus \{\text{NONE}\}$.
- for every function constant $f \in \sigma$ and $\boldsymbol{\xi} \in |I|^n$ where n is the arity of f,

$$f^{J^{invnone}}(\boldsymbol{\xi}) = \begin{cases} f^J(\boldsymbol{\xi}) & \text{if } f^J(\boldsymbol{\xi}) \neq \text{NONE}; \\ u & \text{otherwise;} \end{cases}$$

- For every predicate $p \in \sigma$ and $\boldsymbol{\xi} \in |I|^n$ where n is the arity of p, $p^{J^{invnone}}(\boldsymbol{\xi}) = p^J(\boldsymbol{\xi})$.

We now wish to show that $I \models_p G$ and for every partial interpretation K such that $K \prec^{\mathbf{c}} I$, we have $K \not\models_p gr_I[G]^{\underline{I}}$.

Since we assume $J \models G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x})$ it follows that $J \models G^{none}$, then by Lemma 43 (noting that $I^{none} = (J^{invnone})^{none} = J$), we conclude that $I \models_p G$.

We now show that if for any L such that $L <^{\boldsymbol{c}} J$, we have $L \not\models gr_J(G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) \rangle \underline{^J}$, then for every partial interpretation K such that $K \prec^{\boldsymbol{c}} I$, we have $K \not\models_p gr_I[G]\underline{^I}$.

To do so, we prove the contrapositive; if there is some partial interpretation K such that $K \prec^{\boldsymbol{c}} I$ and $K \not\models_p gr_I[G]\underline{^I}$, then there is some L such that $L <^{\boldsymbol{c}} J$ and $L \models gr_J(G^{none} \wedge \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x} \langle f(\boldsymbol{x}) \rangle \underline{^J}$. Given such an K, we construct such a L as follows. First, let $|L| = |J|$. For every predicate $p \in \sigma$, we define $p^L = p^K$. For every function $f \in \sigma$ of arity n and every tuple of objects $\boldsymbol{\xi}$ from $(|J|)^n$, we define

$$f^L(\boldsymbol{\xi}) = \begin{cases} f^K(\boldsymbol{\xi}) & \text{if } f^K(\boldsymbol{\xi}) \neq \mathrm{u} \text{ and } \boldsymbol{\xi} \in |I|^n; \\ \textsc{none} & \text{otherwise.} \end{cases}$$

Assuming $K \prec^{\boldsymbol{c}} I$, we show that $L <^{\boldsymbol{c}} J$.

- By definition, L and J both have the same universe.
- Since K and I agree on all constants not in $\boldsymbol{c}$, it is easy to see by definition of L and J that L and J agree on all constants not in $\boldsymbol{c}$.
- Consider any predicate constant $p \in \boldsymbol{c}$ and any tuple $\boldsymbol{\xi}$ from $|J|$. We first note by definition of L that if $\boldsymbol{\xi}$ has at least one NONE, then $p^L(\boldsymbol{\xi}) = 0$ so there is nothing to be proven. Now we consider when $\boldsymbol{\xi}$ has no NONE. If $p^L(\boldsymbol{\xi}) = 1$, then by definition of L, it must be that $p^K(\boldsymbol{\xi}) = 1$. Then, since $p^K \subseteq p^I$, it must be that $p^I(\boldsymbol{\xi}) = 1$. Finally, by definition of I, it follows that $p^I(\boldsymbol{\xi}) = 1$ since $\boldsymbol{\xi}$ is from $|I|$. Thus it holds that $p^L \subseteq p^J$.
- We wish to show that L and J do not agree on $\boldsymbol{c}$. We consider two cases

– There is some predicate $p \in \boldsymbol{c}$ and some list of objects $\boldsymbol{\xi}$ from $|I|$ such that $p^K(\boldsymbol{\xi}) = 0$ and $p^I(\boldsymbol{\xi}) = 1$. By definition of I, we have $p^J(\boldsymbol{\xi}) = 1$ and by definition of L, we have $p^L(\boldsymbol{\xi}) = 0$ so in this case, the claim holds.

– There is some function $f \in \boldsymbol{c}$ and some list of objects $\boldsymbol{\xi}$ from $|I|$ such that $f^I(\boldsymbol{\xi}) \neq \mathrm{u}$ and $f^K(\boldsymbol{\xi}) \neq f^I(\boldsymbol{\xi})$. In particular, since $K \preceq^{\boldsymbol{c}} I$, this means that $f^K = \mathrm{u}$. Now since $f^I(\boldsymbol{\xi}) \neq \mathrm{u}$, by definition of I, we have $f^I(\boldsymbol{\xi}) \neq \textsc{none}$ and by definition of L, we have $f^L(\boldsymbol{\xi}) = \textsc{none}$ so in this case, the claim holds.

Thus, we have $L <^{\boldsymbol{c}} J$.

To show that $L \models gr_J(\bigwedge_{f\in\boldsymbol{c}} \forall \boldsymbol{x}\langle f(\boldsymbol{x})\rangle^{\underline{J}}$ we first note that $gr_J(\bigwedge_{f\in\boldsymbol{c}} \forall \boldsymbol{x}\langle f(\boldsymbol{x})\rangle^{\underline{J}}$ is equivalent to the conjunction of $f(\boldsymbol{\xi}) = \textsc{none}$ for every $f \in \sigma$ and $\boldsymbol{\xi}$ from $|J|$ such that $f^J(\boldsymbol{\xi}) = \textsc{none}$. We see that by definition of I that it must be that $f^I(\boldsymbol{\xi}) = \mathrm{u}$. Then, since we assume that $K \prec^{\boldsymbol{c}} I$, we have that $f^K(\boldsymbol{\xi}) = \mathrm{u}$. Then, by definition of L, we have $f^L(\boldsymbol{\xi}) = \textsc{none}$. Thus we have $L \models gr_J(\bigwedge_{f\in\boldsymbol{c}} \forall \boldsymbol{x}\langle f(\boldsymbol{x})\rangle^{\underline{J}}$.

We now show by induction on G that $K \models_p gr_I[G]^{\underline{I}}$ iff $L \models gr_J(G^{none})^{\underline{J}}$ and then since we assume $K \models_p gr_I[G]^{\underline{I}}$, we will conclude that $L \models gr_J(G^{none})^{\underline{J}}$. Finally, since we have already seen that $L \models gr_J(\bigwedge_{f\in\boldsymbol{c}} \forall \boldsymbol{x}\langle f(\boldsymbol{x})\rangle^{\underline{J}}$, we will concluded further that $L \models gr_J(G^{none} \wedge \bigwedge_{f\in\boldsymbol{c}} \forall \boldsymbol{x}\langle f(\boldsymbol{x})\rangle^{\underline{J}}$

- Case 1: G is a (σ-plain) ground atomic formula of extended signature σ^* which is σ extended with object names from $|I|$ (not including $\textsc{none}$).

  If $I \not\models_p G$, then $J \not\models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and $gr_J(G^{none})^{\underline{J}}$ are both $\perp$, which neither L nor K satisfy so in this case, the claim holds.

  If instead $I \models_p G$, then $J \models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and

$gr_J(G^{none})^{\underline{J}}$ are both G, so by definition of K, $K \models_p G$ iff $L \models G$.

- Case 2: G is $H_1 \odot H_2$ where $\odot \in \{\wedge, \vee, \rightarrow\}$ and so $gr_I[G]^{\underline{I}}$ is $gr_I[H_1]^{\underline{I}} \odot gr_I[H_2]^{\underline{I}}$. G^{none} is $(H_1)^{none} \odot (H_2)^{none}$.

  If $I \not\models_p G$, then $J \not\models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and $gr_J(G^{none})^{\underline{J}}$ are both $\perp$, which neither L nor K satisfy so in this case, the claim holds.

  On the other hand, if $I \models_p G$, then $J \models G^{none}$ by Lemma 43 and so $gr_J[G^{none}]^{\underline{J}}$ is

$$gr_J[(H_1)^{none}]^{\underline{J}} \odot gr_J[(H_2)^{none}]^{\underline{J}}.$$

  By I.H. on $gr_I[H_1]^{\underline{I}}$ and $gr_J[(H_1)^{none}]^{\underline{J}}$ and $gr_I[H_2]^{\underline{I}}$ and $gr_J[(H_2)^{none}]^{\underline{J}}$.

- Case 3: G is $\exists x H(x)$. G^{none} is $\exists x(H(x)^{none} \wedge x \neq \textsc{none})$.

  If $I \not\models_p G$, then $J \not\models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and $gr_J(G^{none})^{\underline{J}}$ are both $\perp$, which neither L nor K satisfy so in this case, the claim holds.

  On the other hand, if $I \models_p G$, then $J \models G^{none}$ by Lemma 43 and so $gr_I[G]^{\underline{I}}$ is

$$\{gr_I[H(\xi^\diamond)]^{\underline{I}} : \xi \in |I|\}^\vee$$

  and $gr_J[G^{none}]^{\underline{J}}$ is

$$\{gr_J[H(\xi^\diamond)^{none}]^{\underline{J}} \wedge \xi^\diamond \neq \textsc{none} : \xi \in |J|\}^\vee.$$

  Now we note that $|J| = |I| \cup \{\textsc{none}\}$. Further, we note that since $\xi^\diamond \neq \textsc{none}$ is not satisfied when $\xi = \textsc{none}$, the latter reduct is equivalent to

$$\{gr_J[H(\xi^\diamond)^{none}]^{\underline{J}} \wedge \xi^\diamond \neq \textsc{none} : \xi \in |I|\}^\vee.$$

The further, we note that for all $\xi \in |I|$, $\xi^\diamond \neq \textsc{none}$ is satisfied so that this is further equivalent to

$$\{gr_J[H(\xi^\diamond)^{none}]^{\underline{J}} : \xi \in |I|\}^\vee.$$

Thus the claim follows by induction on $gr_I[H(\xi^\diamond)]$ and $gr_J[H(\xi^\diamond)^{none}]$ for every $\xi \in |I|$.

- Case 4: G is $\forall x H(x)$. G^{none} is $\forall x(x \neq \textsc{none} \to H(x)^{none})$.

  If $I \not\models_p G$, then $J \not\models G^{none}$ by Lemma 43 and so the reducts $gr_I[G]^{\underline{I}}$ and $gr_J(G^{none})^{\underline{J}}$ are both $\bot$, which neither L nor K satisfy so in this case, the claim holds.

