1

Eliminating Function Symbols from a Nonmonotonic Causal Theory

VLADIMIR LIFSCHITZ AND FANGKAI YANG

ABSTRACT. Nonmonotonic causal logic is a knowledge representation language designed for describing domains that involve actions and change. Problems related to action domains described in this language can be often solved using computational methods of answer set programming. This idea has led researchers at the University of Potsdam to the development of COALA, a compiler from action languages to answer set programming. Our goal is to address a significant limitation of the theory that COALA is based on: it is not directly applicable to fluents with non-Boolean values, represented by function symbols, such as the location of an object. We show that a function symbol in a causal theory can be often eliminated in favor of a new predicate symbol in such a way that the resulting theory can be translated into an executable logic program.

1 Introduction

Nonmonotonic causal logic is a knowledge representation language designed for describing domains that involve actions and change. Its original version [McCain and Turner 1997] was propositional. The first-order formulation defined in [Lifschitz 1997] provides additional expressive power that is essential for describing the semantics of action descriptions with variables [Lifschitz and Ren 2007]. Nonmonotonic causal logic was used for defining the semantics of action description languages $\mathcal{C}$ [Giunchiglia and Lifschitz 1998], $\mathcal{C}+$ [Giunchiglia, Lee, Lifschitz, McCain, and Turner 2004], and MAD [Lifschitz and Ren 2006].

A causal theory consists of causal rules $F \Leftarrow G$, where F and G are first-order formulas. The rule reads "F is caused if G is true." For instance, the causal rule

$$\mathit{loc}(x, t+1) = y \Leftarrow \mathit{move}(x, y, t)$$

expresses that there is a cause for the object x to be located at y at time $t+1$ if x is moved to y at time t. (Executing the move action is the cause.) The distinction between being true and having a cause is used in [McCain and Turner 1997] as a basis for an elegant solution to the frame problem. For instance, the commonsense law of inertia for locations can be expressed by the causal law

$$\mathit{loc}(x, t+1) = y \Leftarrow \mathit{loc}(x, t) = y \wedge \mathit{loc}(x, t+1) = y$$

(if the location of x at time $t+1$ is the same as at time t then there is a cause for this; inertia is the cause).

Results of [McCain 1997; Lifschitz and Yang 2010; Ferraris, Lee, Lierler, Lifschitz, and Yang 2011] show that causal logic is closely related to logic programming under the stable model semantics [Gelfond and Lifschitz 1988; Gelfond and Lifschitz 1991; Ferraris, Lee, and Lifschitz 2007; Ferraris, Lee, and Lifschitz 2011]. For this reason, computational methods of answer set programming (ASP) [Marek and Truszczyński 1999; Niemelä 1999; Lifschitz 2002; Lifschitz 2008] can be used for answering questions about action domains described in causal logic and in related action languages[Doğandağ, Alpaslan, and Akman 2001; Doğandağ, Ferraris, and Lifschitz 2004]. This idea has led to the development of COALA, a compiler from action languages to ASP [Gebser, Grote, and Schaub 2010].

Our goal here is to address a significant limitation of the theory that COALA is based on: it is restricted to Boolean-valued fluents, represented in causal logic by predicate symbols. It is not directly applicable to fluents with non-Boolean values, represented by function symbols, such as the location of an object in the example above. There is a good reason for this limitation. Describing non-Boolean fluents by logic programs involves an additional difficulty in view of the fact that rules in a logic program containing function symbols cannot be used to characterize values of functions; they can only characterize extents of predicates.[1]

Our approach is to show how a function symbol in a causal theory can be eliminated in favor of a new predicate symbol. In classical logic, this process is well understood. For instance, addition in first-order arithmetic can be described using a ternary predicate symbol, instead of a binary function symbol: we can write $\mathit{sum}(x, y, z)$ instead of $x + y = z$. (This alternative leads to the use of cumbersome formulas, of course, when complex algebraic expressions are involved.) Extending this process to nonmonotonic causal logic is not straightforward, especially if we want to arrive eventually at an executable ASP program.

