Functional Completion

Vladimir Lifschitz and Fangkai Yang
Department of Computer Science
University of Texas at Austin
Austin, TX, 78712-0233, USA
{vl,fkyang}@cs.utexas.edu

Abstract

Nonmonotonic causal logic is a knowledge representation language designed for describing domains that involve actions and change. The process of literal completion, similar to program completion familiar from the theory of logic programming, can be used to translate some nonmonotonic causal theories into classical logic. Its applicability is restricted, however, to theories that deal with truth-valued fluents, represented by predicate symbols. In this note we introduce functional completion—a more general process that can be applied to causal theories in which fluents are treated as functions.

1 Introduction

Nonmonotonic causal logic is a knowledge representation language designed for describing domains that involve actions and change. It was used for defining the semantics of action description languages $\mathcal{C}$ [Giunchiglia and Lifschitz, 1998], $\mathcal{C}+$ [Giunchiglia *et al.*, 2004], and MAD [Lifschitz and Ren, 2006]. Its implementation, called the Causal Calculator [McCain, 1997, Lee, 2005, Casolary and Lee, 2011], has been used to solve several challenging commonsense reasoning problems, including problems of nontrivial size [Akman *et al.*, 2004], to provide a group of robots with high-level reasoning [Caldiran *et al.*, 2009], to give executable specifications of norm-governed computational societies [Artikis *et al.*, 2009], and to automate the analysis of business processes under authorization constraints [Armando *et al.*, 2009].

The propositional version of nonmonotonic causal logic [McCain and Turner, 1997] is based on a fixpoint construction involving reducts, similar to the one employed in the original definition of a stable model [Gelfond and Lifschitz, 1988]. The first result on the relationship between the two formalisms [McCain, 1997, Proposition 6.7] is generalized in [Ferraris, 2006]. The first-order version of nonmonotonic causal logic [Lifschitz, 1997] provides additional expressive power that is essential for describing the semantics of action descriptions with variables [Lifschitz and Ren, 2007]. Its semantics is based on a syntactic transformation that turns causal theories into second-order sentences. A somewhat similar syntactic transformation is used in the first-order theory of stable models proposed in [Ferraris *et al.*, 2011]. A first-order counterpart of the result of [Ferraris, 2006] is proved in [Ferraris *et al.*, 2012].

In nonmonotonic causal logic, we distinguish between being true and having a cause. Syntactically, a first-order causal theory consists of "causal rules" $F \Leftarrow G$, where F (the head) and G (the body) are first-order formulas. The rule reads "F is caused if G is true." Some function constants and/or predicate constants of the underlying signature are declared to be "(causally) explainable." For instance, the rule

$$\mathit{at}(x, y, t+1) \Leftarrow \mathit{move}(x, y, t), \tag{1}$$

where the predicate constant *at* is explainable, expresses that there is a cause for the object x to be at place y at time $t + 1$ if x is moved to y at time t. (Executing the move action is the cause.) The rule

$$\mathit{at}(x, y, t+1) \Leftarrow \mathit{at}(x, y, t) \wedge \mathit{at}(x, y, t+1) \tag{2}$$

expresses the commonsense law of inertia for the fluent *at*: if at time $t + 1$ object x is at the same place as at time t then there is a cause for this. (Inertia is the cause; this is how nonmonotonic causal logic solves the frame problem.)

The process of "literal completion," defined in [McCain and Turner, 1997] and extended to the first-order case in [Lifschitz, 1997], allows us, under some conditions, to turn a given causal theory into a formula without second-order quantifiers. This process is similar to Clark's completion familiar from logic programming [Clark, 1978, Lloyd and Topor, 1984], except that it applies to rules that may have both positive and negative literals in their heads, and it generates two equivalences for each explainable predicate constant, "positive" and "negative."