  On the other hand, if $I \models_p G$, then $J \models G^{none}$ by Lemma 43 and so $gr_I[G]^{\underline{I}}$ is

  $$\{gr_I[H(\xi^\diamond)]^{\underline{I}} : \xi \in |I|\}^\wedge$$

  and $gr_J[H^{none}]^{\underline{J}}$ is

  $$\{\xi^\diamond \neq \textsc{none} \to gr_J[H(\xi^\diamond)^{none}]^{\underline{J}} : \xi \in |J|\}^\wedge.$$

  Now we note that $|J| = |I| \cup \{\textsc{none}\}$. Further, we note that since $\xi^\diamond \neq \textsc{none}$ is not satisfied when $\xi = \textsc{none}$, the latter reduct is equivalent to

  $$\{\xi^\diamond \neq \textsc{none} \to gr_J[H(\xi^\diamond)^{none}]^{\underline{J}} : \xi \in |I|\}^\wedge.$$

  The further, we note that for all $\xi \in |I|$, $\xi^\diamond \neq \textsc{none}$ is satisfied so that this is further equivalent to

  $$\{gr_J[H(\xi^\diamond)^{none}]^{\underline{J}} : \xi \in |I|\}^\wedge.$$

  Thus the claim follows by induction on $gr_I[H(\xi^\diamond)]$ and $gr_J[H(\xi^\diamond)^{none}]$ for every $\xi \in |I|$.

∎

10.7.11 Proof of Corollary 8

Corollary 8 *For any $\boldsymbol{c}$-plain sentence F and any partial interpretation I that satisfies $\exists xy(x \neq y)$, I is a stable model of F relative to $\boldsymbol{c}$ iff I is a Cabalar stable model of $F \wedge \neg\neg \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x}(f(\boldsymbol{x}) = f(\boldsymbol{x}))$ relative to $\boldsymbol{c}$.*

Proof. First note that $I \models \neg\neg \bigwedge_{f \in \boldsymbol{c}} \forall \boldsymbol{x}(f(\boldsymbol{x}) = f(\boldsymbol{x}))$ iff I is complete. Then, the claim follows by Theorem 5 from Bartholomew and Lee (2013b).

Chapter 11

OTHER RELATED WORK

11.1 Loose Integrations with other Declarative Paradigms

We first examine several formalisms that loosely integrate answer set programming with declarative paradigms that view functions as in classical logic. While these approaches address the grounding bottleneck present in answer set programming, they do not address the inability to perform defeasible reasoning on functions. We provide explicit relationships between these formalisms and ASPMT. Each is effectively a special case of ASPMT where none of the functions are seen as intensional.

11.1.1 Clingcon

A *constraint satisfaction problem* (CSP) is a tuple (V, D, C), where V is a set of *constraint variables* with the respective *domains* D, and C is a set of *constraints* that specify legal assignments of values in the domains to the constraint variables.

A *clingcon program* Π with a constraint satisfaction problem (V, D, C) is a set of rules of the form

$$a \leftarrow B, N, Cn, \tag{11.1}$$

where a is a propositional atom or $\bot$, B is a set of positive propositional literals, N is a set of negative propositional literals, and Cn is a set of constraints from C, possibly preceded by not.

Clingcon programs can be viewed as ASPMT instances. Below is a reformulation of the semantics in terms of ASPMT. We assume that constraints are expressed by ASPMT sentences of signature $V \cup \sigma^{bg}$, where V is a set of object constants identified with constraint variables V in (V, D, C), whose value sorts are identified with domains in D; we assume that σ^{bg} is disjoint from V and contains all values in D as object constants, and other symbols to represent constraints, such as $+$, $\times$, and $\geq$. In other words, we represent a constraint as a formula $F(v_1, \dots, v_n)$ over $V \cup \sigma^{bg}$ where $F(x_1, \dots, x_n)$ is a formula of the signature σ^{bg} and $F(v_1, \dots, v_n)$ is obtained from $F(x_1, \dots, x_n)$ by substituting the object constants $(v_1, \dots, v_n)$ in V for $(x_1, \dots, x_n)$.

For any signature σ that consists of object constants and propositional constants, we identify an interpretation I of σ as the tuple $\langle I^f, X\rangle$, where I^f is the restriction of I on the object constants in σ, and X is a set of propositional constants in σ that are true under I.

Given a clingcon program Π with (V, D, C), and an interpretation $I = \langle I^f, X\rangle$, we define the *constraint reduct of* Π *relative to* X *and* I^f (denoted by $\Pi^X_{I^f}$) as the set of rules $a \leftarrow B$ for each rule (11.1) is in Π such that $I^f \models_{bg} Cn$, and $X \models N$. We say that a set X of propositional atoms is a *constraint answer set* of Π relative to I^f if X is a minimal model of $\Pi^X_{I^f}$.

Example 7 continued *Recall the leaking bucket example. The rules*

$$(amount_1 = Y) \vee \neg(amount_1 = Y) \;\leftarrow\; amount_0 = Y+1$$
$$amount_1 = 10 \;\leftarrow\; fillUp$$

are identified with

$$\bot \leftarrow not\ FillUp, not(Amount_1 + 1 =^{\$} Amount_0)$$
$$\bot \leftarrow FillUp, not(Amount_1 =^{\$} 10)$$

under the semantics of clingcon programs. Consider I in Example 7, which can be represented as $\langle I^f, X\rangle$ where I^f maps $Amount_0$ to 6, and $Amount_1$ to 5, and $X = \emptyset$. X is the constraint answer set relative to I^f because X is the minimal model of the constraint reduct relative to X and I^f, which is the empty set.

Similar to the way that rules are identified as a special case of formulas Ferraris *et al.* (2011), we identify a clingcon program Π with the conjunction of implications $B \wedge N \wedge Cn \rightarrow a$ for all rules (11.1) in Π. The following theorem tells us that clingcon programs are a special case of ASPMT, in which the background theory is specified by (V, D, C), and intensional constants are limited to propositional constants only, and do not allow function constants.

Theorem 29 *Let Π be a clingcon program with CSP (V, D, C), let $\boldsymbol{p}$ be the set of all propositional constants occurring in Π, and let I be an interpretation $\langle I^f, X\rangle$ of signature $V \cup \boldsymbol{p}$. Set X is a constraint answer set of Π relative to I^f iff $I \models_{bg} SM[\Pi; \boldsymbol{p}]$.*

Note that a clingcon program does not allow an atom that consists of elements from both V and $\boldsymbol{p}$. Thus the truth value of any atom is determined by either I^f or X, but not by involving both of them. This allows loose coupling of an ASP solver and a constraint solver. On the other hand, Gebser *et al.* (2009a) sketches a method to extend clingcon programs to allow predicate constants of positive arity, possibly containing constraint variables as arguments. This however leads to some unintuitive cases under the semantics of CLINGCON programs, as the following example shows.

```
$domain(100..199).   % Office numbers
myoffice(a).         % a is my office number,
:- myoffice(b).      % and b is not.
:- not a $==b.       % Nevertheless, a equals b.
```

System CLINGCON does not notice that this set of assumptions is inconsistent. This is because symbols a and b in ASP atoms and the same symbols in the constraint are not related. On the other hand, ASPMT, which allows first-order signatures, does not have this anomaly; there is no stable model under ASPMT.

11.1.2 Lin and Wang's Logic Programs with Functions

11.1.3 Lin-Wang Programs

Lin and Wang (2008) extended answer set semantics with functions by extending the definition of a reduct, and also provided loop formulas for such programs. We can provide an alternative account of their results by considering the notions there as special cases of the definitions presented in this paper. For simplicity, we assume non-sorted languages. [1] Essentially, they restricted attention to a special case of non-Herbrand interpretations such that object constants form the universe, and ground terms other than object constants are mapped to object constants. According to Lin and Wang (2008), an *LW-program P* consists of *type definitions* and a set of rules of the form

$$A \leftarrow B_1, \dots, B_m, not\ C_1, \dots, not\ C_n \tag{11.2}$$

where A is $\bot$ or an atom and B_i, where $1 \leq i \leq m$ and C_j, where $1 \leq j \leq n$ are atomic formulas.

Type definitions introduce the domains for a many-sorted signature consisting of

[1] Lin and Wang (2008) considers essentially many-sorted languages. The result of this section can be extended to that case by considering many-sorted SM Kim *et al.* (2009).

some object constants, and includes the evaluation of each function symbol of positive arity that maps a list of object constants to an object constant. Since we assume here non-sorted languages, we consider only a single domain (universe). We say that an interpretation I is a *P-interpretation* if the universe is the set of object constants specified by P, object constants are evaluated to itself, and ground terms other than object constants are evaluated conforming to the type definitions of P.

The *reduct of a ground program P with respect to a P-interpretation I is denoted P^I and is obtained from P by*

1 replacing each functional term $f(t_1, \ldots, t_n)$ with c where $f^I(t_1, \ldots, t_n) = c$;

2 removing any rule with an atomic formula B_i that contains an equality and is not satisfied by I;

3 removing any remaining equalities from the remaining rules;

4 removing any rule containing not A in the body of the rule where $A^I = \boldsymbol{t}$;

5 removing any remaining conjunctive term form not A.

A P-interpretation I is an answer set of P in the sense of Lin and Wang (2008) if I satisfies every rule in P and the set of atoms in I is precisely the set of atoms in the minimal model of P^I.

Theorem 30 *Let P be an LW-program and let F be the FOL-representation of the set of rules in P. The following conditions are equivalent to each other:*

(a) I is an answer set of P in the sense of Lin and Wang (2008);

(b) I is a P-interpretation that satisfies SM$[F; \boldsymbol{p}]$ where $\boldsymbol{p}$ is the list of all predicate constants occurring in F.

Thus the definition does not allow functions to be intensional.

11.1.4 ASP(LC) Programs

Liu *et al.* (2012) considers logic programs with linear constraints, or *ASP(LC)* programs, comprised of rules of the form

$$a \leftarrow B, N, LC \tag{11.3}$$

where a is a propositional atom or $\bot$, B is a set of positive propositional literals, and N is a set of negative propositional literals, and LC is a set of *theory atoms*—linear constraints of the form $\Sigma_{i=1}^{n}(c_i \times x_i) \ \bowtie \ k$ where $\bowtie \in \{\leq, \geq, =\}$, each x_i is an object constant whose value sort is integers (or reals), and each c_i, k is an integer (or real).

