Translating First-Order Causal Theories into Answer Set Programming

Vladimir Lifschitz and Fangkai Yang

Department of Computer Sciences
University of Texas at Austin

Abstract. Nonmonotonic causal logic became a basis for the semantics of several expressive action languages. Norman McCain and Paolo Ferraris showed how to embed propositional causal theories into logic programming, and this work paved the way to the use of answer set solvers for answering queries about actions described in causal logic. In this paper we generalize these embeddings to first-order causal logic—a system that has been used to simplify the semantics of variables in action descriptions.

1 Introduction

Propositional nonmonotonic causal logic [McCain and Turner, 1997] and its generalizations became a basis for the semantics of several expressive action languages [Giunchiglia and Lifschitz, 1998, Giunchiglia *et al.*, 2004, Lifschitz and Ren, 2006, Lifschitz and Ren, 2007]. The last paper argues, in particular, that one of these generalizations—first-order causal logic in the sense of [Lifschitz, 1997]—is useful for defining the semantics of variables in action descriptions.

An important theorem due to Norm McCain [McCain, 1997, Proposition 6.7] shows how to embed a subset of propositional causal logic into the language of logic programming under the answer set semantics [Gelfond and Lifschitz, 1991]. A similar translation, applicable to arbitrary propositional causal theories, is defined in [Ferraris, 2007]. These results (reviewed in the next section) paved the way to the use of answer set programming (ASP) for answering queries about actions described in causal logic.[1]

In this note we extend the translations given by McCain and Ferraris to first-order causal theories. Our generalizations rely on the approach to stable models (answer sets) proposed in [Ferraris *et al.*, 2007, Ferraris *et al.*, 2010].

2 Background: Translating Propositional Causal Theories into ASP

2.1 Propositional Causal Theories

A nonmonotonic causal theory in the sense of [McCain and Turner, 1997] is a set of *causal rules* of the form $F \Leftarrow G$, where F and G are propositional formulas

[1] See, for instance, `http://www.cs.uni-potsdam.de/~tgrote/coala/`.

(the *head* and the *body* of the rule). The rule can be read "F is caused if G is true."

Distinguishing between being true and having a cause turns out to be essential for the study of commonsense reasoning. The assertion "if the light is on at time 2 and you toggle the switch then the light will be off at time 3" can be written as an implication:

$$On_2 \wedge Toggle_2 \rightarrow Off_3.$$

In causal logic we can express that under the same assumption *there is a cause* for the light to be off at time 3:

$$Off_3 \Leftarrow On_2 \wedge Toggle_2.$$

(Performing the toggle action is the cause.) McCain and Turner show that distinctions like this help us solve the frame problem and overcome other difficulties arising in the theory of reasoning about actions.

The semantics of theories of this kind defines when a propositional interpretation (truth assignment) is a model of the given theory (is "causally explained" by the theory, in the terminology of McCain and Turner). We do not reproduce the definition here, because a more general semantics is described below in Section 3.1. But here is an example: the causal theory

$$\begin{aligned} p &\Leftarrow \neg q \\ \neg q &\Leftarrow p \end{aligned} \tag{1}$$

has one model, according to the semantics from [McCain and Turner, 1997]. In this model, p is true and q is false.

2.2 McCain's Translation

McCain's translation is applicable to a propositional causal theory T if the head of each rule of T is a literal, and the body is a conjunction of literals:

$$L \Leftarrow A_1 \wedge \cdots \wedge A_m \wedge \neg A_{m+1} \wedge \cdots \neg A_n. \tag{2}$$

The logic program corresponding to T consists of the logic programming rules

$$L \leftarrow not\ \neg A_1, \ldots, not\ \neg A_m, not\ A_{m+1}, \ldots, not\ A_n \tag{3}$$

for all rules (2) of T. According to Proposition 6.7 from [McCain, 1997], complete answer sets of this logic program are identical to the models of T. (A set of literals is *complete* if it contains exactly one member of each complementary pair of literals $A, \neg A$. In the statement above, we identify a complete set of literals with the corresponding truth assignment.)

For instance, McCain's translation turns causal theory (1) into

$$\begin{aligned} p &\leftarrow not\ q \\ \neg q &\leftarrow not\ \neg p. \end{aligned} \tag{4}$$

The only answer set of this program is $\{p, \neg q\}$. It is complete, and it corresponds to the model of causal theory (1).

2.3 Eliminating Strong Negation

Rule (3) involves two kinds of negation: negation as failure (*not*) and strong, or classical, negation ($\neg$). As observed in [Gelfond and Lifschitz, 1991], strong negation can be eliminated from a logic program in favor of additional atoms. Denote the new atom, representing a negative literal $\neg A$, by $\widehat{A}$. Then (3) will become

$$A_0 \leftarrow \mathit{not}\ \widehat{A_1}, \ldots, \mathit{not}\ \widehat{A_m}, \mathit{not}\ A_{m+1}, \ldots, \mathit{not}\ A_n \tag{5}$$

if L is a positive literal A_0, and

$$\widehat{A_0} \leftarrow \mathit{not}\ \widehat{A_1}, \ldots, \mathit{not}\ \widehat{A_m}, \mathit{not}\ A_{m+1}, \ldots, \mathit{not}\ A_n \tag{6}$$

if L is a negative literal $\neg A_0$. The *modified McCain translation* of T consists of

- the rules (5), (6) corresponding to the rules of T, and
- the completeness constraints

$$\begin{array}{l} \leftarrow A, \widehat{A} \\ \leftarrow \mathit{not}\ A, \mathit{not}\ \widehat{A} \end{array} \tag{7}$$

for all atoms A.

For instance, the modified McCain translation of (1) is

$$\begin{array}{r} p \leftarrow \mathit{not}\ q \\ \widehat{q} \leftarrow \mathit{not}\ \widehat{p} \\ \leftarrow p, \widehat{p} \\ \leftarrow \mathit{not}\ p, \mathit{not}\ \widehat{p} \\ \leftarrow q, \widehat{q} \\ \leftarrow \mathit{not}\ q, \mathit{not}\ \widehat{q}. \end{array} \tag{8}$$

The only answer set of this program is $\{p, \widehat{q}\}$.

2.4 Rules as Formulas

The definition of an answer set for sets of propositional formulas proposed in [Ferraris, 2005] is a generalization of the concept of an answer set for propositional logic programs without strong negation, in the sense that rewriting each rule of a given program in the syntax of propositional logic produces a collection of formulas with the same answer sets as the given program. For instance, rules (5) and (6), rewritten as propositional formulas, become

$$\neg\widehat{A_1} \wedge \cdots \wedge \neg\widehat{A_m} \wedge \neg A_{m+1} \wedge \cdots \wedge \neg A_n \rightarrow A_0$$

and

$$\neg\widehat{A_1} \wedge \cdots \wedge \neg\widehat{A_m} \wedge \neg A_{m+1} \wedge \cdots \wedge \neg A_n \rightarrow \widehat{A_0}.$$

The completeness constraints for an atom A turn into

$$\begin{array}{l} \neg(A \wedge \widehat{A}) \\ \neg(\neg A \wedge \neg\widehat{A}). \end{array} \tag{9}$$

Note that the process of rewriting a rule as a formula is applicable only when the rule does not contain strong negation; the symbol $\neg$ in the resulting formula corresponds to the negation as failure symbol (*not*) in the rule.

