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 $C$ [Giunchiglia and Lifschitz, 1998], $C+$ [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

$$at(x, y, t + 1) \leftarrow move(x, y, t),$$

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

$$at(x, y, t + 1) \leftarrow at(x, y, t) \land at(x, y, t + 1)$$

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. Explainable 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

$$loc(x, t + 1) = y \leftarrow move(x, y, t)$$
$$loc(x, t + 1) = y \leftarrow loc(x, t) = y \land loc(x, t + 1) = y.$$  

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 $sum(x, y, z)$: 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

$$loc(x_1, t) = loc(x_2, t)$$
instead of
\[ \exists y (at(x_1, y, t) \land at(x_2, y, t)). \]

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 \( e \), choose a new variable \( \upsilon c \) similar to \( c \), and let \( \upsilon c \) stand for the list of all these variables. By \( T^\dagger (\upsilon c) \) we denote the conjunction of the formulas

\[ \forall x (G \rightarrow F^c_{\upsilon c}) \]  

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

\[ \forall \upsilon c (T^\dagger (\upsilon c) \leftrightarrow (\upsilon c = c)). \]  

(By \( \upsilon c = e \) we denote the conjunction of the formulas \( \upsilon c = c \) for all members \( c \) of the tuple \( e \).) 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 \( e \). Similarly, propositional symbols are viewed as predicate constants of arity 0. Equality, on the other hand, may not be included in \( e \).

\(^2\)That is to say, if \( e \) is a function constant then \( \upsilon e \) should be a function variable of the same arity; if \( e \) is a predicate constant then \( \upsilon e \) should be a predicate variable of the same arity.
2.2 Examples

Causal theory $T_0$ has two rules:

\[
\begin{align*}
p(x) & \iff q(x), \\
\neg p(x) & \iff \neg q(x),
\end{align*}
\]

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

\[
\forall \upsilon p (\forall x (q(x) \rightarrow \upsilon p(x)) \land \forall x (\neg p(x) \rightarrow \neg \upsilon p(x)) \iff \upsilon p = p),
\]

where $\upsilon p$ is a predicate variable. This formula is logically equivalent to

\[
\forall x (p(x) \iff q(x)).
\]

Causal theory $T_1$ has the rules

\[
\begin{align*}
\bot & \iff a = b, \\
c & = a & \iff c = a, \\
c & = b & \iff q,
\end{align*}
\]

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

\[
\forall \upsilon c ((a = b \rightarrow \bot) \land (c = a \rightarrow \upsilon c = a) \land (q \rightarrow \upsilon c = b) \iff \upsilon c = c)
\]

where $\upsilon c$ is an object variable. This formula is equivalent to

\[
a \neq b \land (q \rightarrow c = b) \land (\neg q \rightarrow c = a).
\]

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 $\text{none}$, which is used as the value of $\text{loc}(x, t)$ 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

\[
\begin{align*}
\perp & \leftarrow 0 = 1, \\
\perp & \leftarrow 0 = \text{none}, \\
\perp & \leftarrow 1 = \text{none}, \\
\text{obj}(x) \land \text{place}(y) & \leftarrow \text{move}(x, y), \\
\text{loc}(x, 0) & = y \leftarrow \text{loc}(x, 0) = y \land \text{obj}(x) \land \text{place}(y), \\
\text{loc}(x, 1) & = y \leftarrow \text{move}(x, y), \\
\text{loc}(x, 1) & = y \leftarrow \text{loc}(x, 0) = y \land \text{loc}(x, 1) = y \land \text{obj}(x) \land \text{place}(y), \\
\text{loc}(x, t) & = \text{none} \leftarrow \neg \text{obj}(x), \\
\text{loc}(x, t) & = \text{none} \leftarrow t \neq 0 \land t \neq 1,
\end{align*}
\]

and the function constant \( \text{loc} \) is explainable. The rule with \( \text{loc}(x, 0) \) in the head allows an object \( x \) to be initially anywhere: whichever place is the value of \( \text{loc}(x, 0) \), 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

\[
\forall v \text{loc}(T_2^\dagger(v \text{loc}) \leftrightarrow (v \text{loc} = \text{loc})),
\]

where \( v \text{loc} \) 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 \textit{Clark normal form} if it consists of