An ASP(LC) program Π can be viewed as an ASPMT formula whose background theory bg is the theory of integers or the theory of reals. Let σ^p denote the set of all propositional atoms occurring in Π and σ^f denote all object constants occurring in Π that do not belong to the background signature. Theory atoms are essentially ASPMT formulas of signature $\sigma^f \cup \sigma^{bg}$. We identify ASP(LC) program Π with the conjunction of ASPMT formulas $B \wedge N \wedge LC \rightarrow a$ for all rules (11.3) in Π.

An *LJN-intepretation* is a pair (X, T) where $X \subseteq \sigma^p$ and T is a subset of theory atoms occurring in Π such that there is some interpretation I of signature σ^f such that $I \models_{bg} T \cup \overline{T}$, where $\overline{T}$ is the set of negations of each theory atom occurring in Π but not in T. An LJN-interpretation (X, T) satisfies an atom b if $b \in X$, the negation of an atom *not* c if $c \notin X$, and a theory atom t if $t \in T$. The notion of satisfaction is extended to other propositional connectives as usual.

The *LJN-reduct* of a program Π with respect to an LJN-interpretation (X, T), denoted by $\Pi^{(X,T)}$, consists of rules $a \leftarrow B$ for each rule (11.3) such that (X, T) satisfies $N \wedge LC$. (X, T) is an *LJN-answer set* of Π if (X, T) satisfies Π, and X is the smallest set of atoms satisfying $\Pi^{(X,T)}$.

The following theorem tells us that there is a one-to-many relationship between LJN-answer sets and the stable models in the sense of ASPMT.

Theorem 31 *Let Π be an ASP(LC) program, and σ^p and σ^f are defined as above.*

(a) If (X, T) is an LJN-answer set of Π, then for any interpretation $\langle I^f, X\rangle$ of signature $\sigma^p \cup \sigma^f$ such that $I^f \models_{bg} T \cup \overline{T}$, we have $\langle I^f, X\rangle \models_{bg} SM[\Pi; \sigma^p]$.

(b) For any interpretation $I = \langle I^f, X\rangle$ of signature $\sigma^p \cup \sigma^f$, if $\langle I^f, X\rangle \models_{bg} SM[\Pi; \sigma^p]$, then an LJN-interpretation (X, T) where

$$T = \{t \mid t \textit{ is a theory atom in } \Pi \textit{ such that } I^f \models_{bg} t\}$$

is an LJN-answer set of Π.

Example 32 *Let F be*

$$a \leftarrow x - z > 0. \qquad b \leftarrow x - y \leq 0.$$
$$c \leftarrow b,\ y - z \leq 0. \quad \leftarrow not\ a.$$
$$b \leftarrow c.$$

The LJN-interpretation $L = \langle \{a\}, \{x - z > 0\}\rangle$ is an answer set of F since $\{(x - z > 0, \neg(x - y \leq 0), \neg(y - z \leq 0)\}$ is satisfiable (e.g. take $x^I = 2, y^I = 1, z^I = 0$) and the set $\{a\}$ is the minimal model satisfying the reduct $F^L = (\top \rightarrow a) \wedge c \rightarrow b$. On the other hand the interpretation I such that $x^I = 2, y^I = 1, z^I = 0, a^I = \boldsymbol{t}, b^I = \boldsymbol{f}, c^I = \boldsymbol{f}$ satisfies $I \models_{bg} SM[F; abc]$.

As with clingcon programs, ASP(LC) programs are more restrictive than ASPMT. ASP(LC) programs do not allow theory atoms in the head of a rule, and like clingcon programs, cannot express intensional functions.

11.2 Other Approaches to Intensional Functions

In Chapter 10, we explored the relationship between our semantics and the Cabalar semantics in depth. In this section, we present three other definitions of intensional functions–Nonmonotonic Causal Logic from Giunchiglia *et al.* (2004), IF-Programs from Lifschitz (2012), and the Balduccini semantics Balduccini (2012). We show that IF-Programs exhibit some undesirable characteristics not present in the either the stable model semantics or the functional stable model semantics. We then show that the Balduccini semantics is essentially a special case of the Cabalar semantics.

11.2.1 Relation to Nonmonotonic Causal Logic

A *(nonmonotonic) causal theory* is a finite list of rules of the form

$$F \Leftarrow G$$

where F and G are formulas. We identify a rule with the universal closure of the implication $G \rightarrow F$. A *causal model* of a causal theory T is defined as the models of the second-order sentence

$$\mathrm{CM}[T; \boldsymbol{f}] = T \wedge \neg\exists \widehat{\boldsymbol{f}}(\widehat{\boldsymbol{f}} \neq \boldsymbol{f} \wedge T^{\dagger}(\widehat{\boldsymbol{f}}))$$

where $\boldsymbol{f}$ is a list of *explainable* function constants, and $T^{\dagger}(\widehat{\boldsymbol{f}})$ denotes the conjunction of the formulas

$$\widetilde{\forall}(G \rightarrow F(\widehat{\boldsymbol{f}})) \tag{11.4}$$

for all rules $F \Leftarrow G$ of T.

By a *definite* casual theory, we mean the causal theory whose rules have the form either

$$f(\boldsymbol{t}) = t_1 \Leftarrow B \tag{11.5}$$

or

$$\bot \Leftarrow B, \tag{11.6}$$

where f is an explainable function constant, $\boldsymbol{t}$ is a list of terms that does not contain explainable function constants, and t_1 is a term that does not contain explainable function constants. By $Tr(T)$ we denote the theory consisting of conjunction of the following formulas: $\widetilde{\forall}(\neg\neg B \rightarrow f(\boldsymbol{t}) = t_1)$ for each rule (11.5) in T, and $\widetilde{\forall}\neg B$ for each rule (11.6) in T. The causal models of such T coincide with the stable models of $Tr(T)$.

Theorem 32 *For any definite causal theory T, $I \models CM[T; \boldsymbol{f}]$ iff $I \models SM[Tr(T); \boldsymbol{f}]$.*

For non-definite ones, they do not coincide.

Example 33 *Consider the following non-definite causal theory T:*

$$\neg(f = 1) \Leftarrow \top$$
$$\neg(f = 2) \Leftarrow \top$$

An interpretation I where $|I| = \{1, 2, 3\}$ and $f^I = 3$ is a causal model of T. However, the corresponding formula $Tr(T)$ is equivalent to

$$\neg(f = 1) \wedge \neg(f = 2)$$

which has no stable models.

Reduct-based Characterization of IF-Programs

We first present a reformulation of IF-Programs in terms of reduct.

For any ground formula F, $F^{\underline{I}}$ is a formula obtained from F by replacing every maximal negated formula $\neg G$ with

- $\top$ if $I \models \neg G$, and
- $\bot$ if $I \not\models \neg G$.

Let Π be a ground IF-program. The IF-reduct $\Pi^{\underline{I}}$ of an IF-program Π relative to an interpretation I consists of rules

$$H^{\underline{I}} \leftarrow B^{\underline{I}}$$

for every rule $H \leftarrow B$ in Π.

Theorem 33 *Let F be the FOL-representation of a ground IF-program of signature σ and let $\boldsymbol{f}$ be a list of intensional function constants. For any interpretation I of σ, $I \models \mathit{IF}[F; \boldsymbol{f}]$ iff*

- *I satisfies Π, and*
- *no interpretation J of σ that disagrees with I only on $\boldsymbol{f}$ satisfies $\Pi^{\underline{I}}$.*

Comparison

The definition of the IF operator above looks close to our definition of the SM operator. However, they often behave quite differently. Neither semantics is stronger than the other.

Example 34 *Let F be the following program*

$$d = 2 \leftarrow c = 1,$$
$$d = 1$$

and let I be an interpretation such that $|I| = \{1, 2\}$, $I(c) = 2$ and $I(d) = 1$. I is a model of IF$[F; cd]$, but not a model of SM$[F; cd]$.

Example 35 *Let F be the following program*

$$(c = 1 \vee d = 1) \wedge (c = 2 \vee d = 2)$$

and let I_1 and I_2 be interpretations such that $|I_1| = |I_2| = \{1, 2, 3\}$ and $I_1(c) = 1$, $I_1(d) = 2$, $I_2(c) = 2$, $I_2(d) = 1$. I_1 and I_2 are models of SM$[F; cd]$. On the other hand, IF$[F; cd]$ has no models.

Example 36 *Let F be $c \neq 1 \leftarrow \top$ and let F_1 be $\bot \leftarrow c = 1$. Under our semantics, they are strongly equivalent to each other, and neither of them has a stable model. However, this is not the case with IF-programs. For instance, let I be an interpretation such that $|I| = \{1, 2\}$ and $I(c) = 2$. I satisfies IF$[F_1; c]$ but not IF$[F; c]$.*

While $\bot \leftarrow F$ is a constraint in our formalism, in view of Theorem 3, the last example illustrates that $\bot \leftarrow F$ is not considered as a constraint in the semantics of IF-programs. Indeed, the definition of a constraint given in Lifschitz (2012) is stronger than ours.

Example 36 illustrates that a model of an IF-program may map a function to a value that does not even occur in the program. For the stable model semantics, syntactic conditions on variables ensure that the universe has no impact on the stable models whereas for IF-programs, even variable-free programs may have different stable models for different universes.

Let T be an IF-program whose rules have the form

$$f(\boldsymbol{t}) = t_1 \leftarrow \neg\neg B \tag{11.7}$$

where f is an intensional function constant, $\boldsymbol{t}$ and t_1 do not contain intensional function constants, and B is an arbitrary formula. We identify T with the corresponding first-order formula.

Theorem 34 $I \models \mathit{SM}[T; \boldsymbol{f}]$ *iff* $I \models \mathit{IF}[T; \boldsymbol{f}]$.

11.2.3 Balducccini Semantics

Relationship to the Cabalar Semantics

It turns out that the Balduccini semantics presented in Section 3.7 is closely related to the Cabalar semantics. This is shown by reformulating the Balduccini semantics using the notion of partial interpretations and partial satisfaction. We identify a consistent set of seed literals I with a partial interpretation that maps all object constants in $\sigma \setminus \boldsymbol{c}$ to themselves. For example, for signature $\sigma = \{f, g, 1, 2\}$ where $f, g \in \boldsymbol{c}$, we identify the consistent set of seed literals $I = \{f = 1\}$ with the partial interpretation I such that $f^I = 1, g^I = u, 1^I = 1, 2^I = 2$.