One of the advantages of writing rules as formulas is that it allows us to relate properties of answer sets to subsystems of classical logic. We know, for instance, that if the equivalence of two sets Γ, Δ of formulas can be proved in intuitionistic logic (or even in the stronger logic of here-and-there) then Γ and Δ have the same answer sets [Ferraris, 2005, Proposition 2]. It follows that replacing the completeness constraints (9) with the intuitionistically equivalent formula

$$\neg(A \leftrightarrow \widehat{A}) \tag{10}$$

does not affect the class of answer sets.

If we rewrite program (8) in the syntax of propositional logic and modify the completeness constraints as shown above then (8) will turn into

$$\begin{array}{l} \neg q \rightarrow p \\ \neg\widehat{p} \rightarrow \widehat{q} \\ \neg(p \leftrightarrow \widehat{p}) \\ \neg(q \leftrightarrow \widehat{q}). \end{array} \tag{11}$$

This collection of formulas is essentially identical to logic program (8), and it has the same answer set.

2.5 Translating Arbitrary Definite Theories

The paper [Ferraris, 2007] shows, among other things, how to lift the requirement, in the definition of McCain's translation, that the bodies of all causal rules should be conjunctions of literals. Take any set T of causal rules of the forms

$$A \Leftarrow G \tag{12}$$

and

$$\neg A \Leftarrow G \tag{13}$$

where A is an atom and G is an arbitrary formula (such rules are called *definite*). Form, for each rule (12), the formula

$$\neg\neg G \rightarrow A,$$

and, for each rule (13), the formula

$$\neg\neg G \rightarrow \widehat{A}.$$

Then add completeness constraints (10) for all atoms A. Answer sets of this collection of propositional formulas correspond to the models of T.

In application to example (1), this modification of McCain's translation gives

$$\begin{array}{l} \neg\neg\neg q \rightarrow p \\ \neg\neg p \rightarrow \widehat{q} \\ \neg(p \leftrightarrow \widehat{p}) \\ \neg(q \leftrightarrow \widehat{q}). \end{array} \tag{14}$$

It is not surprising that (14) has the same answer set as (11): the two collections of formulas are intuitionistically equivalent to each other.

2.6 Ferraris's Translation

The main result of [Ferraris, 2007] deals with causal theories "in clausal form": bodies of rules are arbitrary formulas, but the head of every rule is a disjunction of literals. This is essentially the general case, because any propositional causal theory can be converted to clausal form by converting the head of each rule to conjunctive normal form $D_1 \wedge \cdots \wedge D_k$ and then breaking it into k rules with the heads $D_1, \ldots, D_k$.

Ferraris's translation turns the rule

$$\bigvee_{A \in Pos} A \vee \bigvee_{A \in Neg} \neg A \Leftarrow G$$

(Pos and Neg are sets of atoms) into the implication

$$\neg\neg G \wedge \bigwedge_{A \in Pos} (\widehat{A} \vee \neg\widehat{A}) \wedge \bigwedge_{A \in Neg} (A \vee \neg A) \rightarrow \bigvee_{A \in Pos} A \vee \bigvee_{A \in Neg} \widehat{A}.$$

For instance, it transforms

$$p \vee \neg q \Leftarrow r$$

into

$$\neg\neg r \wedge (\widehat{p} \vee \neg\widehat{p}) \wedge (q \vee \neg q) \rightarrow p \vee \widehat{q}. \tag{15}$$

The number of "excluded middle formulas" in the antecedent of the implication, such as $\widehat{p} \vee \neg\widehat{p}$ and $q \vee \neg q$ in (15), equals the number of disjunctive terms in the head of the given causal rule. In particular, the result of Ferraris's translation includes one such formula when the head of the given causal rule is a single literal, as in Section 2.5. For instance, in application to (8) this process would produce

$$\begin{array}{l} \neg q \wedge (\widehat{p} \vee \neg\widehat{p}) \rightarrow p \\ \neg\widehat{p} \wedge (q \vee \neg q) \rightarrow \widehat{q} \\ \neg(p \leftrightarrow \widehat{p}) \\ \neg(q \leftrightarrow \widehat{q}). \end{array} \tag{16}$$

This collection of formulas differs from (14) by the presence of excluded middle formulas in the antecedents of the two implications. These conjuncive terms are redundant, in the sense that dropping them from (16) is an intuitionistically equivalent transformation and consequently does not affect the class of answer sets.

But when the result of translating a rule has more that one excluded middle formula in the antecedent, as in example (15), then the presence of these formulas may be crucial for the validity of the translation [Ferraris, 2007, Section 4].

3 Review: Causal Theories and Stable Models in First-Order Setting

In this section we review the definition of a first-order causal theory from [Lifschitz, 1997] and the definition of a stable model of a first-order sentence from [Ferraris *et al.*, 2010]. Both definitions are based on syntactic transformations that produce second-order formulas.

3.1 First-Order Causal Theories

According to [Lifschitz, 1997], a first-order causal theory T is defined by

- a finite set $\mathbf{p}$ of predicate constants (other than equality), called the *explainable symbols* of T,[2] and
- a finite set of *causal rules* of the form $F \Leftarrow G$, where F and G are first-order formulas.

The semantics of first-order causal theories can be described as follows. For each $p \in \mathbf{p}$, choose a new predicate variable υp of the same arity, and let $\upsilon\mathbf{p}$ stand for the list of all these variables. By $T^\dagger(\upsilon\mathbf{p})$ we denote the conjunction of the formulas

$$\forall\mathbf{x}(G \rightarrow F^{\mathbf{p}}_{\upsilon\mathbf{p}}) \tag{17}$$

for all rules $F \Leftarrow G$ of T, where $\mathbf{x}$ is the list of all free variables of F, G. (The expression $F^{\mathbf{p}}_{\upsilon\mathbf{p}}$ denotes the result of substituting the variables $\upsilon\mathbf{p}$ for the corresponding constants $\mathbf{p}$ in F.) We view T as shorthand for the sentence

$$\forall\upsilon\mathbf{p}(T^\dagger(\upsilon\mathbf{p}) \leftrightarrow (\upsilon\mathbf{p} = \mathbf{p})). \tag{18}$$

(By $\upsilon\mathbf{p} = \mathbf{p}$ we denote the conjunction of the formulas $\forall\mathbf{x}(\upsilon p(\mathbf{x}) \leftrightarrow p(\mathbf{x}))$ for all $p \in \mathbf{p}$, where $\mathbf{x}$ is a tuple of distinct object variables.)