Literal completion is applicable to a theory only if each of its explainable symbols is a predicate constant; function constants are allowed in the signature, but they cannot be explainable. Expainable function symbols are often useful, however. For example, the binary function symbol *loc* can be used to describe locations of objects instead of the ternary predicate symbol *at*. Then rules (1), (2) will turn into

$$\begin{array}{l} \mathit{loc}(x, t+1) = y \Leftarrow \mathit{move}(x, y, t) \\ \mathit{loc}(x, t+1) = y \Leftarrow \mathit{loc}(x, t) = y \wedge \mathit{loc}(x, t+1) = y. \end{array} \tag{3}$$

The advantages of using functional notation in such cases are the same as the advantages of writing $x+y = z$ in formal arithmetic in comparison with $\mathit{sum}(x, y, z)$: there is no need to postulate the existence and uniqueness of the value of the function, and many ideas can be expressed more concisely. For instance, we can write

$$\mathit{loc}(x_1, t) = \mathit{loc}(x_2, t)$$

instead of

$$\exists y(at(x_1, y, t) \wedge at(x_2, y, t)).$$

Our goal here is to extend the definition of literal completion and the theorem on literal completion from [Lifschitz, 1997, Section 5] to causal theories with explainable function symbols.

We are very pleased to contribute this note to a collection honoring Professor David Pearce, a dear friend and respected colleague who has made outstanding contributions to the field of nonmonotonic logic.

2 Review of Causal Logic

2.1 Syntax and Semantics

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

- a list **c** of distinct function and/or predicate constants,[1] called the *explainable symbols* of T, and
- a finite set of *causal rules* of the form $F \Leftarrow G$, where F and G are first-order formulas.

The semantics of causal theories is defined by a syntactic transformation that is somewhat similar to circumscription [McCarthy, 1986]; its result is usually a second-order formula. For each member c of **c**, choose a new variable υc similar to c,[2] and let $\upsilon\mathbf{c}$ stand for the list of all these variables. By $T^\dagger(\upsilon\mathbf{c})$ we denote the conjunction of the formulas

$$\forall\mathbf{x}(G \rightarrow F^{\mathbf{c}}_{\upsilon\mathbf{c}}) \tag{4}$$

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

$$\forall\upsilon\mathbf{c}(T^\dagger(\upsilon\mathbf{c}) \leftrightarrow (\upsilon\mathbf{c} = \mathbf{c})). \tag{5}$$

(By $\upsilon\mathbf{c} = \mathbf{c}$ we denote the conjunction of the formulas $\upsilon c = c$ for all members c of the tuple **c**.) Accordingly, by a model of the causal theory T we understand a model of (5) in the sense of classical logic. The models of T are characterized, informally speaking, by the fact that the interpretation of the explainable symbols **c** in the model is the only interpretation of these symbols that is "causally explained" by the rules of T.

[1] We view object constants as function constants of arity 0, so that they are allowed in **c**. Similarly, propositional symbols are viewed as predicate constants of arity 0. Equality, on the other hand, may not be included in c.

[2] That is to say, if c is a function constant then υc should be a function variable of the same arity; if c is a predicate constant then υc should be a predicate variable of the same arity.

2.2 Examples

Causal theory T_0 has two rules:

$$\begin{aligned} p(x) &\Leftarrow q(x), \\ \neg p(x) &\Leftarrow \neg p(x), \end{aligned} \tag{6}$$

and the predicate constant p is explainable. The second rule, "if $p(x)$ is false then there is a cause for this," expresses, in the language of causal logic, the closed world assumption for p. According to the semantics of causal logic, T_0 is shorthand for the sentence

$$\forall \upsilon p(\forall x(q(x) \rightarrow \upsilon p(x)) \wedge \forall x(\neg p(x) \rightarrow \neg \upsilon p(x)) \;\leftrightarrow\; \upsilon p = p),$$

where υp is a predicate variable. This formula is logically equivalent to

$$\forall x(p(x) \leftrightarrow q(x)). \tag{7}$$

Causal theory T_1 has the rules

$$\begin{aligned} \bot &\Leftarrow a = b, \\ c = a &\Leftarrow c = a, \\ c = b &\Leftarrow q, \end{aligned}$$

and the object constant c is explainable.[3] The first rule of T_1 says that a is different from b. The second rule ("if $c = a$ then there is a cause for this") expresses, in the language of causal logic, that by default $c = a$. The last rule says that there is a cause for c to be equal to b if q is true. Theory T_1 is shorthand for the sentence

$$\forall \upsilon c((a = b \rightarrow \bot) \wedge (c = a \rightarrow \upsilon c = a) \wedge (q \rightarrow \upsilon c = b) \;\leftrightarrow\; \upsilon c = c)$$

where υc is an object variable. This formula is equivalent to

$$a \neq b \wedge (q \rightarrow c = b) \wedge (\neg q \rightarrow c = a). \tag{8}$$