The following theorem states that, in the absence of strong negation, Balduccini semantics can be viewed as a special, ground case of the Cabalar semantics.

Theorem 35 *For any ASP{f} program* Π *with intensional constants* $\boldsymbol{c}$ *and any consistent set* I *of seed literals, if* Π *contains no strong negation, then* I *is a Balduccini answer set of* Π *iff* $I \models_p CBL[\Pi; \boldsymbol{c}]$.

Theorem 35 can be extended to full ASP{f} programs that contain strong negation. Since the language in Cabalar (2011) does not allow strong negation, this requires us to eliminate strong negation. It is well known that strong negation in front of standard atoms can be eliminated using new atoms.

In order to eliminate strong negation in front of t-atoms, by $\Pi^{\#}$ we denote the program obtained from Π by replacing $\sim(f = g)$ with $(f = f) \wedge (g = g) \wedge \neg(f = g)$. As we noted earlier, this formula is true iff f^I and g^I are defined, and have different values. This is the same understanding as the construct $f \# g$ in Cabalar (2011).

Theorem 36 *For any ASP{f} program* Π *with intensional constants* $\boldsymbol{c}$ *and any consistent set* I *of seed literals,* I *is a Balduccini answer set of* Π *iff* I *is a Balduccini answer set of* $\Pi^{\#}$.

11.3.1 Proof of Theorem 29

Theorem 29 *Let Π be a clingcon program with CSP (V, D, C), let $\boldsymbol{p}$ be the set of all propositional constants occurring in Π, and let I be an interpretation $\langle I^f, X\rangle$ of signature $V \cup \boldsymbol{p}$. Set X is a constraint answer set of Π relative to I^f iff $I \models_{bg}$ SM$[\Pi; \boldsymbol{p}]$.*

Proof.

We wish to show that X is a constraint answer set of Π relative to I^f iff $\langle I^f, X\rangle \models_{bg}$ SM$[\Pi; \boldsymbol{p}]$. That is, we wish to show that there is no set Y that is smaller than X such that $Y \models \Pi^X_{I^f}$ iff $\langle I^f, X\rangle \models_{bg} \Pi \wedge \neg\exists\widehat{\boldsymbol{p}}(\widehat{\boldsymbol{p}} < \boldsymbol{p} \wedge (\Pi^{FOL})^*(\widehat{\boldsymbol{p}}))$. In the case that $\langle I^f, X\rangle \not\models_{bg} \Pi$, we have that $\Pi^X_{I^f}$ is equivalent to $\bot$ and so for this case, the claim holds. Thus, we only need to show that in the case that $\langle I^f, X\rangle \models_{bg} \Pi$, there is no set Y that is smaller than X such that $Y \models \Pi^X_{I^f}$ iff $\langle I^f, X\rangle \models_{bg} \neg\exists\widehat{\boldsymbol{p}}(\widehat{\boldsymbol{p}} < \boldsymbol{p} \wedge (\Pi^{FOL})^*(\widehat{\boldsymbol{p}}))$.

Equivalently, we can show that there is a set Y that is smaller than X such that $Y \models \Pi^X_{I^f}$ iff $\langle I^f, X\rangle \models_{bg} \exists\widehat{\boldsymbol{p}}(\widehat{\boldsymbol{p}} < \boldsymbol{p} \wedge (\Pi^{FOL})^*(\widehat{\boldsymbol{p}}))$.

($\Rightarrow$) Assume that there is a set Y that is smaller than X such that $Y \models \Pi^X_{I^f}$. We wish to show that $\langle I^f, X\rangle \models_{bg} \exists\widehat{\boldsymbol{p}}(\widehat{\boldsymbol{p}} < \boldsymbol{p} \wedge (\Pi^{FOL})^*(\widehat{\boldsymbol{p}}))$. We will consider another interpretation of the a signature σ' that extends $\sigma = V \cup \boldsymbol{p}$ with a list $\boldsymbol{q}$ that is the same length as $\boldsymbol{p}$. This interpretation will be $I' = \langle I^f, Y \cup Y^{\boldsymbol{p}}_{\boldsymbol{q}}\rangle$ where $Y^{\boldsymbol{p}}_{\boldsymbol{q}}$ denotes the interpretation obtained from Y by replacing each $p \in \boldsymbol{p}$ with the corresponding $q \in \boldsymbol{q}$. Then, wish to show that $I' \models_{bg} (\boldsymbol{q} < \boldsymbol{p} \wedge (\Pi^{FOL})^*(\boldsymbol{q}))$.

We first verify that $I' \models_{bg} \boldsymbol{q} < \boldsymbol{p}$. Since we assume Y that is smaller than X, it must be the case that for every $p \in \boldsymbol{p}$, $p^Y \subseteq p^X$ and there is some $p \in \boldsymbol{p}$ such that $p^Y \subsetneq p^X$. That is, there is some tuple $\boldsymbol{\xi}$ from $|\langle I^f, X\rangle|$ such that $p(\boldsymbol{\xi}) \in X$ and $p(\boldsymbol{\xi}) \notin Y$. It is immediate from the definition of q and the fact that for every $p \in \boldsymbol{p}$,

$p^Y \subseteq p^X$ that $I' \models_{bg} \models \boldsymbol{q} \leq \boldsymbol{p}$. Then, since $p(\boldsymbol{\xi}) \in X$ and $p(\boldsymbol{\xi}) \notin Y$, by definition of q, we have that $I' \models_{bg} q < p$ and so we conclude that $I' \models_{bg} \boldsymbol{q} < \boldsymbol{p}$.

We now show that $I' \models_{bg} (\Pi^{FOL})^*(\boldsymbol{q})$. Since Π is a conjunction of implications of the form $a \leftarrow B, N, Cn$, we simply need to show this to be the case for any such implication. We consider the possible cases.

- $I^f \not\models_{bg} Cn$. In this case the rule has no corresponding presence in the reduct $\Pi^X_{I^f}$ so certainly Y satisfies this part of the reduct. In this case $((a \leftarrow B, N, Cn)^{FOL})^*(\boldsymbol{q})$ is equivalent to $a^*(\boldsymbol{q}) \leftarrow B^*(\boldsymbol{q}) \wedge N \wedge Cn$ Now, since $I^f \not\models_{bg} Cn$, it follows that $I' \not\models_{bg} Cn$ and so I' trivially satisifies the implication $((a \leftarrow B, N, Cn)^{FOL})^*(\boldsymbol{q})$ and so in this case, the claim holds.

- $X \not\models_{bg} N$. In this case the rule has no corresponding presence in the reduct $\Pi^X_{I^f}$ so certainly Y satisfies this part of the reduct. In this case $((a \leftarrow B, N, Cn)^{FOL})^*(\boldsymbol{q})$ is equivalent to $a^*(\boldsymbol{q}) \leftarrow B^*(\boldsymbol{q}) \wedge N \wedge Cn$ Now, since $X \not\models_{bg} N$, it follows that $I' \not\models_{bg} N$ and so I' trivially satisifies the implication $((a \leftarrow B, N, Cn)^{FOL})^*(\boldsymbol{q})$ and so in this case, the claim holds.

- $I^f \models_{bg} Cn$, $X \models_{bg} N$, and $X \not\models_{bg} B$. In this case, the corresponding implication in the reduct $\Pi^X_{I^f}$ is $a \leftarrow B$. Now since B is a conjunction of propositional literals, and Y is a subset of X, it follows that $Y \not\models_{bg} B$ and so Y trivially satisfies the implication. In this case $((a \leftarrow B, N, Cn)^{FOL})^*(\boldsymbol{q})$ is equivalent to $a^*(\boldsymbol{q}) \leftarrow B^*(\boldsymbol{q}) \wedge N \wedge Cn$. Thus, since $I' \models_{bg} B^*(\boldsymbol{q})$ iff $Y \models_{bg} B$, I' trivially satisifies the implication $((a \leftarrow B, N, Cn)^{FOL})^*(\boldsymbol{q})$ and so in this case, the claim holds.

- $I^f \models_{bg} Cn$, $X \models_{bg} N$, $X \models_{bg} B$, and $X \not\models_{bg} a$. In this case, the corresponding implication in the reduct $\Pi^X_{I^f}$ is $a \leftarrow B$. Now since a is a propositional literal or $\bot$, and Y is a subset of X, it follows that $Y \not\models_{bg} a$ and so Y satisfies the implication iff $Y \models_{bg} B$. In this case $((a \leftarrow B, N, Cn)^{FOL})^*(\boldsymbol{q})$ is equivalent to $a^*(\boldsymbol{q}) \leftarrow B^*(\boldsymbol{q}) \wedge N \wedge Cn$. Now $I' \models_{bg} B^*(\boldsymbol{q})$ iff $Y \models_{bg} B$ and $I' \models_{bg} a^*(\boldsymbol{q})$ iff $Y \models_{bg} a$, the claim immediately follows.

- $I^f \models_{bg} Cn$, $X \models_{bg} N$, $X \models_{bg} B$, and $X \models_{bg} a$. In this case, the corresponding implication in the reduct $\Pi^X_{I^f}$ is $a \leftarrow B$. In this case, Y satisfies the implication iff $Y \not\models_{bg} B$ or $Y \models_{bg} a$. In this case $((a \leftarrow B, N, Cn)^{FOL})^*(\boldsymbol{q})$ is equivalent to $a^*(\boldsymbol{q}) \leftarrow B^*(\boldsymbol{q}) \wedge N \wedge Cn$. Now $I' \models_{bg} B^*(\boldsymbol{q})$ iff $Y \models_{bg} B$ and $I' \models_{bg} a^*(\boldsymbol{q})$ iff $Y \models_{bg} a$, the claim immediately follows.

($\Leftarrow$) Assume that $\langle I^f, X\rangle \models_{bg} \exists \widehat{\boldsymbol{p}}(\widehat{\boldsymbol{p}} < \boldsymbol{p} \wedge (\Pi^{FOL})^*(\widehat{\boldsymbol{p}}))$. That is, there is some interpretation I' of the extended signature $\sigma' = \sigma \cup \boldsymbol{q}$ such that $I' \models_{bg} (\boldsymbol{q} < \boldsymbol{p} \wedge (\Pi^{FOL})^*(\boldsymbol{q}))$. Let this signature be $\langle I^f, X \cup Y$ where I^f is the interpretation of functions from σ', X is the interpretation of propositional literals in $\boldsymbol{p}$ and Y is the interpretation of propositional literals in $\boldsymbol{q}$. We will show that Y is smaller than X and $Y \models \Pi^X_{I^f}$.