Consider, for instance, the causal theory T with the explainable symbol p that consists of two rules:

$$p(a) \Leftarrow \top$$

(here $\top$ is the logical constant *true*) and

$$\neg p(x) \Leftarrow \neg p(x).$$

[2] To be precise, the definition in [Lifschitz, 1997] is more general: object and function constants can be treated as explainable as well.

The first rule says that there is a cause for a to have property p. The second rule says that if an object does not have property p then there is a cause for that; including this rule in a causal theory has, informally speaking, the same effect as saying that p is "false by default" [Lifschitz, 1997, Section 3]. In this case, $T^\dagger(\upsilon p)$ is

$$\upsilon p(a) \land \forall x(\neg p(x) \to \neg \upsilon p(x)),$$

so that T is understood as shorthand for the sentence

$$\forall \upsilon p(\upsilon p(a) \land \forall x(\neg p(x) \to \neg \upsilon p(x)) \leftrightarrow \forall x(\upsilon p(x) \leftrightarrow p(x))).$$

This sentence is equivalent to the first-order formula $\forall x(p(x) \leftrightarrow x = a)$.

3.2 Operator SM

If p and q are predicate constants of the same arity then $p \leq q$ stands for the formula $\forall \mathbf{x}(p(\mathbf{x}) \to q(\mathbf{x}))$, where $\mathbf{x}$ is a tuple of distinct object variables. If $\mathbf{p}$ and $\mathbf{q}$ are tuples $p_1, \dots, p_n$ and $q_1, \dots, q_n$ of predicate constants then $\mathbf{p} \leq \mathbf{q}$ stands for the conjunction

$$(p_1 \leq q_1) \land \cdots \land (p_n \leq q_n),$$

and $\mathbf{p} < \mathbf{q}$ stands for $(\mathbf{p} \leq \mathbf{q}) \land \neg(\mathbf{q} \leq \mathbf{p})$. In second-order logic, we apply the same notation to tuples of predicate variables.

We will define the *stable model operator with the intensional predicates* $\mathbf{p}$, denoted by $\text{SM}_\mathbf{p}$ [Ferraris *et al.*, 2009]. Some details of the definition depend on which propositional connectives and quantifiers are treated as primitives, and which of them are viewed as abbreviations. We assume that

$$\bot \text{ (falsity)}, \ \land, \ \lor, \ \to, \ \forall, \ \exists$$

are the primitives; $\neg F$ stands for $F \to \bot$, $\top$ stands for $\bot \to \bot$, and $F \leftrightarrow G$ is $(F \to G) \land (G \to F)$.

Let $\mathbf{p}$ be a list of distinct predicate constants (other than equality). For each $p \in \mathbf{p}$, choose a new predicate variable υp of the same arity, and let $\upsilon \mathbf{p}$ stand for the list of all these variables. For any first-order sentence F, by $\text{SM}_\mathbf{p}[F]$ we denote the second-order sentence

$$F \land \neg \exists \upsilon \mathbf{p}((\upsilon \mathbf{p} < \mathbf{p}) \land F^*(\upsilon \mathbf{p})),$$

where $F^*(\upsilon \mathbf{p})$ is defined recursively:

- $p(\mathbf{t})^* = \upsilon p(\mathbf{t})$ for any $p \in \mathbf{p}$ and any tuple $\mathbf{t}$ of terms;
- $F^* = F$ for any atomic F that does not contain members of $\mathbf{p}$;
- $(F \land G)^* = F^* \land G^*$;
- $(F \lor G)^* = F^* \lor G^*$;
- $(F \to G)^* = (F^* \to G^*) \land (F \to G)$;
- $(\forall x F)^* = \forall x F^*$;
- $(\exists x F)^* = \exists x F^*$.

A model of F is *stable* (relative to the set $\mathbf{p}$ of intensional predicates) if it satisfies $\mathrm{SM}_{\mathbf{p}}[F]$.

For instance, let F be the formula

$$\forall x(p(x) \rightarrow (q(x) \vee \neg q(x)) \tag{19}$$

(it can be thought of as a representation of the choice rule `{q(X)} :- p(X)` in the input language of LPARSE).[3] If we take q to be the only intensional predicate then $F^*(\upsilon q)$ is

$$\forall x((p(x) \rightarrow (\upsilon q(x) \vee (\neg \upsilon q(x) \wedge \neg q(x)))) \wedge \ (p(x) \rightarrow (q(x) \vee \neg q(x)))),$$

which is equivalent to

$$\forall x((p(x) \rightarrow (\upsilon q(x) \vee \neg q(x))).$$

Consequently $\mathrm{SM}_q[F]$ is equivalent to

$$\begin{array}{l} \forall x(p(x) \rightarrow (q(x) \vee \neg q(x))) \\ \quad \wedge \neg \exists \upsilon q((\upsilon q < q) \wedge \forall x(p(x) \rightarrow (\upsilon q(x) \vee \neg q(x))). \end{array} \tag{20}$$

The first conjunctive term here is logically valid and can be dropped. The second is equivalent to the first-order formula $\forall x(q(x) \rightarrow p(x))$, which reflects the intuitive meaning of choice: q is an arbitrary subset of p.

4 Translating First-Order Causal Theories

4.1 A First-Order Counterpart of McCain's Translation

In this section we extend the McCain translation as described in Section 2.5 to first-order causal theories. By T we denote here a causal theory in the sense of Section 3.1 such that the head of every rule of T is a literal containing an explainable predicate. Thus every rule of T has the form

$$p(\mathbf{t}) \Leftarrow G \tag{21}$$

or the form

$$\neg p(\mathbf{t}) \Leftarrow G, \tag{22}$$

where p is an explainable predicate and $\mathbf{t}$ is a tuple of terms. For instance, the example at the end of Section 3.1

$$\begin{array}{r} p(a) \Leftarrow \top \\ \neg p(x) \Leftarrow \neg p(x) \end{array} \tag{23}$$

is a causal theory of this type.

[3] For a description of the language see `http://www.tcs.hut.fi/Software/smodels/lparse.ps`.

For every element p of the set $\mathbf{p}$ of explainable predicates, let $\widehat{p}$ be a new predicate constant of the same arity, and let $\widehat{\mathbf{p}}$ be the list of all these predicate constants. By CC we denote the conjunction of the formulas

$$\forall \mathbf{x} \neg (p(\mathbf{x}) \leftrightarrow \widehat{p}(\mathbf{x})), \tag{24}$$

where $\mathbf{x}$ is a tuple of distinct object variables, for all p from $\mathbf{p}$. These formulas are first-order counterparts of completeness constraints (10).

The *McCain translation* MC[T] of T is the conjunction of

- formulas $\widetilde{\forall}(\neg\neg G \rightarrow p(\mathbf{t}))$ for all rules (21) of T,[4]
- formulas $\widetilde{\forall}(\neg\neg G \rightarrow \widehat{p}(\mathbf{t}))$ for all rules (22) of T,
- completeness constraints CC.