The second conjunctive term shows that if q holds then the value of c is different from its default value a.

In the next example, we describe the commonsense domain mentioned in the introduction: the effect of moving objects on their locations. For simplicity, we only consider the time instants 0, 1 and the execution of the move action at time 0. On the other hand, we would like to take into account the fact (glossed over in the introduction) that the domain involves things of several kinds: movable objects, places, and time instants. To this end, we include the auxiliary symbol *none*, which is used as the value of *loc*(x, t) when the arguments are "not of the right kind" (that

[3] By $\bot$ and $\top$ we denote the 0-place connectives *false* and *true*.

is, when x is not a movable object or when t is not a time instant). The rules of the causal theory T_2 are

$$\begin{aligned}
\bot &\Leftarrow 0 = 1, \\
\bot &\Leftarrow 0 = \mathit{none}, \\
\bot &\Leftarrow 1 = \mathit{none}, \\
\mathit{obj}(x) \wedge \mathit{place}(y) &\Leftarrow \mathit{move}(x, y), \\
\mathit{loc}(x, 0) = y &\Leftarrow \mathit{loc}(x, 0) = y \wedge \mathit{obj}(x) \wedge \mathit{place}(y), \\
\mathit{loc}(x, 1) = y &\Leftarrow \mathit{move}(x, y), \\
\mathit{loc}(x, 1) = y &\Leftarrow \mathit{loc}(x, 0) = y \wedge \mathit{loc}(x, 1) = y \wedge \mathit{obj}(x) \wedge \mathit{place}(y), \\
\mathit{loc}(x, t) = \mathit{none} &\Leftarrow \neg \mathit{obj}(x), \\
\mathit{loc}(x, t) = \mathit{none} &\Leftarrow t \neq 0 \wedge t \neq 1,
\end{aligned}$$

and the function constant *loc* is explainable. The rule with $\mathit{loc}(x, 0)$ in the head allows an object x to be initially anywhere: whichever place is the value of $\mathit{loc}(x, 0)$, there is a cause for that. The next two rules describe the effect of moving objects and the inertia property of locations. According to the semantics of causal logic, T_2 is shorthand for the formula

$$\forall \upsilon \mathit{loc}(T_2^\dagger(\upsilon \mathit{loc}) \leftrightarrow (\upsilon \mathit{loc} = \mathit{loc})),$$

where $\upsilon \mathit{loc}$ is a binary function variable. In Section 4 we will see how functional completion allows us to rewrite this formula without second-order quantifiers.

2.3 Literal Completion

The definition of the literal completion of a causal theory in [Lifschitz, 1997] assumes that each rule of the theory is definite, which means that the head of the rule is a literal or doesn't contain explainable symbols. In this review, we impose a more restrictive condition, similar to the definition of Clark normal form in [Ferraris *et al.*, 2011, Section 6.1]. This is not a significant limitation, because any definite causal theory can be converted to the normal form defined below by equivalent transformations.

Let T be a causal theory such that all its explainable symbols are predicate constants. We say that T is in *Clark normal form* if it consists of

- rules of the form
$$p(\mathbf{x}) \Leftarrow G(\mathbf{x}), \tag{9}$$
one for each explanable predicate symbol p, where $\mathbf{x}$ is a tuple of distinct variables, and $G(\mathbf{x})$ is a formula without any free variables other than the members of $\mathbf{x}$,

- rules of the form
$$\neg p(\mathbf{x}) \Leftarrow G(\mathbf{x}), \tag{10}$$

one for each explanable predicate symbol p, where $\mathbf{x}$ and $G(\mathbf{x})$ are as above, and

- rules without explainable symbols in the head.