We first show that $Z = Y^{\boldsymbol{q}}_{\boldsymbol{p}}$ is smaller than than X. We assume that $I' \models_{bg} \boldsymbol{q} < \boldsymbol{p}$. From this it is immediate that for any $q \in \boldsymbol{q}$, $q^{I'}$ is a subset of $p^{I'}$ for the corresponding $p \in \boldsymbol{p}$ and so Z is not a superset of X. Then, since $I' \models_{bg} \boldsymbol{q} < \boldsymbol{p}$ there must be some $q \in \boldsymbol{q}$ and some tuple in $|I'|$ such that $q(\boldsymbol{\xi})^{I'} = 0$ but $p(\boldsymbol{\xi})^{I'} = 1$ for the corresponding $p \in \boldsymbol{p}$. From this, it follows that Z is strictly smaller than X.

We now show that $Z \models_{bg} \Pi^X_{I^f}$. Since Π is a conjunction of implications of the form $a \leftarrow B, N, Cn$, we simply need to show this to be the case for any such implication. We consider the possible cases.

- $I^f \not\models_{bg} Cn$. In this case the rule has no corresponding presence in the reduct

$\Pi^X_{I^f}$ so certainly Z satisfies this part of the reduct.

- $X \not\models_{bg} N$. In this case the rule has no corresponding presence in the reduct $\Pi^X_{I^f}$ so certainly Z satisfies this part of the reduct.

- $I^f \models_{bg} Cn$ and $X \models_{bg} N$. In this case, the corresponding implication in the reduct $\Pi^X_{I^f}$ is $a \leftarrow B$. In this case $((a \leftarrow B, N, Cn)^{FOL})^*(\boldsymbol{q})$ is equivalent to $a^*\boldsymbol{q} \leftarrow B^*(\boldsymbol{q}) \wedge N \wedge Cn$. Since we assume $I^f \models_{bg} Cn$, $X \models_{bg} N$, I' satisfies the implication iff $I' \models_{bg} a^*(\boldsymbol{q}) \leftarrow B^*(\boldsymbol{q})$. Now, since $I' \models_{bg} B^*(\boldsymbol{q})$ iff $Y \models_{bg} B$ and $I' \models_{bg} a^*(\boldsymbol{q})$ iff $Y \models_{bg} a$, the claim immediately follows. ∎

11.3.2 Proof of Theorem 30

Lemma 44 *Let P be a LW-program, F be the first-order representation of P, I be a P-interpretation of the signature σ of P such that I is a model of P. and $\boldsymbol{p}$ be the list of all predicates in σ. For any interpretation J such that $J <^{\boldsymbol{p}} I$ and any set of atoms K such that for any atom A, we have $A \in K$ iff $A^J = 1$, then $J \models F^{\underline{I}}$ iff K satisfies P^I.*

Proof.

Since F is comprised of a conjunctions of the form

$$B_1 \wedge \cdots \wedge B_m \wedge (\neg\, C_1) \wedge \cdots \wedge (\neg\, C_n) \rightarrow A$$

$F^{\underline{I}}$ is by definition a conjunction of

$$(B_1 \wedge \cdots \wedge B_m \wedge (\neg\, C_1) \wedge \cdots \wedge (\neg\, C_n) \rightarrow A)^{\underline{I}}$$

and so we will consider each conjunctive subformula G_i of F separately. Each conjunctive term G_i corresponds to a rule r_i in P of the form

$$A \leftarrow B_1, \ldots, B_m, not\ C_1, \ldots, not\ C_n.$$

and so we simply need to show that for any conjunctive subformula G_i of F and the corresponding rule r_i, we have $J \models G_i^{\underline{I}}$ iff $K \models r_i^I$. In the context of this comparison, we note that when a rule is removed in P^I, this is equivalent to replacing the rule with $\top$. We consider the following cases for a rule r and the corresponding conjunctive subformula G in F.

- Case 1: There is some atomic formula B_i in r such that B_i contains an equality and is not satisfied by I.

  In this case r^I is replaced with $\top$ and so $K \models r^I$. In $G^{\underline{I}}$, it may be the case that some $(\neg\, C_k)$ is replaced by $\bot$ but B_i will certainly remain as a conjunctive term in the precedent of the implication G and so since J agrees with I on all functions, we have $J \not\models B_i$ and so $J \models G^{\underline{I}}$.

- Case 2: There are no atomic formulas B_i in r such that B_i contains an equality and is not satisfied by I, but there is a conjunctive term *not* C_k in r such that $C_k^I = 1$.

  In this case r^I is replaced with $\top$ and so $K \models r^I$. In $G^{\underline{I}}$, $(\neg\, C_k)$ will be replaced with $\bot$ and so $J \models G^{\underline{I}}$.

- Case 3: There are no atomic formulas B_i in r such that B_i contains an equality and is not satisfied by I, and there is no conjunctive term *not* C_k in r such that $C_k^I = 1$.

  In this case, r^I is obtained from r by replacing each $f(t_1, \ldots, t_m)$ with c where $f^I(t_1, \ldots, t_m) = c$ and from removing all conjunctive terms containing equality. On the other hand, there are two cases for $G^{\underline{I}}$: is either $\bot$ if $I \not\models G$ or $G^{\underline{I}}$ is precisely G otherwise. However, the assumption that I is a model of P means that the former case cannot arise and so $G^{\underline{I}}$ is precisely G.

Now, since J and I agree on all functions, we have that $J \models G^{\underline{I}}$ iff $J \models H$ where H is obtained from $G^{\underline{I}}$ by replacing $f(t_1, \ldots, t_m)$ with c where $f^I(t_1, \ldots, t_m) = c$. Also since J and I agree on all functions, and since we assumed in this case that I satisfies all conjunctive terms containing equality, we have $J \models H$ iff $J \models H'$ where H' is obtained from H by removing all conjunctive terms containing equality. Now the only remaining difference between H' and r^I is that every remaining conjunctive term *not* A in H' is absent in r^I. Note that since we assumed for this case that there is no conjunctive term *not* C_k in r such that $C_k^I = 1$, it must be that every such conjunctive term is such that $A^I = 0$. However, since we have that $J <^{\boldsymbol{p}} I$, it must be that $A^J = 0$ and so $J \models \neg A$. Thus, $J \models H'$ iff $J \models H''$ where H'' is obtained from H' by removing all conjunctive terms of the form *not* A. Now H'' is exactly the first-order representation of r^I and since J and K agree on all predicates, it is clear that $J \models G^{\underline{I}}$ iff $K \models r^I$. ∎

Theorem 30 *Let P be an LW-program and let F be the FOL-representation of the set of rules in P. The following conditions are equivalent to each other:*

(a) I is an answer set of P in the sense of Lin and Wang (2008);

(b) I is a P-interpretation that satisfies SM$[F; \boldsymbol{p}]$ where $\boldsymbol{p}$ is the list of all predicate constants occurring in F.

Proof. We will use the reduct-based characterization of the SM semantics in this proof. When programs are restricted

($\Rightarrow$) Let us assume I is an answer set of P in the sense of Lin and Wang (2008). We wish to show that I satisfies SM$[F; \boldsymbol{p}]$ where $\boldsymbol{p}$ is the list of all predicate constants occurring in F. That is, we assume I satisfies every rule in P and there is no subset K of atoms in I such that K is a model of P^I and we wish to show that $I \models F$ and no interpretation J such that $J <^{\boldsymbol{p}} I$ satisfies $F^{\underline{I}}$. Since I is an answer set of P, Isatisfies every rule of P and so it immediately follows that $I \models F$. So it only remains to be shown that if there is no subset K of atoms in I such that K is a model of P^I, then there is no interpretation J such that $J <^{\boldsymbol{p}} I$ satisfies $F^{\underline{I}}$. To show this, we will consider the contrapositive; we assume there is some interpretation J such that $J <^{\boldsymbol{p}} I$ satisfies $F^{\underline{I}}$ and will show that there is a subset K of the atoms in I such that K is a model of P^I.

We first note that since $J <^{\boldsymbol{p}} I$, J and I differ only predicates so that J^{pred} is a subset of I^{pred}. Thus, we will take $K = J^{pred}$ so that K is a subset of the atoms in I and show that K is a model of P^I. Then the claim follows by Lemma 44.

($\Leftarrow$) Let us assume I satisfies $\text{SM}[F; \boldsymbol{p}]$ where $\boldsymbol{p}$ is the list of all predicate constants occurring in F. We wish to show that I is an answer set of P in the sense of Lin and Wang (2008). That is, we assume $I \models F$ and no interpretation J such that $J <^{\boldsymbol{p}} I$ satisfies $F^{\underline{I}}$ and we wish to show that I satisfies every rule in P and there is no subset K of atoms in I such that K is a model of P^I. Since we assume $I \models F$, then it follows that Isatisfies every rule of P. So it only remains to be shown that if there is no interpretation J such that $J <^{\boldsymbol{p}} I$ satisfies $F^{\underline{I}}$, then there is no subset K of atoms in I such that K is a model of P^I. To show this, we will consider the contrapositive; we assume there a subset K of the atoms in I such that K is a model of P^I and we will show that there is some interpretation J such that $J <^{\boldsymbol{p}} I$ satisfies $F^{\underline{I}}$.

We will take J such that J and I agree on all functions and such that for any atomic formula A, we have $A^J = 1$ iff $A \in K$. It is clear that since K is a subset of the atoms in I, that $J <^{\boldsymbol{p}} I$. The claim then follows by Lemma 44. ∎

11.3.3 Proof of Theorem 31

Lemma 45 *Given an ASP(LC) program Π, for any LJN interpretation (X, T) and any interpretation $I = \langle I^f, Y\rangle$, the following are equivalent:*

- *(a) $I \models_{bg} T \cup \overline{T}$,*
- *(b) $(X, T) \models t$ iff $I \models_{bg} t$ for every t occurring in Π.*

Proof.

(a) Assume $I \models_{bg} T \cup \overline{T}$.

($\Rightarrow$) Assume $(X, T) \models t$ for some t occurring in Π. This means $t \in T$ and so by the condition on I, $I \models_{bg} t$.

($\Leftarrow$) Assume $I \models_{bg} t$ for some t occurring in Π. By the condition on I, $t \in T$ and so $(X, T) \models t$.