For instance, if T is (23) then MC[T] is the conjunction of the formulas

$$\begin{array}{l} \neg\neg\top \rightarrow p(a) \\ \forall x(\neg\neg\neg p(x) \rightarrow \widehat{p}(x)) \\ \forall x \neg (p(x) \leftrightarrow \widehat{p}(x)), \end{array}$$

or, after (intuitionistically acceptable) simplifications,

$$p(a) \wedge \forall x(\neg p(x) \rightarrow \widehat{p}(x)) \wedge \forall x \neg (p(x) \leftrightarrow \widehat{p}(x)).$$

In logic programming syntax, this formula can be rewritten as

$$\begin{array}{l} p(a) \\ \widehat{p}(x) \leftarrow \textit{not } p(x) \\ \phantom{\widehat{p}(x)} \leftarrow p(x), \widehat{p}(x) \\ \phantom{\widehat{p}(x)} \leftarrow \textit{not } p(x), \textit{not } \widehat{p}(x). \end{array} \tag{25}$$

Theorem 1. *The sentence* $\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\mathrm{MC}[T]]$ *is equivalent to* $T \wedge CC$.

Note that formula (24) is classically equivalent to

$$\forall \mathbf{x}(\widehat{p}(\mathbf{x}) \leftrightarrow \neg p(\mathbf{x})), \tag{26}$$

so that CC can be viewed as the conjunction of explicit definitions of the predicates $\widehat{\mathbf{p}}$ in terms of the predicates $\mathbf{p}$. Since the predicates $\widehat{\mathbf{p}}$ do not belong to the language of T, Theorem 1 shows that $\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\mathrm{MC}[T]]$ is a definitional, and consequently conservative, extension of T. In other words, the $\mathbf{p}\widehat{\mathbf{p}}$-stable models of MC[$T$] are identical to the models of T extended by the interpretations of the predicates $\widehat{\mathbf{p}}$ given by explicit definitions (26).

Note also that in this characterization of the stable models of MC[T] the set of intensional predicates includes both the explainable predicates $\mathbf{p}$ of T and the corresponding predicates $\widehat{\mathbf{p}}$.

[4] The symbol $\widetilde{\forall}$ denotes universal closure.

4.2 A First-Order Counterpart of Ferraris's Translation

In this section, T is a causal theory in the sense of Section 3.1 such that

- the head of each rule of T is a disjunction of literals, and
- all predicate constants occurring in the heads of rules are explainable.

In other words, we assume that every rule of T has the form

$$\bigvee_{A\in Pos} A \vee \bigvee_{A\in Neg} \neg A \Leftarrow G \tag{27}$$

for some sets *Pos*, *Neg* of atomic formulas that contain a predicate constant from the set $\mathbf{p}$ of explainable symbols.

As in Section 4.1, for each $p \in \mathbf{p}$ we choose a new predicate constant $\widehat{p}$ of the same arity. If A is an atomic formula $p(\mathbf{t})$, where $p \in \mathbf{p}$ and $\mathbf{t}$ is a tuple of terms, then $\widehat{A}$ stands for $\widehat{p}(\mathbf{t})$.

The *Ferraris translation* Fer$[T]$ of T is the conjunction of

- formulas

$$\widetilde{\forall}\left(\neg\neg G \wedge \bigwedge_{A\in Pos}(\widehat{A}\vee\neg\widehat{A}) \wedge \bigwedge_{A\in Neg}(A\vee\neg A) \rightarrow \bigvee_{A\in Pos} A \vee \bigvee_{A\in Neg} \widehat{A}\right) \tag{28}$$

  for all rules (27) of T, and
- completeness constraints CC.

For instance, if T is (23) then MC$[T]$ is the conjunction of the formulas

$$\begin{array}{l}\neg\neg\top \wedge (\widehat{p}(a) \vee \neg\widehat{p}(a)) \rightarrow p(a)\\ \forall x(\neg\neg\neg p(x) \wedge (p(x) \vee \neg p(x)) \rightarrow \widehat{p}(x))\\ \forall x \neg(p(x) \leftrightarrow \widehat{p}(x)).\end{array}$$

Theorem 2. *The sentence* $\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\mathrm{Fer}[T]]$ *is equivalent to* $T \wedge CC$.

Thus the $\mathbf{p}\widehat{\mathbf{p}}$-stable models of Fer$[T]$ are identical to the models of T extended by the interpretations of the predicates $\widehat{\mathbf{p}}$ given by explicit definitions (26).

5 Proofs

Our proofs of Theorems 1 and 2 are quite different from the published proofs of similar results for the propositional case, because of the difference between the semantics used in this paper (Section 3) and the semantics of propositional causal theories and logic programs, which are based on reducts.

We will prove Theorem 2 first.

5.1 Proof of Theorem 2

Let $\varPi$ be the conjunction of the implications

$$\widetilde{\forall}\left(\neg\neg G \wedge \bigwedge_{A\in Pos}(\widehat{A}\vee\neg\widehat{A}) \wedge \bigwedge_{A\in Neg}(A\vee\neg A) \rightarrow \bigvee_{A\in Pos} A \vee \bigvee_{A\in Neg}\widehat{A}\right)$$

for all rules (27) in T. Then $\mathrm{Fer}[T]$ is $\varPi\wedge CC$. Since CC has no strictly positive occurrences of intensional predicates, $\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\mathrm{Fer}[T]]$ is equivalent to

$$\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\varPi]\wedge CC$$

[Ferraris *et al.*, 2010, Section 5.1]. Therefore the statement of Theorem 2 is equivalent to the claim that CC entails

$$\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\varPi]\leftrightarrow T.$$

We will establish this claim by proving several lemmas. The key steps are Lemma 5 (one half of the equivalence) and Lemma 8 (the other half).

In the statement of the following lemma, $\neg\mathbf{p}$ stands for the list of predicate expressions[5] $\lambda\mathbf{x}\neg p(\mathbf{x})$, where $\mathbf{x}$ is a list of distinct object variables, for all p from $\mathbf{p}$. By $\upsilon\mathbf{p}$, $\upsilon\widehat{\mathbf{p}}$ we denote the lists of predicate variables used in the second-order formula $\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\varPi]$ (see Section 3.2).