For example, T_0 is in Clark normal form.

The *literal completion* of a causal theory T in Clark normal form is the conjunction of the sentences

$$\forall \mathbf{x}(p(\mathbf{x}) \leftrightarrow G(\mathbf{x})) \tag{11}$$

for all rules of T of the form (9), the sentences

$$\forall \mathbf{x}(\neg p(\mathbf{x}) \leftrightarrow G(\mathbf{x})) \tag{12}$$

for all rules of T of the form (10), and the sentences

$$\widetilde{\forall}(G \rightarrow F) \tag{13}$$

(the symbol $\widetilde{\forall}$ denotes the universal closure) for all rules $F \Leftarrow G$ of T without explainable symbols in the head. For example, the literal completion of T_0 consists of two formulas: (7) and the logically valid formula

$$\forall x(\neg p(x) \leftrightarrow \neg p(x)).$$

Completion Theorem from [Lifschitz, 1997, Section 5] shows that any causal theory in Clark normal form is equivalent to its literal completion.

3 Clark Normal Form Extended to Explainable Functions

The definition of Clark normal form is extended to causal theories with explainable functions by adding an extra clause. About a causal theory T we say that it is in *Clark normal form* if it consists of

- rules of the form (9), one for each explanable predicate symbol p,
- rules of the form (10), one for each explanable predicate symbol p,
- rules of the form

$$f(\mathbf{x}) = y \Leftarrow G(\mathbf{x}, y), \tag{14}$$

  one for each explanable function symbol f, where $\mathbf{x}, y$ is a tuple of distinct variables, and $G(\mathbf{x}, y)$ is a formula without any free variables other than the members of $\mathbf{x}, y$,

- rules without explainable symbols in the head.

In many cases, a causal theory can be transformed into an equivalent causal theory in Clark normal form. For instance, T_1 (see Section 2.2) can be converted to Clark normal form by rewriting its last two rules as

$$\begin{array}{l} c = x \Leftarrow x = a \wedge c = a, \\ c = x \Leftarrow x = b \wedge q \end{array}$$

and then merging them into one rule:

$$c = x \Leftarrow (x = a \wedge c = a) \vee (x = b \wedge q). \tag{15}$$

It is clear that the part of $T_1^\dagger(\upsilon c)$ contributed by the last two rules of T_1 is logically equivalent to the part contributed by (15). Similarly, the Clark normal form of T_2 is

$$\begin{array}{rl} \bot & \Leftarrow 0 = 1, \\ \bot & \Leftarrow 0 = \mathit{none}, \\ \bot & \Leftarrow 1 = \mathit{none}, \\ \mathit{obj}(x) \wedge \mathit{place}(y) & \Leftarrow \mathit{move}(x, y), \\ \mathit{loc}(x, t) = y & \Leftarrow (t = 0 \wedge \mathit{loc}(x, 0) = y \wedge \mathit{obj}(x) \wedge \mathit{place}(y)) \\ & \quad \vee (t = 1 \wedge \mathit{move}(x, y)) \\ & \quad \vee (t = 1 \wedge \mathit{loc}(x, 0) = y \wedge \mathit{loc}(x, 1) = y \wedge \mathit{obj}(x) \wedge \mathit{place}(y)) \\ & \quad \vee (y = \mathit{none} \wedge \neg \mathit{obj}(x)) \\ & \quad \vee (y = \mathit{none} \wedge t \neq 0 \wedge t \neq 1). \end{array} \tag{16}$$

4 Literal Completion Extended to Explainable Functions

Functional completion is a generalization of literal completion to causal theories in Clark normal form that may include explainable functions. The *functional completion* of a causal theory T in Clark normal form is the conjunction of

- sentences (11) for all rules of T of the form (9),
- sentences (12) for all rules of T of the form (10),
- sentences

  $$\widetilde{\forall}(f(\mathbf{x}) = y \leftrightarrow G(\mathbf{x}, y)) \tag{17}$$

  for all rules of T of the form (14), and
- sentences (13) for all rules $F \Leftarrow G$ of T without explainable symbols in the heads.