(b) Assume $(X, T) \models t$ iff $I \models_{bg} t$ for every t occurring in Π.

By definition of $(X, T) \models t$, $t \in T$ iff $I \models_{bg} t$ for every t occurring in Π. Thus $I \models_{bg} T$ and $I \models_{bg} \overline{T}$ so $I \models_{bg} T \cup \overline{T}$. ∎

Lemma 46 *Given an ASP(LC) program Π, two LJN-interpretations (X, T) and (Y, T) where $(X, T) \models \Pi$ and $Y \subseteq X$, two interpretations $I = \langle I^f, X\rangle$ and $J = \langle I^f, Y\rangle$ such that*

- *$I \models \Pi$,*
- *$I^f \models_{bg} T \cup \overline{T}$,*

we have $Y \models \Pi^{(X,T)}$ *iff* $J \models \Pi^{\underline{I}}$.

Proof.

($\Rightarrow$) Assume $Y \models \Pi^{(X,T)}$. This means that Y satisfies every rule in the reduct $\Pi^{(X,T)}$. For any rule r of the form (11.3) in Π, there are two cases:

- Case 1: $(X, T) \models N \wedge LC$.

  In this case, the corresponding rule in the reduct $\Pi^{(X,T)}$ is

  $$a \leftarrow B.$$

  On the other hand, $r^{\underline{I}}$ has two cases:

  - Subcase 1: $I \models B$.

    Since we assume $I \models \Pi$, it must be that $I \models a$. By Lemma 45, since $(X, T) \models t$ for all t in LC, so too does I and so $I \models LC$. In this case, $r^{\underline{I}}$ is

    $$a \leftarrow B, \top, \ldots, \top, LC^{\underline{I}}.$$

    Since I and J interpret object constants in the same way and $I \models LC^{\underline{I}}$, $J \models LC^{\underline{I}}$. Thus by definition of J, it follows that $J \models B$ iff $Y \models B$ and $J \models a$ iff $Y \models a$, so the claim holds.

  - Subcase 2: $I \not\models B$. The reduct $r^{\underline{I}}$ is either $a \leftarrow \bot$ or $\bot \leftarrow \bot$ and in either case, $J \models r^{\underline{I}}$.

- Case 2: $(X, T) \not\models N \wedge LC$.

  By the condition of I and by Lemma 45, $I \not\models N \wedge LC$ so $r^{\underline{I}}$ is $a \leftarrow \bot$ or $\bot \leftarrow \bot$ depending on whether $I \models a$. Thus, J trivially satisfies $r^{\underline{I}}$.

($\Leftarrow$) Assume $J \models \Pi^{\underline{I}}$. This means that J satisfies every rule in $\Pi^{\underline{I}}$. For any rule r of the form (11.3) in Π

- Case 1: $I \not\models N \wedge LC$.

  By the condition of I and by Lemma 45, $(X, T) \not\models N \wedge LC$. Thus the reduct $\Pi^{(X,T)}$ does not contain a corresponding rule so there is nothing for Y to satisfy.

- Case 2: $I \models N \wedge LC$

  - Subcase 1: $I \models \neg B$.

    By the condition of I and by Lemma 45, $(X, T) \models N \wedge LC$ so the reduct $r^{(X,T)}$ is $a \leftarrow B$. Now by the condition of I, $X \not\models B$ and since $Y \subseteq X$, $Y \not\models B$. Thus, $Y \models r^{(X,T)}$.

  - Subcase 2: $I \models B$

    By the condition of I and by Lemma 45, $(X, T) \models N \wedge LC$ so the reduct $r^{(X,T)}$ is $a \leftarrow B$. Now, since $I \models \Pi$, it must be that $I \models a$ so the reduct $r^{\underline{I}}$ is $a \leftarrow B \wedge LC^{\underline{I}}$. Now since J and I agree on every object constant and since $I \models LC^{\underline{I}}$, $J \models LC^{\underline{I}}$. Thus, since $J \models r^{\underline{I}}$ iff $J \models a \leftarrow B$ so $J \models a \leftarrow B$. Now by definition of J, it follows that $Y \models r^{(X,T)}$.

Theorem 31 *Let Π be an ASP(LC) program, and σ^p and σ^f are defined as above.*

(a) If (X, T) is an LJN-answer set of Π, then for any interpretation $\langle I^f, X\rangle$ of signature $\sigma^p \cup \sigma^f$ such that $I^f \models_{bg} T \cup \overline{T}$, we have $\langle I^f, X\rangle \models_{bg} SM[\Pi; \sigma^p]$.

(b) For any interpretation $I = \langle I^f, X\rangle$ of signature $\sigma^p \cup \sigma^f$, if $\langle I^f, X\rangle \models_{bg} SM[\Pi; \sigma^p]$, then an LJN-interpretation (X, T) where

$$T = \{t \mid t \text{ is a theory atom in } \Pi \text{ such that } I^f \models_{bg} t\}$$

is an LJN-answer set of Π.

Proof.

In this proof, we use Theorem 1 and refer to the reduct characterization.

(a) Assume (X,T) is an LJN-answer set of Π. Take any interpretation $I = \langle I^f, X\rangle$ such that $I^f \models_{bg} T \cup \overline{T}$.

Now for any atom p, by the condition of I, $I \models p$ iff $(X,T) \models p$. Similarly, for any theory atom t occuring in Π, by the condition of I and by Lemma 45, $I \models t$ iff $(X,T) \models t$. Thus, since $(X,T) \models \Pi$, $I \models \Pi$.

We must now show that there is no interpretation J such that $J <^{\sigma_p} I$ and $J \models \Pi^{\underline{I}}$. Take any $J <^{\sigma_p} I$. That is, $J = \langle I^f, Y\rangle$ such that $Y \subset X$. By Lemma 46, $J \models \Pi^{\underline{I}}$ iff $Y \models \Pi^{(X,T)}$ but since (X,T) is an LJN-answer set of Π, $Y \not\models \Pi^{(X,T)}$ and thus $J \not\models \Pi^{\underline{I}}$ so I is a stable model of Π.

(b) Assume $I = \langle I^f, X\rangle$ is a stable model of Π.

Now for any atom p, by definition of (X,T), $(X,T) \models p$ iff $I \models p$. Similarly, for any theory atom t occuring in Π, by the condition of I and Lemma 45, $(X,T) \models t$ iff $I \models t$. Thus, since $I \models \Pi$, $(X,T) \models \Pi$.

We must now show that there is no set of atoms Y such that $Y \subset X$ and $Y \models \Pi^{(X,T)}$. Take any $Y \subset X$. By Lemma 46, $Y \models \Pi^{(X,T)}$ iff $J \models \Pi^{\underline{I}}$ where $J = \langle I^f, Y\rangle$. Since $J <^{\sigma^p} I$ and I is a stable model of Π, $J \not\models \Pi^{\underline{I}}$. Thus $Y \not\models \Pi^{(X,T)}$ and so (X,T) is an LJN-answer set of Π.

11.3.4 Proof of Theorem 32

Theorem 32 *For any definite causal theory T, $I \models CM[T;\boldsymbol{f}]$ iff $I \models SM[Tr(T);\boldsymbol{f}]$.*

Proof. Assume that, without loss of generality, the rules (11.5)–(11.6) have no free variables. It is sufficient to prove that under the assumption that I satisfies T, for every rule (11.5), $J^{\boldsymbol{f}}_{\boldsymbol{g}} \cup I$ satisfies

$$B \rightarrow g(\boldsymbol{t})=t_1$$

iff $J^{\boldsymbol{f}}_{\boldsymbol{g}} \cup I$ satisfies

$$(\neg\neg B)^*(\boldsymbol{g}) \rightarrow g(\boldsymbol{t})=t_1 \wedge f(\boldsymbol{t})=t_1.$$

The claim follows since both B is equivalent to $(\neg\neg B)^*(\boldsymbol{g})$, and I satisfies B. ∎

11.3.5 Proof of Theorem 33

Theorem 33 *Let F be the FOL-representation of a ground IF-program of signature σ and let $\boldsymbol{f}$ be a list of intensional function constants. For any interpretation I of σ, $I \models \mathrm{IF}[F; \boldsymbol{f}]$ iff*

- *I satisfies Π, and*
- *no interpretation J of σ that disagrees with I only on $\boldsymbol{f}$ satisfies $\Pi^{\underline{I}}$.*

Proof. First we prove that, for any implication-free formula F, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$ satisfies $F^{\diamond}(\boldsymbol{d})$ iff $J \models F^I$. This proof is easy by induction.

- Case 1: F is a formula $\neg G$. Then $F^{\diamond}(\boldsymbol{d})$ is $\neg G$. Now, since the members of $\boldsymbol{c}$ are exclusive to I and are not interpreted by $J^{\boldsymbol{c}}_{\boldsymbol{d}}$, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models (\neg G)(\boldsymbol{d})$ iff $I \models \neg G$, We consider two cases for the reduct F^I:
  - Subcase 1: $I \not\models G$. Then F^I is $\top$ and so $J \models F^I$. In this case, $I \models \neg G$, which we saw was equivalent to $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models (\neg G)(\boldsymbol{d})$ so for this subcase, the claim holds.

– Subcase 2: $I \models G$. Then F^{I} is $\bot$ and so $J \not\models F^{I}$. In this case, $I \not\models \neg G$, which we saw was equivalent to $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \not\models (\neg G)(\boldsymbol{d})$ so for this subcase, the claim holds.

- Case 2: F is $\bot$. Then, $F^{\diamond}(\boldsymbol{d})$ is $\bot$ and F^{I} is $\bot$. Since both $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \not\models F^{\diamond}(\boldsymbol{d})$ and $J \not\models F^{I}$, the claim holds in this case.

- Case 3: F is an atomic formula A not in the scope of any negation. Then $F^{\diamond}(\boldsymbol{d})$ is $A(\boldsymbol{d})$. F^{I} is A. Now, since the members of $\boldsymbol{c}$ are exclusive to $J^{\boldsymbol{c}}_{\boldsymbol{d}}$ and are not interpreted by I, $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models A(\boldsymbol{d})$ iff $J^{\boldsymbol{c}}_{\boldsymbol{d}} \models A(\boldsymbol{d})$, which we can rewrite as $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I \models A(\boldsymbol{d})$ iff $J \models A$. Then it is clear that the claim holds in this case.