Lemma 1. *Formula* $(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})<(\mathbf{p},\neg\mathbf{p})$ *is equivalent to*

$$\bigvee_{p\in\mathbf{p}}(((\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})\le(\mathbf{p},\neg\mathbf{p}))\wedge\exists\mathbf{x}(\neg\upsilon p(\mathbf{x})\wedge\neg\upsilon\widehat{p}(\mathbf{x}))).$$

Proof. Note first that

$$\begin{aligned}
&(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})<(\mathbf{p},\neg\mathbf{p})\\
&\quad\Leftrightarrow((\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})\le(\mathbf{p},\neg\mathbf{p}))\wedge\neg\,((\mathbf{p},\neg\mathbf{p})\le(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}}))\\
&\quad\Leftrightarrow((\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})\le(\mathbf{p},\neg\mathbf{p}))\wedge\textstyle\bigvee_{p\in\mathbf{p}}\exists\mathbf{x}((p(\mathbf{x})\wedge\neg\upsilon p(\mathbf{x}))\vee(\neg p(\mathbf{x})\wedge\neg\upsilon\widehat{p}(\mathbf{x})))\\
&\quad\Leftrightarrow\textstyle\bigvee_{p\in\mathbf{p}}(((\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})\le(\mathbf{p},\neg\mathbf{p}))\wedge\exists\mathbf{x}((p(\mathbf{x})\wedge\neg\upsilon p(\mathbf{x}))\vee(\neg p(\mathbf{x})\wedge\neg\upsilon\widehat{p}(\mathbf{x})))).
\end{aligned}$$

The disjunction after $\exists\mathbf{x}$ is equivalent to

$$(p(\mathbf{x})\vee\neg\upsilon\widehat{p}(\mathbf{x}))\wedge(\neg\upsilon p(\mathbf{x})\vee\neg p(\mathbf{x}))\wedge(\neg\upsilon p(\mathbf{x})\vee\neg\upsilon\widehat{p}(\mathbf{x})).\tag{29}$$

Since $(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})\le(\mathbf{p},\neg\mathbf{p})$ entails

$$\upsilon p(\mathbf{x})\rightarrow p(\mathbf{x})\text{ and }\upsilon\widehat{p}(\mathbf{x})\rightarrow\neg p(\mathbf{x}),$$

the first conjunctive term of (29) can be rewritten as $\neg\upsilon\widehat{p}(\mathbf{x})$, and the second term as $\neg\upsilon p(\mathbf{x})$, so that (29) will turn into $\neg\upsilon p(\mathbf{x})\wedge\neg\upsilon\widehat{p}(\mathbf{x})$.

[5] See [Lifschitz, 1994, Section 3.1].

For any formula F, by $F_{\Sigma 1}$ we denote the formula

$$F^{(\upsilon\mathbf{p})(\upsilon\widehat{\mathbf{p}})}_{(\upsilon\mathbf{p}\wedge\mathbf{p})(\neg\upsilon\mathbf{p}\wedge\neg\mathbf{p})}$$

where $\upsilon\mathbf{p}\wedge\mathbf{p}$ is understood as the list of predicate expressions $\lambda\mathbf{x}(\upsilon p(\mathbf{x})\wedge p(\mathbf{x}))$ for all $p\in\mathbf{p}$, and $\neg\upsilon\mathbf{p}\wedge\neg\mathbf{p}$ is understood in a similar way.

Lemma 2. *Formula*

$$((\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})<(\mathbf{p},\neg\mathbf{p}))_{\Sigma 1}$$

is equivalent to $\upsilon\mathbf{p}\neq\mathbf{p}$.

Proof. In view of Lemma 1, $((\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})<(\mathbf{p},\neg\mathbf{p}))_{\Sigma 1}$ is equivalent to the disjunction of the formulas

$$\begin{aligned}\left(\textstyle\bigwedge_{p\in\mathbf{p}}\forall\mathbf{x}(\upsilon p(\mathbf{x})\rightarrow p(\mathbf{x}))_{\Sigma 1}\right)\wedge\left(\textstyle\bigwedge_{p\in\mathbf{p}}\forall\mathbf{x}(\upsilon\widehat{p}(\mathbf{x})\rightarrow\neg p(\mathbf{x}))_{\Sigma 1}\right)&\\ \wedge\,\exists\mathbf{x}(\neg\upsilon p(\mathbf{x})\wedge\neg\upsilon\widehat{p}(\mathbf{x}))_{\Sigma 1}&\end{aligned}\tag{30}$$

for all $p\in\mathbf{p}$. It is easy to verify that

$$(\upsilon p(\mathbf{x})\rightarrow p(\mathbf{x}))_{\Sigma 1}=(\upsilon p(\mathbf{x})\wedge p(\mathbf{x})\rightarrow p(\mathbf{x}))\Leftrightarrow\top,$$

$$(\upsilon\widehat{p}(\mathbf{x})\rightarrow\neg p(\mathbf{x}))_{\Sigma 1}=(\neg\upsilon p(\mathbf{x})\wedge\neg p(\mathbf{x})\rightarrow\neg p(\mathbf{x}))\Leftrightarrow\top,$$

$$\begin{aligned}(\neg\upsilon p(\mathbf{x})\wedge\neg\upsilon\widehat{p}(\mathbf{x}))_{\Sigma 1}&\Leftrightarrow((\neg\upsilon p(\mathbf{x})\vee\neg p(\mathbf{x}))\wedge\neg(\neg\upsilon p(\mathbf{x})\wedge\neg p(\mathbf{x})))\\&\Leftrightarrow(\upsilon p(\mathbf{x})\leftrightarrow\neg p(\mathbf{x}))\\&\Leftrightarrow\neg(\upsilon p(\mathbf{x})\leftrightarrow p(\mathbf{x})).\end{aligned}$$

Therefore (30) is equivalent to $\exists\mathbf{x}\neg(\upsilon p(\mathbf{x})\leftrightarrow p(\mathbf{x}))$, so that the disjunction of all formulas (30) is equivalent to $\upsilon\mathbf{p}\neq\mathbf{p}$.

If A is an atomic formula $p(\mathbf{t})$, where $p\in\mathbf{p}$ and $\mathbf{t}$ is a tuple of terms, then we will write υA for $\upsilon p(\mathbf{t})$, and $\widehat{A}$ for $\upsilon\widehat{p}(\mathbf{t})$. By $\widetilde{\forall}_{obj}F$ we denote the formula $\forall\mathbf{x}F$, where $\mathbf{x}$ is list of all free object variables of F ("object-level universal closure").

Define $H(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})$ be the conjunction of the implications

$$\widetilde{\forall}_{obj}\left(G\rightarrow\bigvee_{A\in Pos}((\upsilon\widehat{A}\vee A)\rightarrow\upsilon A)\vee\bigvee_{A\in Neg}((\upsilon A\vee\neg A)\rightarrow\upsilon\widehat{A})\right)\tag{31}$$

for all rules (27) in T.

Lemma 3. *Formula CC entails*

$$\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\mathit{\Pi}]\leftrightarrow\mathit{\Pi}\wedge\forall(\upsilon\mathbf{p})(\upsilon\widehat{\mathbf{p}})(((\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})<(\mathbf{p},\widehat{\mathbf{p}}))\rightarrow\neg H(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})).\tag{32}$$

Proof. Recall that $\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\mathit{\Pi}]$ is

$$\mathit{\Pi}\wedge\neg\exists(\upsilon\mathbf{p})(\upsilon\widehat{\mathbf{p}})(((\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})<(\mathbf{p},\widehat{\mathbf{p}}))\wedge\mathit{\Pi}^*(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})).$$

This formula is equivalent to

$$\mathit{\Pi} \wedge \forall(\upsilon\mathbf{p})(\upsilon\widehat{\mathbf{p}})(((\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}}) < (\mathbf{p},\widehat{\mathbf{p}})) \rightarrow \neg\mathit{\Pi}^*(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})).$$

We will conclude the proof by showing that under the assumptions

$$CC \text{ and } (\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}}) < (\mathbf{p},\widehat{\mathbf{p}}) \tag{33}$$

$\mathit{\Pi}^*(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})$ can be equivalently transformed into $\mathit{\Pi} \wedge H(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})$.