We will denote the functional completion of T by FC$[T]$.

Theorem *For any causal theory T in Clark normal form,*

$$\exists x_1 x_2 (x_1 \neq x_2) \tag{18}$$

entails $T \leftrightarrow FC[T]$.

Corollary *If a causal theory in Clark normal form contains a rule of the form $\bot \Leftarrow t_1 = t_2$ then it is equivalent to its functional completion.*

Consider, for instance, theory T_1. As discussed above, its Clark normal form consists of rules (15) and

$$\bot \Leftarrow a = b.$$

Its functional completion is the conjunction of the formulas

$$\forall x (c = x \leftrightarrow (x = a \wedge c = a) \vee (x = b \wedge q))$$

and $a = b \rightarrow \bot$ (that is, $a \neq b$). By the corollary, this conjunction is equivalent to T_1.

The Clark normal form of T_2 is (16). The functional completion of this theory is the conjunction of the formulas

$$\begin{array}{rl}
 & 0 \neq 1, \quad 0 \neq \mathit{none}, \quad 1 \neq \mathit{none}, \\
 & \forall xy(\mathit{move}(x,y) \rightarrow \mathit{obj}(x) \wedge \mathit{place}(y)), \\
\forall xty(\mathit{loc}(x,t) = y & \leftrightarrow (t = 0 \wedge \mathit{loc}(x,0) = y \wedge \mathit{obj}(x) \wedge \mathit{place}(y)) \\
 & \vee (t = 1 \wedge \mathit{move}(x,y)) \\
 & \vee (t = 1 \wedge \mathit{loc}(x,0) = y \wedge \mathit{loc}(x,1) = y \wedge \mathit{obj}(x) \wedge \mathit{place}(y)) \\
 & \vee (y = \mathit{none} \wedge \neg \mathit{obj}(x)) \\
 & \vee (y = \mathit{none} \wedge t \neq 0 \wedge t \neq 1)).
\end{array}$$

By the corollary, this conjunction is equivalent to T_2. Using equivalent transformations in first-order-logic, we can rewrite it as the conjunction of the formulas

$$\begin{array}{c}
0 \neq 1, \quad 0 \neq \mathit{none}, \quad 1 \neq \mathit{none}, \\
\forall xy(\mathit{move}(x,y) \rightarrow \mathit{obj}(x) \wedge \mathit{place}(y)), \\
\forall x(\mathit{obj}(x) \rightarrow \mathit{place}(\mathit{loc}(x,0))), \\
\forall xt((\neg \mathit{obj}(x) \vee (t \neq 0 \wedge t \neq 1)) \rightarrow \mathit{loc}(x,t) = \mathit{none}), \\
\forall xy(\mathit{obj}(x) \rightarrow \\
\quad \mathit{loc}(x,1) = y \leftrightarrow (\mathit{move}(x,y) \vee (\mathit{loc}(x,0) = y \wedge \neg \exists w\, \mathit{move}(x,w)))).
\end{array}$$

The last of these formulas characterizes the location of an object at time 1 in terms of its location at time 0 and the actions that have been executed. In this sense, it is similar to successor state axioms as defined in [Reiter, 1991].

Without the assumption that the theory contains a rule of the form $\bot \Leftarrow t_1 = t_2$ the assertion of the corollary would be incorrect. For instance, consider the causal theory consisting of one rule

$$c = x \Leftarrow \bot,$$

where c is an explainable object constant. This theory is equivalent to $\forall \upsilon c(\upsilon c = c)$; its completion is equivalent to $\bot$.