- Case 4: F is a formula $G \odot H$ where $\odot \in \{\wedge, \vee\}$ that is not in the scope of any negation. Then $F^{\diamond}(\boldsymbol{d})$ is $G^{\diamond}(\boldsymbol{d}) \odot H^{\diamond}(\boldsymbol{d})$. F^{I} is $G \odot H$. The claim holds by I.H. on G and H.

The claim then follows since $J^{\boldsymbol{c}}_{\boldsymbol{d}} \cup I$ satisfies $B^{\diamond} \rightarrow H^{\diamond}$ iff J satisfies $H^{\underline{I}} \leftarrow B^{\underline{I}}$. ∎

11.3.6 *Proof of Theorem 34*

Let T be an IF-program whose rules have the form

$$f(\boldsymbol{t}) = t_1 \leftarrow \neg\neg B \tag{11.8}$$

above is (11.7)) where f is an intensional function constant, $\boldsymbol{t}$ and t_1 do not contain intensional function constants, and B is an arbitrary formula. We identify T with the corresponding first-order formula.

Theorem 34 *$I \models SM[T; \boldsymbol{f}]$ iff $I \models IF[T; \boldsymbol{f}]$.*

Proof.

We wish to show that $I \models T \wedge \neg\exists\widehat{\boldsymbol{f}}(\widehat{\boldsymbol{f}} < \boldsymbol{f} \wedge F^*(\widehat{\boldsymbol{f}}))$ iff $I \models T \wedge \neg\exists\widehat{\boldsymbol{f}}(\widehat{\boldsymbol{f}} \neq \boldsymbol{f} \wedge F^\diamond(\widehat{\boldsymbol{f}}))$. The first conjunctive terms are identical and if $I \not\models T$ then the claim holds.

Let us assume then, that $I \models T$. By definition, $\widehat{\boldsymbol{f}} < \boldsymbol{f}$ is equivalent to $\widehat{\boldsymbol{f}} \neq \boldsymbol{f}$. What remains to be shown is the correspondence between $F^*(\widehat{\boldsymbol{f}})$ and $F^\diamond(\widehat{\boldsymbol{f}})$.

Consider any list of functions $\boldsymbol{g}$ of the same length as $\boldsymbol{f}$. Let $\mathcal{I} = I \cup J^{\boldsymbol{f}}_{\boldsymbol{g}}$ be from an extended signature $\sigma' = \sigma \cup \boldsymbol{g}$ where J is an interpretation of σ and J and I agree on functions not occurring in $\boldsymbol{f}$.

Consider any rule $f(\boldsymbol{t}) = t_1 \leftarrow \neg\neg B$ from T. The corresponding rule in $F^*(\boldsymbol{g})$ is equivalent to

$$f(\boldsymbol{t}) = t_1 \wedge g(\boldsymbol{t}) = t_1 \leftarrow B.$$

The corresponding rule in $F^\diamond(\boldsymbol{g})$ is equivalent to

$$g(\boldsymbol{t}) = t_1 \leftarrow B.$$

Now we consider cases

- $I \not\models B$. Clearly, both versions of the rule are vacuously satisfied by $\mathcal{I}$.
- $I \models B$. Then, since $I \models T$ it must be that $I \models f(\boldsymbol{t}) = t_1$ and so the corresponding rule in $F^*(\boldsymbol{g})$ is further equivalent to

  $$g(\boldsymbol{t}) = t_1 \leftarrow B$$

  which is equivalent to the corresponding rule in $F^\diamond(\boldsymbol{g})$ and so certainly $\mathcal{I}$ is satisfies both corresponding rules or neither.

Thus, $\mathcal{I} \models F^*(\boldsymbol{g})$ iff $\mathcal{I} \models F^\diamond(\boldsymbol{g})$ and so the claim holds. ∎

11.3.7 Proof of Theorem 35

Given a program Π, by Π^{FOL} we denote the FOL representation of Π.

Lemma 47 *Consider a signature σ and a set of constants $\boldsymbol{c}$. Given an ASP{f} program Π of signature σ not containing strong negation,*

(a) For any partial interpretation I of signature σ that maps every constant in $\sigma \setminus \boldsymbol{c}$ to itself, there is a consistent set S of seed literals such that $I \models_p \Pi^{FOL}$ iff $S \models_b \Pi$.

(b) For any consistent set of seed literals S, there is a partial interpretation I such that $I \models_p \Pi^{FOL}$ iff $S \models_b \Pi$.

Proof. *Part (a):* Given a partial interpretation I, let S be the set $\{f(\boldsymbol{v}) = w : f(\boldsymbol{v})^I = w\} \cup \{p(\boldsymbol{v}) : p(\boldsymbol{v})^I = 1\}$. We note that this is a consistent set of seed literals since a partial interpretation maps $f(\boldsymbol{v})$ to at most one object constant.

We also note that by the definition of S, for any atomic sentence A, we have $I \models_p A$ iff $S \models_b A$. Now, consider any rule r from Π. $I \models_p r^{FOL}$ iff $I \models_p head(r)^{FOL}$ or $I \not\models_p body(r)^{FOL}$. By the previous observation, this is equivalent to $S \models_b head(r)$ or $S \not\models_b body(r)$ since $body(r)$ is a conjunction of atomic formulas. This is precisely the definition of $S \models_b r^{FOL}$.

Part (b): Given a consistent set of seed literals S, let I be the partial interpretation defined as follows:

- for every object constant $v \in \sigma \setminus \boldsymbol{c}$, we have $v^I = v$.
- for every predicate constant $p \in \boldsymbol{c}$ and every list of object constants $\boldsymbol{v}$, we have $p(\boldsymbol{v})^I = 1$ iff $p(\boldsymbol{v}) \in S$.

- for every function constant $f \in \boldsymbol{c}$ and every list of object constants $\boldsymbol{v}$, we have $f(\boldsymbol{v})^I = u$ if S does not contain $f(\boldsymbol{v})$, and $f(\boldsymbol{v})^I = w$ if $f(\boldsymbol{v}) = w$ is in S.

We note that the last bullet is well-defined since S is a consistent set of seed literals so that there cannot be two distinct object constants a and b such that $f(\boldsymbol{v}) = a \in S$ and $f(\boldsymbol{v}) = b \in S$.

We also note that by definition of I, for any atomic sentence A, we have $I \models_p A$ iff $S \models_b A$. Now, consider any rule r from Π. $S \models_b r$ iff $S \models_b head(r)$ or $S \not\models_b body(r)$. By the previous observation, this is equivalent to $I \models_p head(r)^{FOL}$ or $I \not\models_p body(r)^{FOL}$ since $body(r)$ is a conjunction of atomic formulas. This is precisely the definition of $I \models_p r$. ∎

The proof of Lemma 47 tells us that a consistent set of seed literals can be identified with a partial interpretation.

Lemma 48 *For consistents sets of seed literals J and I of the same signature, J is a proper subset of I iff $J \prec^{\mathbf{c}} I$ when we view them as partial interpretations.*

Proof. We first note that since consistent sets of literals map every object constant in $\sigma \setminus \boldsymbol{c}$ to itself, the partial interpretation view does the same which corresponds to the first condition for $J \prec^{\mathbf{c}} I$. The second condition of $J \prec^{\mathbf{c}} I$ is $p^J \subseteq p^I$ for all predicate constants in $\boldsymbol{c}$, which corresponds exactly to the predicate part of J being a subset of the predicate part of I. Finally, the third condition of $J \prec^{\mathbf{c}} I$ is $f^J(\boldsymbol{\xi}) = u$ or $f^J(\boldsymbol{\xi}) = f^I(\boldsymbol{\xi})$ corresponds to the function part of J being a subset of the function part of I since we identify a partial interpretation mapping an element to u to the absence of that element in the set. ∎

Theorem 35 *For any ASP{f} program Π with intensional constants $\boldsymbol{c}$ and any consistent set I of seed literals, if Π has no strong negation, then I is a Balduccini* *answer set of* Π *iff* $I \models_p CBL[\Pi; \boldsymbol{c}]$.

Proof. By definition and by using the equivalent reformulation presented and justified in Lemma 48 and Lemma 47, I is a Balduccini answer set of a program Π iff $I \models_p \Pi$ and for any partial interpretation J such that $J \prec^{\boldsymbol{c}} I$, we have $J \not\models_p \Pi^I$. Since this definition uses the same reduct and same notion of satisfaction, this is equivalent to the reduct reformulation of the Cabalar semantics. Further, this is equivalent to $I \models_p \mathrm{CBL}[\Pi^{FOL}; \boldsymbol{c}]$ by Theorem 21. ∎

11.3.8 Proof of Theorem 36

Theorem 36 *For any ASP{f} program* Π *with intensional constants* $\boldsymbol{c}$ *and any consistent set* I *of seed literals,* I *is a Balduccini answer set of* Π *iff* I *is a Balduccini answer set of* $\Pi^{\#}$.

Proof. First, we show that $I \models_b \sim(f = g)$ iff $I \models_b (f = f) \wedge (g = g) \wedge \neg(f = g)$.

Left-to-right: Asssume $I \models_b \sim(f = g)$. By definition, I contains both $f = c_1$ and $g = c_2$ for some object constants c_1 and c_2 such that $c_1 \neq c_2$. Clearly, each of $I \models f = f$, $I \models g = g$ and $I \not\models f = g$ holds.

Right-to-left: $I \models_b (f = f) \wedge (g = g) \wedge \neg(f = g)$. Since $I \models_b f = f$ and $I \models g = g$, it follows that I contains $f = c_1$ and I contains $f = c_2$ for some c_1 and c_2. Further, since $I \models \neg(f = g)$, it must be that $c_1 \neq c_2$, from which the claim follows.

From this it is not difficult to check that Π^I is equivalent to $(\Pi^{\#})^I$, from which the claim follows. ∎

Chapter 12

CONCLUSION

Reasoning about real-world domains faces several challenges among which are performing defeasible reasoning and efficient computation in the presence of large domains. Answer Set Programming, based on the stable model semantics, addressed the issue of defeasible reasoning for predicates only but due to grounding based computation, large domains preclude efficient computation in ASP.

Recent proposals have loosely integrated ASP with other declarative paradigms including constraint programming, satisfiability modulo theories, and mixed integer programming. These proposals have resulted in systems such as ACSOLVER Mellarkod *et al.* (2008), CLINGCON Gebser *et al.* (2009b), EZCSP Balduccini (2009), IDP Mariën *et al.* (2008), and MINGO Liu *et al.* (2012) that have partially alleviated the grounding bottleneck. However, the functions there were treated as in first-order logic so that defeasible reasoning could only be performed on predicates and not functions.