Note first that $\mathit{\Pi}^*(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})$ is equivalent to the conjunction of $\mathit{\Pi}$ and the formulas

$$\widetilde{\forall}_{obj}\left(\left(\neg\neg G \wedge \bigwedge_{A\in Pos}(\widehat{A} \vee \neg\widehat{A}) \wedge \bigwedge_{A\in Neg}(A \vee \neg A)\right)^* \rightarrow \left(\bigvee_{A\in Pos} A \vee \bigvee_{A\in Neg} \widehat{A}\right)^*\right)$$

for all rules (27) in T. Under assumptions (33), these formulas can be equivalently rewritten as

$$\widetilde{\forall}_{obj}\left(G \wedge \bigwedge_{A\in Pos}((\upsilon\widehat{A} \vee \neg\widehat{A})) \wedge \bigwedge_{A\in Neg}((\upsilon A \vee \neg A)) \rightarrow \bigvee_{A\in Pos} \upsilon A \vee \bigvee_{A\in Neg} \upsilon\widehat{A}\right)$$

[Ferraris *et al.*, 2010, Section A.3, Lemma 6], and further as

$$\widetilde{\forall}_{obj}\left(G \wedge \bigwedge_{A\in Pos}((\upsilon\widehat{A} \vee A)) \wedge \bigwedge_{A\in Neg}((\upsilon A \vee \neg A)) \rightarrow \bigvee_{A\in Pos} \upsilon A \vee \bigvee_{A\in Neg} \upsilon\widehat{A}\right).$$

This formula is equivalent to

$$\widetilde{\forall}_{obj}\left(G \rightarrow \bigvee_{A\in Pos} \neg(\upsilon\widehat{A} \vee A) \vee \bigvee_{A\in Neg} \neg(\upsilon A \vee \neg A) \vee \bigvee_{A\in Pos} \upsilon A \vee \bigvee_{A\in Neg} \upsilon\widehat{A}\right),$$

and consequently to

$$\widetilde{\forall}_{obj}\left(G \rightarrow \bigvee_{A\in Pos} (\neg(\upsilon\widehat{A} \vee A) \vee \upsilon A) \vee \bigvee_{A\in Neg} (\neg(\upsilon A \vee \neg A) \vee \upsilon\widehat{A})\right),$$

which is equivalent to (31).

Lemma 4. *$T^\dagger(\upsilon\mathbf{p})$ is equivalent to $H(\upsilon\mathbf{p},\upsilon\widehat{\mathbf{p}})_{\Sigma 1}$.*

Proof. Formula $T^\dagger(\upsilon\mathbf{p})$ is the conjunction of the formulas

$$\widetilde{\forall}_{obj}\left(G \rightarrow \left(\bigvee_{A\in Pos} \upsilon A \vee \bigvee_{A\in Neg} \neg\upsilon A\right)\right) \tag{34}$$

for all rules (27) in T. On the other hand, $H(\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}})_{\Sigma 1}$ is the conjunction of the formulas

$$\widetilde{\forall}_{obj}\left(G \to \bigvee_{A \in Pos} ((\upsilon\widehat{A} \vee A) \to \upsilon A)_{\Sigma 1} \vee \bigvee_{A \in Neg} ((\upsilon A \vee \neg A) \to \upsilon\widehat{A})_{\Sigma 1}\right) \qquad (35)$$

for all rules (27) in T. It remains to observe that

$$\begin{aligned} ((\upsilon\widehat{A} \vee A) \to \upsilon A)_{\Sigma 1} &= (\neg\upsilon A \wedge \neg A) \vee A \to \upsilon A \wedge A \\ &\Leftrightarrow \neg\upsilon A \vee A \to \upsilon A \wedge A \\ &\Leftrightarrow (\upsilon A \wedge \neg A) \vee (\upsilon A \wedge A) \\ &\Leftrightarrow \upsilon A, \end{aligned}$$

and that, similarly, $((\upsilon A \vee \neg A) \to \upsilon\widehat{A})_{\Sigma 1}$ is equivalent to $\neg\upsilon A$.

Lemma 5. *CC*, $\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\Pi] \models T$.

Proof. By Lemma 3, the assumptions CC and $\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\Pi]$ entail

$$\Pi \wedge \forall(\upsilon\mathbf{p})(\upsilon\widehat{\mathbf{p}})(((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) < (\mathbf{p}, \widehat{\mathbf{p}})) \to \neg H(\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}})). \qquad (36)$$

The first conjunctive term is equivalent to $T^{\dagger}(\mathbf{p})$. From the second conjunctive term we conclude:

$$\forall\upsilon\mathbf{p}(((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) < (\mathbf{p}, \widehat{\mathbf{p}}))_{\Sigma 1} \to \neg H(\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}})_{\Sigma 1}).$$

By Lemma 2 and Lemma 4, this formula is equivalent to

$$\forall\upsilon\mathbf{p}((\upsilon\mathbf{p} \neq \mathbf{p}) \to \neg T^{\dagger}(\upsilon\mathbf{p})),$$

and consequently to

$$\forall\upsilon\mathbf{p}(T^{\dagger}(\upsilon\mathbf{p}) \to (\upsilon\mathbf{p} = \mathbf{p})).$$

The conjunction of the last formula with $T^{\dagger}(\mathbf{p})$ is equivalent to (18).

For any formula F, by $F_{\Sigma 2}$ we denote the formula

$$F^{\upsilon\mathbf{p}}_{(((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) \leq (\mathbf{p}, \neg\mathbf{p})) \wedge \neg\upsilon\mathbf{p} \wedge \neg\upsilon\widehat{\mathbf{p}}) \leftrightarrow \neg\mathbf{p}}$$

where the subscript

$$(((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) \leq (\mathbf{p}, \neg\mathbf{p})) \wedge \neg\upsilon\mathbf{p} \wedge \neg\upsilon\widehat{\mathbf{p}}) \leftrightarrow \neg\mathbf{p}$$

is understood as the list of predicate expressions

$$\lambda\mathbf{x}((((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) \leq (\mathbf{p}, \neg\mathbf{p})) \wedge \neg\upsilon p(\mathbf{x}) \wedge \neg\upsilon\widehat{p}(\mathbf{x})) \leftrightarrow \neg p(\mathbf{x}))$$

for all $p \in \mathbf{p}$.