5 Proof of the Theorem

5.1 A Special Case

We will first prove the theorem from Section 4 for the special case when T consists of a single rule (14), where f is explainable. We need to show that (18) entails the equivalence between

$$\forall \upsilon f(\forall \mathbf{x} y(G(\mathbf{x}, y) \rightarrow \upsilon f(\mathbf{x}) = y) \leftrightarrow \upsilon f = f) \tag{19}$$

and

$$\forall \mathbf{x} y(f(\mathbf{x}) = y \leftrightarrow G(\mathbf{x}, y)). \tag{20}$$

Right-to-left: under assumption (20), formula (19) is equivalent to the logically valid formula

$$\forall \upsilon f(\forall \mathbf{x} y(f(\mathbf{x}) = y \rightarrow \upsilon f(\mathbf{x}) = y) \leftrightarrow \upsilon f = f).$$

Left-to-right: assume (19), that is,

$$\forall \upsilon f(\forall \mathbf{x} y(G(\mathbf{x}, y) \rightarrow \upsilon f(\mathbf{x}) = y) \rightarrow \upsilon f = f) \tag{21}$$

and

$$\forall \mathbf{x} y(G(\mathbf{x}, y) \rightarrow f(\mathbf{x}) = y). \tag{22}$$

The last formula is one half of equivalence (20). It remains to derive the other half, that is, $G(\mathbf{x}, f(\mathbf{x}))$. Assume that for some $\mathbf{x}^0$, $\neg G(\mathbf{x}^0, f(\mathbf{x}^0))$. By (18), there exists a y_0 different from $f(\mathbf{x}^0)$. We will prove that the function υf defined by the condition

$$\upsilon f(\mathbf{x}^0) = y_0 \wedge \forall \mathbf{x}(\mathbf{x} \neq \mathbf{x}^0 \rightarrow \upsilon f(\mathbf{x}) = f(\mathbf{x}))$$

satisfies the antecedent of (21). Assume $G(\mathbf{x}, y)$. Since $\neg G(\mathbf{x}^0, y)$, $\mathbf{x} \neq \mathbf{x}_0$. Then $\upsilon f(\mathbf{x}) = f(\mathbf{x})$. On the other hand, by (22), $f(\mathbf{x}) = y$. Consequently $\upsilon f(\mathbf{x}) = y$; the antecedent of (21) is proved. It follows that the consequent $\upsilon f = f$ holds, so that $y_0 = \upsilon f(\mathbf{x}^0) = f(\mathbf{x}^0)$. This is impossible by the choice of y_0.

5.2 Review: Disjoint Causal Theories

The proof of Completion Theorem in full generality uses the following definition from [Lifschitz, 1997, Section 6]. About causal theories T_1, T_2 with sets $\mathbf{c}_1$, $\mathbf{c}_2$ of explainable symbols we say that they are *disjoint* if

- $\mathbf{c}_1$ is disjoint from $\mathbf{c}_2$, and
- the symbols in $\mathbf{c}_1$ do not occur in the heads of the rules of T_2, and the symbols in $\mathbf{c}_2$ do not occur in the heads of the rules of T_1.

For any pairwise disjoint causal theories $T_1, \ldots, T_m$, define their *union* to be the causal theory obtained by combining their rules and their explainable symbols.

Lemma ([Lifschitz, 1997, Lemma 1]) *The union of pairwise disjoint causal theories* $T_1, \ldots, T_m$ *is equivalent to the conjunction* $T_1 \wedge \ldots \wedge T_m$.

5.3 The General Case

Let T be a causal theory in Clark normal form, and let $f_1, \ldots, f_m$ be its explainable function symbols. For each $i = 1, \ldots, m$, let T_i be the causal theory whose only rule is the rule of T that contains f_i in the head, with f_i as its only explainable symbol. Let T_{m+1} be the causal theory whose rules are the rules of T that do not contain explainable function symbols in their heads, and whose set of explainable symbols is the set of all explainable predicate symbols of T. It is clear that theories $T_1, \ldots, T_m, T_{m+1}$ are pairwise disjoint, and that their union is T. By the lemma from Section 5.2, it follows that T is equivalent to $T_1 \wedge \ldots T_m \wedge T_{m+1}$. According to the special case proved in Section 5.1, (18) entails

$$T_i \leftrightarrow \mathrm{FC}[T_i] \qquad (i = 1, \ldots, m).$$

By the theorem from [Lifschitz, 1997, Section 5] quoted at the end of Section 2.3, T_{m+1} is equivalent to $\mathrm{FC}[T_{m+1}]$. Consequently (18) entails

$$T \leftrightarrow \mathrm{FC}[T_1] \wedge \cdots \wedge \mathrm{FC}[T_m] \wedge \mathrm{FC}[T_{m+1}].$$

It remains to observe that the right-hand side of this equivalence is $\mathrm{FC}[T]$.