- rules of the form

\[
p(x) \leftarrow G(x),
\]

one for each explanable predicate symbol \( p \), where \( x \) is a tuple of distinct variables, and \( G(x) \) is a formula without any free variables other than the members of \( x \),

- rules of the form

\[
\neg p(x) \leftarrow G(x),
\]

\]
one for each explainable 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

\[
\forall \mathbf{x}(p(\mathbf{x}) \leftrightarrow G(\mathbf{x}))
\]

(11)

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

\[
\forall \mathbf{x}(\neg p(\mathbf{x}) \leftrightarrow G(\mathbf{x}))
\]

(12)

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

\[
\tilde{\forall}(G \rightarrow F)
\]

(13)

(the symbol \( \tilde{\forall} \) 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 explainable predicate symbol \( p \),
- rules of the form (10), one for each explainable predicate symbol \( p \),
- rules of the form

\[
f(\mathbf{x}) = y \Leftarrow G(\mathbf{x}, y),
\]

(14)

one for each explainable 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
type 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
\[
\begin{align*}
c &= x \iff x = a \land c = a, \\
c &= x \iff x = b \land q
\end{align*}
\]
and then merging them into one rule:
\[
c = x \iff (x = a \land c = a) \lor (x = b \land q). \tag{15}
\]
It is clear that the part of $T_1(\nu C)$ 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
\[
\begin{align*}
\bot &\iff 0 = 1, \\
\bot &\iff 0 = \text{none}, \\
\bot &\iff 1 = \text{none}, \\
\text{obj}(x) \land \text{place}(y) &\iff \text{move}(x, y), \\
\text{loc}(x, t) = y &\iff (t = 0 \land \text{loc}(x, 0) = y \land \text{obj}(x) \land \text{place}(y)) \\
&\lor (t = 1 \land \text{move}(x, y)) \\
&\lor (y = \text{none} \land \text{loc}(x, 0) = y \land \text{obj}(x) \land \text{place}(y)) \\
&\lor (y = \text{none} \land t \neq 0 \land t \neq 1). \tag{16}
\end{align*}
\]

\section{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 \textit{functional completion}
of a causal theory $T$ in Clark normal form is the conjunction of
\begin{itemize}
\item sentences (11) for all rules of $T$ of the form (9),
\item sentences (12) for all rules of $T$ of the form (10),
\item sentences
\[
\forall (f(x) = y \iff G(x, y)) \tag{17}
\]
for all rules of $T$ of the form (14), and
\item sentences (13) for all rules $F \iff G$ of $T$ without explainable symbols in the heads.
\end{itemize}
We will denote the functional completion of $T$ by $\text{FC}[T]$.

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

$$\exists x_1x_2(x_1 \neq x_2)$$

entails $T \iff \text{FC}[T]$.

**Corollary** If a causal theory in Clark normal form contains a rule of the form $\perp \leftarrow t_1 = t_2$ 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 $\perp \leftarrow a = b$.

Its functional completion is the conjunction of the formulas

$$\forall x(c = x \leftrightarrow (x = a \land c = a) \lor (x = b \land q))$$

and $a = b \rightarrow \perp$ (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

$$0 \neq 1, \ 0 \neq \text{none}, \ 1 \neq \text{none},$$
$$\forall xy(\text{move}(x, y) \rightarrow \text{obj}(x) \land \text{place}(y)),$$
$$\forall xty(\text{loc}(x, t) = y \leftrightarrow (t = 0 \land \text{loc}(x, 0) = y \land \text{obj}(x) \land \text{place}(y))$$
$$\lor (t = 1 \land \text{move}(x, y)$$
$$\lor (t = 1 \land \text{loc}(x, 0) = y \land \text{loc}(x, 1) = y \land \text{obj}(x) \land \text{place}(y))$$
$$\lor (y = \text{none} \land \neg \text{obj}(x))$$
$$\lor (y = \text{none} \land t \neq 0 \land t \neq 1)),$$