Lemma 6. *Formula*

$$(\upsilon\mathbf{p} \neq \mathbf{p})_{\Sigma 2}$$

is equivalent to $(\upsilon\mathbf{p}, \widehat{\mathbf{p}}) < (\mathbf{p}, \neg\mathbf{p})$*.*

Proof. Formula $(\upsilon\mathbf{p} \neq \mathbf{p})_{\Sigma 2}$ is equivalent to

$$\bigvee_{p \in \mathbf{p}} \exists\mathbf{x}(\upsilon p(\mathbf{x}) \leftrightarrow \neg p(\mathbf{x}))_{\Sigma 2}$$

that is,

$$\bigvee_{p \in \mathbf{p}} \exists\mathbf{x}((((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) \leq (\mathbf{p}, \neg\mathbf{p})) \wedge (\neg\upsilon p(\mathbf{x}) \wedge \neg\upsilon\widehat{p}(\mathbf{x})) \leftrightarrow \neg p(\mathbf{x})) \leftrightarrow \neg p(\mathbf{x})).$$

This formula can be equivalently rewritten as

$$\bigvee_{p \in \mathbf{p}} (((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) \leq (\mathbf{p}, \neg\mathbf{p})) \wedge \exists\mathbf{x}(\neg\upsilon p(\mathbf{x}) \wedge \neg\upsilon\widehat{p}(\mathbf{x}))),$$

which is equivalent to $(\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) < (\mathbf{p}, \neg\mathbf{p})$ by Lemma 1.

Lemma 7. *The implication*

$$(\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) \leq (\mathbf{p}, \neg\mathbf{p}) \rightarrow (T^{\dagger}(\upsilon\mathbf{p})_{\Sigma 2} \leftrightarrow H(\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}))$$

is logically valid.

Proof. Recall that $T^{\dagger}(\upsilon\mathbf{p})$ is the conjunction of implications (34) for all rules (27) in T. Consequently $T^{\dagger}(\upsilon\mathbf{p})_{\Sigma 2}$ is the conjunction of the formulas

$$\widetilde{\forall}_{obj}\left(G \rightarrow \bigvee_{A \in Pos} (\upsilon A)_{\Sigma 2} \vee \bigvee_{A \in Neg} \neg(\upsilon A)_{\Sigma 2}\right),$$

that is to say,

$$\begin{gathered}\widetilde{\forall}_{obj}(G \rightarrow \textstyle\bigvee_{A \in Pos}((((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) \leq (\mathbf{p}, \neg\mathbf{p})) \wedge \neg\upsilon A \wedge \neg\upsilon\widehat{A}) \leftrightarrow \neg A) \vee \\ \textstyle\bigvee_{p \in Neg}((((\mathbf{p}, \widehat{\mathbf{p}}) \leq (\mathbf{p}, \neg\mathbf{p})) \wedge \neg\upsilon A \wedge \neg\upsilon\widehat{A}) \leftrightarrow A)).\end{gathered}$$

Under the assumption

$$(\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) \leq (\mathbf{p}, \neg\mathbf{p}) \tag{37}$$

the last formula can be equivalently rewritten as

$$\widetilde{\forall}_{obj}\left(G \rightarrow \bigvee_{A \in Pos} ((\upsilon A \vee \upsilon\widehat{A}) \leftrightarrow A) \vee \bigvee_{A \in Neg} \neg((\upsilon A \vee \upsilon\widehat{A}) \leftrightarrow A)\right).$$

It remains to check that, under assumption (37),

$$(\upsilon A \vee \upsilon\widehat{A}) \leftrightarrow A \tag{38}$$

can be equivalently rewritten as

$$\upsilon\widehat{A} \vee A \rightarrow \upsilon A, \tag{39}$$

and

$$\upsilon A \vee \upsilon\widehat{A} \leftrightarrow \neg A \tag{40}$$

can be rewritten as

$$\upsilon A \vee \neg A \rightarrow \upsilon\widehat{A}. \tag{41}$$

Formula (38) is equivalent to

$$(\upsilon A \rightarrow A) \wedge (\upsilon\widehat{A} \rightarrow A) \wedge (A \rightarrow \upsilon A \vee \upsilon\widehat{A}). \tag{42}$$

Since assumption (37) entails $\upsilon A \rightarrow A$ and $\upsilon\widehat{A} \rightarrow \neg A$, formula (42) can be rewritten as

$$\neg\upsilon\widehat{A} \wedge (A \rightarrow \upsilon A). \tag{43}$$

Formula (39) is equivalent to

$$(\upsilon\widehat{A} \rightarrow \upsilon A) \wedge (A \rightarrow \upsilon A),$$

which, under assumption (37), can be rewritten as (43) as well. In a similar way, each of the formulas (40), (41) can be transformed into

$$\neg\upsilon A \wedge (\neg A \rightarrow \upsilon\widehat{A}).$$

Lemma 8. *CC, $T \models \mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\varPi]$.*

Proof. Recall that T is equivalent to

$$T^{\dagger}(\mathbf{p}) \wedge \forall\upsilon\mathbf{p}(T^{\dagger}(\upsilon\mathbf{p}) \rightarrow (\upsilon\mathbf{p} = \mathbf{p})).$$

The first conjunctive term is equivalent to $\varPi$. From the second conjunctive term we conclude:

$$\forall(\upsilon\mathbf{p})(\upsilon\widehat{\mathbf{p}})((\upsilon\mathbf{p} \neq \mathbf{p})_{\varSigma 2} \rightarrow \neg T^{\dagger}(\upsilon\mathbf{p})_{\varSigma 2}).$$

By Lemma 6, this is equivalent to

$$\forall(\upsilon\mathbf{p})(\upsilon\widehat{\mathbf{p}})(((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) < (\mathbf{p}, \neg\mathbf{p})) \rightarrow \neg T^{\dagger}(\upsilon\mathbf{p})_{\varSigma 2})$$

and, by Lemma 7, to

$$\forall(\upsilon\mathbf{p})(\upsilon\widehat{\mathbf{p}})(((\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}}) < (\mathbf{p}, \neg\mathbf{p})) \rightarrow \neg H(\upsilon\mathbf{p}, \upsilon\widehat{\mathbf{p}})).$$

Under assumption CC, the conjunction of this formula with $\varPi$ can be equivalently rewritten as $\mathrm{SM}_{\mathbf{p}\widehat{\mathbf{p}}}[\varPi]$.

Theorem 2 is immediate from Lemmas 5 and 8.

5.2 Proof of Theorem 1

Theorem 1 follows from Theorem 2 in view of the following fact:

Lemma 9. *For any causal theory T consisting of rules of the forms (21) and (22),* MC$[T]$ *is intuitionistically equivalent to* Fer$[T]$.