6 Conclusion

The process of completion, extended in this paper to fluents represented by function symbols, allows us in some cases to turn a causal theory into an equivalent first-order formula. This possibility is important because, semantically, first-order languages are simpler and better understood than many nonmonotonic languages. The completion process is useful also because it clarifies the relationship between causal logic and monotonic solutions to the frame problem, such as those based on the approach of [Reiter, 1991].

A process similar to functional completion can be applied to logic programs with intensional functions [Bartholomew and Lee, 2012, Theorem 12].

Acknowledgements

Thanks to Selim Erdoğan and Yuliya Lierler for comments on a draft of this note. This research was supported by the National Science Foundation under grant IIS-0712113.

References

[Akman *et al.*, 2004] Varol Akman, Selim Erdoğan, Joohyung Lee, Vladimir Lifschitz, and Hudson Turner. Representing the Zoo World and the Traffic World in the language of the Causal Calculator. *Artificial Intelligence*, 153(1–2):105–140, 2004.

[Armando *et al.*, 2009] Alessandro Armando, Enrico Giunchiglia, and Serena Elisa Ponta. Formal specification and automatic analysis of business processes under authorization constraints: an action-based approach. In *Proceedings of the 6th International Conference on Trust, Privacy and Security in Digital Business (TrustBus'09)*, 2009.

[Artikis *et al.*, 2009] Alexander Artikis, Marek Sergot, and Jeremy Pitt. Specifying norm-governed computational societies. *ACM Transactions on Computational Logic*, 9(1), 2009.

[Bartholomew and Lee, 2012] Michael Bartholomew and Joohyung Lee. Stable models of formulas with intensional functions. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2012.

[Caldiran *et al.*, 2009] Ozan Caldiran, Kadir Haspalamutgil, Abdullah Ok, Can Palaz, Esra Erdem, and Volkan Patoglu. Bridging the gap between high-level reasoning and low-level control. In *Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pages 242–354, 2009.

[Casolary and Lee, 2011] Michael Casolary and Joohyung Lee. Representing the language of the Causal Calculator in Answer Set Programming. In *Technical Communications of the 27th International Conference on Logic Programming (ICLP)*, pages 51–61, 2011.

[Clark, 1978] Keith Clark. Negation as failure. In Herve Gallaire and Jack Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.

[Ferraris *et al.*, 2011] Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz. Stable models and circumscription. *Artificial Intelligence*, 175:236–263, 2011.

[Ferraris *et al.*, 2012] Paolo Ferraris, Joohyung Lee, Yuliya Lierler, Vladimir Lifschitz, and Fangkai Yang. Representing first-order causal theories by logic programs. *Theory and Practice of Logic Programming*, 12(3):383–412, 2012.

[Ferraris, 2006] Paolo Ferraris. Causal theories as logic programs. In *Proceedings of Workshop on Logic Programming (WLP)*, pages 35–44, 2006.

[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.

[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, 2005] Joohyung Lee. *Automated Reasoning about Actions*[4]. PhD thesis, University of Texas at Austin, 2005.

[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)*, pages 1025–1030, 2007.

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

[Lloyd and Topor, 1984] John Lloyd and Rodney Topor. Making Prolog more expressive. *Journal of Logic Programming*, 1:225–240, 1984.

[4]http://peace.eas.asu.edu/joolee/papers/dissertation.pdf

[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*[5]. PhD thesis, University of Texas at Austin, 1997.

[McCarthy, 1986] John McCarthy. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 26(3):89–116, 1986.

[Reiter, 1991] Raymond Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In Vladimir Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 359–380. Academic Press, 1991.

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