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

$$0 \neq 1, \ 0 \neq \text{none}, \ 1 \neq \text{none},$$
$$\forall xy(\text{move}(x, y) \rightarrow \text{obj}(x) \land \text{place}(y)),$$
$$\forall x(\text{obj}(x) \rightarrow \text{place}(\text{loc}(x, 0))),$$
$$\forall xt((\neg \text{obj}(x) \lor (t \neq 0 \land t \neq 1)) \rightarrow \text{loc}(x, t) = \text{none}),$$
$$\forall xy(\text{obj}(x) \rightarrow$$
$$\text{loc}(x, 1) = y \leftrightarrow (\text{move}(x, y) \lor (\text{loc}(x, 0) = y \land \neg \exists w \text{move}(x, w))))).$$

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 \( \forall ve(vc = c) \); 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
\[
\forall uf(\forall xy(G(x, y) \rightarrow uf(x) = y) \leftrightarrow uf = f)
\]
and
\[
\forall xy(f(x) = y \leftrightarrow G(x, y)).
\]
Right-to-left: under assumption (20), formula (19) is equivalent to the logically valid formula
\[
\forall uf(\forall xy(f(x) = y \rightarrow uf(x) = y) \leftrightarrow uf = f).
\]
Left-to-right: assume (19), that is,
\[
\forall uf(\forall xy(G(x, y) \rightarrow uf(x) = y) \rightarrow uf = f)
\]
and
\[
\forall xy(G(x, y) \rightarrow f(x) = y).
\]
The last formula is one half of equivalence (20). It remains to derive the other half, that is, \( G(x, f(x)) \). Assume that for some \( x^0 \), \( \neg G(x^0, f(x^0)) \). By (18), there exists a \( y_0 \) different from \( f(x^0) \). We will prove that the function \( uf \) defined by the condition
\[
uf(x^0) = y_0 \land \forall x(x \neq x^0 \rightarrow uf(x) = f(x))
\]
satisfies the antecedent of (21). Assume \( G(x, y) \). Since \( \neg G(x^0, y) \), \( x \neq x_0 \). Then \( uf(x) = f(x) \). On the other hand, by (22), \( f(x) = y \). Consequently \( uf(x) = y \); the antecedent of (21) is proved. It follows that the consequent \( uf = f \) holds, so that \( y_0 = uf(x^0) = f(x^0) \). 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 $c_1, c_2$ of explainable symbols we say that they are disjoint if

- $c_1$ is disjoint from $c_2$, and
- the symbols in $c_1$ do not occur in the heads of the rules of $T_2$, and the symbols in $c_2$ 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 $T_1 \land \ldots \land T_m$.\*

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 $T_1 \land \ldots \land T_m \land T_{m+1}$. According to the special case proved in Section 5.1, (18) entails

$$T_i \leftrightarrow FC[T_i] \quad (i = 1, \ldots, m).$$

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

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

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

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 Lif-
schitz, and Hudson Turner. Representing the Zoo World and the Traffic World
in the language of the Causal Calculator. Artificial Intelligence, 153(1–2):105–

[Armando et al., 2009] Alessandro Armando, Enrico Giunchiglia, and Ser-
ena Elisa Ponta. Formal specification and automatic analysis of business pro-
cesses under authorization constraints: an action-based approach. In Proce-
edings of the 6th International Conference on Trust, Privacy and Security in Dig-
tal Business (TrustBus’09), 2009.

[Artikis et al., 2009] Alexander Artikis, Marek Sergot, and Jeremy Pitt. Specify-
ing norm-governed computational societies. ACM Transactions on Compu-
tational Logic, 9(1), 2009.

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.


---