On the other hand, several recent formalisms Cabalar (2011); Lifschitz (2012); Balduccini (2012) extend the stable model semantics to support intensional functions so that defeasible reasoning can be performed on both functions and predicates. However, these approaches focused on rich modeling and did not address the grounding bottleneck.

This research is a novel framework that tightly integrates ASP and SMT in order to address the grounding bottleneck faced by ASP while still supporting defeasible reasoning on both functions and predicates for which SMT is unsuitable. This framework is based on the newly-introduced functional stable model semantics.

The prototype implementations presented in this dissertation serve as a proof-of-concept for this framework. We are able to perform defeasible reasoning on functional fluents directly with functions rather than with predicates. This is not simply syntactic sugar however; the ASPMT2SMT system is able to avoid the grounding bottleneck in some domains and dramatically outperform the state-of-the-art ASP systems. While more mature systems that loosely couple ASP with other declarative paradigms are able to achieve slightly better performance, these systems lack the ability to perform defeasible reasoning in a suitable way. The advantages of the ASPMT2SMT system were leveraged to create a non-monotonic spatial reasoning system described in Walega *et al.* (2015).

We have investigated many properties of the first-order stable model semantics and have shown that analogous properties hold for the functional stable model semantics. By defining our semantics in the style of the first-order stable model semantics and studying the relationship between these two semantics, we were able to establish a body of results that other formalisms would need to establish using dissimilar terminology and concepts. However, by establishing formal relationships between different definitions of intensional functions, we have been able to establish results for these other definitions such as the generalization of the unfolding process in Theorem 1 in Cabalar (2011).

We expect that future research in this area will establish further results for the functional stable model semantics analogous to the useful results that have made the stable model semantics successful. We also expect that future implementations of ASPMT will achieve similar performance improvements to those elicited by SMT for the SAT community.

Babb, J. and J. Lee, "Cplus2ASP: Computing action language $\mathcal{C}+$ in answer set programming", in "Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)", pp. 122–134 (2013).

Balduccini, M., "Representing constraint satisfaction problems in answer set programming", in "Workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP 2009)(July 2009)", pp. 16–30 (2009).

Balduccini, M., "A "conservative" approach to extending answer set programming with non-herbrand functions", in "Correct Reasoning - Essays on Logic-Based AI in Honour of Vladimir Lifschitz", pp. 24–39 (2012).

Balduccini, M., "ASP with non-herbrand partial functions: a language and system for practical use", Theory and Practice of Logic Programming **13**, 4-5, 547–561 (2013).

Balduccini, M., M. Gelfond, M. Nogueira, R. Watson and M. Barry, "An A-Prolog decision support system for the Space Shuttle", in "Working Notes of the AAAI Spring Symposium on Answer Set Programming", pp. 169–183 (2001).

Barrett, C., R. Sebastiani, S. A. Seshia and C. Tinelli, "Satisfiability modulo theories", in "Handbook of Satisfiability", pp. 825–885 (2009).

Bartholomew, M. and J. Lee, "Stable models of formulas with intensional functions", in "Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)", pp. 2–12 (2012).

Bartholomew, M. and J. Lee, "Functional stable model semantics and answer set porgramming modulo theories", in "23rd International Joint Conference on Articial Intelligence (IJCAI 2013)", pp. 718–724 (2013a).

Bartholomew, M. and J. Lee, "On the stable model semnatics for intensional functions", Journal of Theory and Practice of Logic Programming (TPLP) **13**, 4-5, 863–876 (2013b).

Bounimova, E., P. Godefroid and D. Molnar, "Billions and billions of constraints: Whitebox fuzz testing in production", in "Proceedings of the International Conference on Software Engineering", pp. 122–131 (2013).

Brooks, D. R., E. Erdem, S. T. Erdoğan, J. W. Minett and D. Ringe, "Inferring phylogenetic trees using answer set programming", Journal of Automated Reasoning **39**, 471–511 (2007).

Cabalar, P., "Functional answer set programming", Theory and Practice of Logic Programming (TPLP) **11**, 2-3, 203–233 (2011).

Cabalar, P., L. F. del Cerro, D. Pearce and A. Valverde, "A free logic for stable models with partial intensional functions", in "Proceedings of European Conference on Logics in Artificial Intelligence (JELIA)", pp. 340–354 (2014).

Chintabathina, S., "Towards answer set prolog based architectures for intelligent agents.", in "AAAI'08", pp. 1843–1844 (2008).

del Cerro, L. F., D. Pearce and A. Valverde, "Fqht: The logic of stable models for logic programs with intensional functions", in "Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence", pp. 891–897 (2013).

Ferraris, P., "Answer sets for propositional theories", in "Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)", pp. 119–131 (2005).

Ferraris, P., J. Lee and V. Lifschitz, "Stable models and circumscription", Artificial Intelligence **175**, 236–263 (2011).

Ferraris, P., J. Lee, V. Lifschitz and R. Palla, "Symmetric splitting in the general theory of stable models", in "Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)", pp. 797–803 (2009).

Ge, Y., C. Barrett and C. Tinelli, "Solving quantified verification conditions using satisfiability modulo theories", Proceedings of the 21st International Conference on Automated Deduction (CADE 2007) **21**, 167–182 (2007).

Gebser, M., M. Ostrowski and T. Schaub, "Constraint answer set solving", in "Proceedings of International Conference on Logic Programming (ICLP)", pp. 235–249 (2009a).

Gebser, M., M. Ostrowski and T. Schaub, "Constraint answer set solving", in "Proceedings of 25th International Conference on Logic Programming (ICLP)", pp. 235–249 (Springer, 2009b).

Gelfond, M. and Y. Kahl, *Knowledge Representation, Reasoning and the Design of Intelligent Agents* (Cambridge University Press, 2014).

Gelfond, M. and V. Lifschitz, "The stable model semantics for logic programming", in "Proceedings of International Logic Programming Conference and Symposium", edited by R. Kowalski and K. Bowen, pp. 1070–1080 (MIT Press, 1988).

Gelfond, M. and V. Lifschitz, "Classical negation in logic programs and disjunctive databases", New Generation Computing **9**, 365–385 (1991).

Gelfond, M., V. Lifschitz and A. Rabinov, "What are the limitations of the situation calculus?", in "Automated Reasoning: Essays in Honor of Woody Bledsoe", edited by R. Boyer, pp. 167–179 (Kluwer, 1991).

Giunchiglia, E., J. Lee, V. Lifschitz, N. McCain and H. Turner, "Nonmonotonic causal theories", Artificial Intelligence **153(1–2)**, 49–104 (2004).

Janhunen, T., G. Liu and I. Niemelä, "Tight integration of non-ground answer set programming and satisfiability modulo theories", Proceedings of Grounding and Transformations for Theories with Variables **11**, 1–13 (2011).

Kim, T.-W., J. Lee and R. Palla, "Circumscriptive event calculus as answer set programming", in "Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)", pp. 823–829 (2009).

Lee, J., Y. Lierler, V. Lifschitz and F. Yang, "Representing synonymity in causal logic and in logic programming", in "Proceedings of International Workshop on Nonmonotonic Reasoning (NMR)", (2010).

Lee, J. and V. Lifschitz, "Describing additive fluents in action language $\mathcal{C}+$", in "Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)", pp. 1079–1084 (2003).

Lee, J. and R. Palla, "System F2LP – computing answer sets of first-order formulas", in "Procedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)", pp. 515–521 (2009).

Lifschitz, V., "What is answer set programming?", in "Proceedings of the AAAI Conference on Artificial Intelligence", pp. 1594–1597 (MIT Press, 2008).

Lifschitz, V., "Logic programs with intensional functions", in "Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)", pp. 24–31 (2012).

Lifschitz, V., L. Morgenstern and D. Plaisted, "Knowledge representation and classical logic", in "Handbook of Knowledge Representation", edited by F. van Harmelen, V. Lifschitz and B. Porter, pp. 3–88 (Elsevier, 2008).

Lifschitz, V., D. Pearce and A. Valverde, "Strongly equivalent logic programs", ACM Transactions on Computational Logic **2**, 526–541 (2001).

Lifschitz, V. and F. Yang, "Eliminating function symbols from a nonmonotonic causal theory", in "Knowing, Reasoning, and Acting: Essays in Honour of Hector J. Levesque", edited by G. Lakemeyer and S. A. McIlraith (College Publications, 2011), URL `http://www.cs.utexas.edu/users/ai-lab/pub-view.php?PubID=127054`.

Lifschitz, V. and F. Yang, "Functional completion", Journal of Applied Non-Classical Logics **23**, 1-2, 121–130 (2013).

Lin, F. and Y. Wang, "Answer set programming with functions", in "Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)", pp. 454–465 (2008).

Liu, G., T. Janhunen and I. Niemelä, "Answer set programming via mixed integer programming", in "Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)", pp. 32–42 (2012).

Mariën, M., J. Wittocx and M. Denecker, "The idp system: a model expansion system for an extension of classical logic", in "In Workshop on Logic and Search", pp. 153–165 (2008).

McCarthy, J., "Elaboration tolerance", in "Common Sense", vol. 98 (1998), updated version available at http://jmc.stanford.edu/articles/elaboration/elaboration.pdf.

Mellarkod, V. S., M. Gelfond and Y. Zhang, "Integrating answer set programming and constraint logic programming", Annals of Mathematics and Artificial Intelligence **53**, 1-4, 251–287 (2008).

Moy, Y., N. Bjørner and D. Sielaf, "Modular bug-finding for integer overflows in the large: Sound, efficient, bit-precise static analysis", Tech. rep., Microsoft Research (2009).

Tiihonen, J., T. Soininen, I. Niemelä and R. Sulonen, "A practical tool for mass-customising configurable products", in "Proceedings of the 14th International Conference on Engineering Design", pp. 1290–1299 (2003).

Truszczynski, M., "Connecting first-order asp and the logic fo(id) through reducts", in "Correct Reasoning", pp. 543–559 (2012).

Walega, P., M. Bhatt and C. Schultz, "ASPMT(QS): Non-monotonic spatial reasoning with answer set programming modulo theories", in "Proceedings of the 13th International Conference on Logic Programming and Nonmonotonic Reasoning", LPNMR '15, pp. 488–501 (2015).