Proof. Formula MC$[T]$ can be obtained from Fer$[T]$ by replacing some subformulas of the form

$$\neg\neg G \wedge (\widehat{p}(\mathbf{t}) \vee \neg\widehat{p}(\mathbf{t})) \rightarrow p(\mathbf{t})$$

with

$$\neg\neg G \rightarrow p(\mathbf{t}),$$

and some subformulas of the form

$$\neg\neg G \wedge (p(\mathbf{t}) \vee \neg p(\mathbf{t})) \rightarrow \widehat{p}(\mathbf{t})$$

with

$$\neg\neg G \rightarrow \widehat{p}(\mathbf{t}).$$

Since CC intuitionistically entails

$$\neg(p(\mathbf{t}) \leftrightarrow \widehat{\mathbf{p}}(\mathbf{t})), \tag{44}$$

it is sufficient to check that (44) and

$$\widehat{p}(\mathbf{t}) \vee \neg\widehat{p}(\mathbf{t}) \rightarrow p(\mathbf{t}) \tag{45}$$

intuitionistically entail $p(\mathbf{t})$, and that (44) and

$$p(\mathbf{t}) \vee \neg p(\mathbf{t}) \rightarrow \widehat{p}(\mathbf{t}) \tag{46}$$

intuitionistically entail $\widehat{p}(\mathbf{t})$.

Since (45) is classically equivalent to $p(\mathbf{t})$, it is easy to see that (44) and (45) classically entail $\neg\widehat{p}(\mathbf{t})$. By Glivenko's theorem,[6] it follows that this entailment holds intuitionsitically as well. Since $p(\mathbf{t})$ is an intuitionistic consequence of (45) and $\neg\widehat{p}(\mathbf{t})$, we can conclude that $p(\mathbf{t})$ is an intuitionistic consequence of (44) and (45).

The proof of the fact that $\widehat{p}(\mathbf{t})$ is an intuitionistic consequence of (44) and (46) is similar.

[6] This theorem [Mints, 2000, Theorem 3.1] asserts that if a formula beginning with negation can be derived from a set Γ of formulas in classical propositional logic then it can be derived in intuitionistic propositional logic too.

6 Conclusion

The definition of a stable model based on the operator SM, reviewed in Section 3.2, is more general than the traditional defintion [Gelfond and Lifschitz, 1988] in several ways. It is more general syntactically, because it is applicable to formulas containing quantifiers. It is more general semantically, in the sense that it is applicable to non-Herbrand models. It also allows us to distinguish between intensional and extensional predicates. Ferraris *et al.* [2010] argued that these features can be useful in applications to knowledge representation. They showed how to extend many familiar properties of stable models to the first-order case. This line of work was continued in [Lee *et al.*, 2008] and [Ferraris *et al.*, 2009], and the theorems presented in this paper belong to the same direction of research.

We expect that Theorem 2 will help us extend the theorem on synonymity proved in [Lee *et al.*, 2010] to first-order causal theories, and that it will help us in this way to design a new implementation of modular action language MAD [Erdoğan, 2008, Ren, 2009].

In application to causal theories with variables, the translations defined in this paper generate usually logic programs that are not safe, and consequently cannot be processed by existing answer set solvers.[7] For instance, the second rule of (25) is unsafe, because the only occurrence of x in its body is in the scope of negation as failure. It may be possible to find modifications of the McCain and Ferraris translations that produce safe logic programs in practically important cases.

Acknowledgements

This research was partially supported by the National Science Foundation under grant IIS-0712113.

References

[Erdoğan, 2008] Selim T. Erdoğan. *A Library of General-Purpose Action Descriptions.*[8]. PhD thesis, University of Texas at Austin, 2008.

[Ferraris *et al.*, 2007] Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz. A new perspective on stable models. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 372–379, 2007.

[Ferraris *et al.*, 2009] Paolo Ferraris, Joohyung Lee, Vladimir Lifschitz, and Ravi Palla. Symmetric splitting in the general theory of stable models. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 797–803, 2009.

[Ferraris *et al.*, 2010] Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz. Stable models and circumscription.[9]. *Artificial Intelligence*, 2010. To appear.

[7] See Chapter 3 of the DLV manual, `http://www.dbai.tuwien.ac.at/proj/dlv/man/`.

[8] `http://www.cs.utexas.edu/users/tag/mad/erdogan-dissertation.pdf`

[9] `http://peace.eas.asu.edu/joolee/papers/smcirc.pdf`

[Ferraris, 2005] Paolo Ferraris. Answer sets for propositional theories. In *Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pages 119–131, 2005.

[Ferraris, 2007] Paolo Ferraris. A logic program characterization of causal theories. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 366–371, 2007.

[Gelfond and Lifschitz, 1988] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Robert Kowalski and Kenneth Bowen, editors, *Proceedings of International Logic Programming Conference and Symposium*, pages 1070–1080. MIT Press, 1988.

[Gelfond and Lifschitz, 1991] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[Giunchiglia and Lifschitz, 1998] Enrico Giunchiglia and Vladimir Lifschitz. An action language based on causal explanation: Preliminary report. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, pages 623–630. AAAI Press, 1998.

[Giunchiglia *et al.*, 2004] Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner. Nonmonotonic causal theories. *Artificial Intelligence*, 153(1–2):49–104, 2004.

[Lee *et al.*, 2008] Joohyung Lee, Vladimir Lifschitz, and Ravi Palla. Safe formulas in the general theory of stable models (preliminary report). In *Proceedings of International Conference on Logic Programming (ICLP)*, pages 672–676, 2008.

[Lee *et al.*, 2010] Joohyung Lee, Yuliya Lierler, Vladimir Lifschitz, and Fangkai Yang. Representing synonymity in causal logic and in logic programming.[10]. In *Proceedings of International Workshop on Nonmonotonic Reasoning (NMR)*, 2010.

[Lifschitz and Ren, 2006] Vladimir Lifschitz and Wanwan Ren. A modular action description language. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, pages 853–859, 2006.

[Lifschitz and Ren, 2007] Vladimir Lifschitz and Wanwan Ren. The semantics of variables in action descriptions. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 2007.

[Lifschitz, 1994] Vladimir Lifschitz. Circumscription. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in AI and Logic Programming*, volume 3, pages 298–352. Oxford University Press, 1994.

[Lifschitz, 1997] Vladimir Lifschitz. On the logic of causal explanation. *Artificial Intelligence*, 96:451–465, 1997.

[McCain and Turner, 1997] Norman McCain and Hudson Turner. Causal theories of action and change. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, pages 460–465, 1997.

[McCain, 1997] Norman McCain. *Causality in Commonsense Reasoning about Actions.*[11]. PhD thesis, University of Texas at Austin, 1997.

[Mints, 2000] Grigori Mints. *A short introduction to intuitionistic logic*. Kluwer, 2000.

[Ren, 2009] Wanwan Ren. *A Modular Language for Describing Actions.*[12]. PhD thesis, University of Texas at Austin, 2009.

[10] http://userweb.cs.utexas.edu/users/vl/papers/syn.pdf

[11] ftp://ftp.cs.utexas.edu/pub/techreports/tr97-25.ps.gz

[12] http://www.cs.utexas.edu/users/rww6/dissertation.pdf