This paper touches on several topics within logic-based artificial intelligence: representing properties of actions, nonmonotonic reasoning, and declarative programming. We are very pleased to have it published in a collection honoring Hector Levesque, who has contributed so much to all these areas.

The paper is organized as follows. After reviewing the syntax and semantics of first-order causal logic, we will describe two procedures for eliminating function constants from a causal theory in favor of predicate constants, "general" and "definite." Then we will show how definite elimination can help us turn a causal theory into executable ASP code, and how it can be extended to rules that express the synonymity of function symbols.

2 Review of Causal Logic

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

- a list $\mathbf{c}$ of distinct function and/or predicate constants,[2] 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 for-

[1]The language of [Lin and Wang 2008] is not an exception: it does not permit formulas like $\mathit{loc}(x, t+1) = y$ in heads of rules.

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

mula. For each member c of **c**, choose a new variable υc similar to c,[3] 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

$$\forall\mathbf{x}(G \rightarrow F^{\mathbf{c}}_{\upsilon\mathbf{c}}) \tag{1}$$

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

$$\forall\upsilon\mathbf{c}(T^{\dagger}(\upsilon\mathbf{c}) \leftrightarrow (\upsilon\mathbf{c} = \mathbf{c})). \tag{2}$$

(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 (2) 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.

EXAMPLE 1. Let T_1 be the causal theory

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

where the object constant c is the only explainable symbol (so that the object constants a and b and the propositional symbol q are not explainable).[4] The first rule of T_1 says that a and b are different from each other. 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. According to the semantics of causal logic, 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. (There are no second-order variables in this formula. More generally, (1) does not involve second-order variables whenever all explainable symbols are object constants.) This sentence is equivalent to the quantifier-free formula

$$a \neq b \wedge (q \rightarrow c = b) \wedge (\neg q \rightarrow c = a). \tag{3}$$

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

Causal rules with the head $\bot$, such as the first rule of T_1, are called *constraints*. The causal theory obtained from a causal theory T by adding a constraint $\bot \Leftarrow F$ is equivalent to $T \wedge \widetilde{\forall}\neg F$. The effect of adding the rule $\neg F \Leftarrow \top$ to T is usually different, unless F does not contain explainable symbols. Since the first rule of T_1 does not contain explainable symbols, it can be equivalently replaced by the rule $a \neq b \Leftarrow \top$.

EXAMPLE 2. We would like to describe the effect of moving an object. For simplicity, we only consider the time instants 0, 1 and the execution of the move action at time 0.

[3]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.

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

The formulation below includes the auxiliary symbol *none*, which is used as the value of $loc(x,t)$ when the arguments are "not of the right kind" (that is, when x is not a physical object or when t is not a time instant). The rules of the causal theory T_2 are

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

where the function constant *loc* is the only explainable symbol. The rule with $loc(x,0)$ in the head says that initially an object x can be anywhere: whatever value $loc(x,0)$ has, there is a cause for that. The next two rules describe the effect of moving objects and the inertia property of locations. Causal theory T_2 is shorthand for the formula

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

where υloc is a binary function variable.

3 Plain Causal Theories

Let f be a function constant. An atomic formula is *f-plain* if

- it does not contain f, or
- has the form $f(\mathbf{t}) = u$, where $\mathbf{t}$ is a tuple of terms not containing f, and u is a term not containing f.

A first-order formula, a causal rule, or a causal theory is *f-plain* if all its atomic subformulas are f-plain. For instance, the causal theory from Example 1 is c-plain, and the causal theory from Example 2 is *loc*-plain.

It is easy to transform any first-order formula into an equivalent f-plain formula. For instance, $p(f(f(x))$ is equivalent to the f-plain formula

$$\exists yz(f(x) = y \wedge f(y) = z \wedge p(z)).$$

Any causal theory can be transformed into an equivalent f-plain causal theory by applying this transformation to the heads and bodies of all rules. In Sections 4–7 we will assume that the given causal theory is f-plain.

4 General Elimination

Let T be an f-plain causal theory, where f is an explainable function constant. The causal theory T' is obtained from T as follows:

(1) in the signature of T, replace f with a new explainable predicate constant p of arity $n+1$, where n is the arity of f;

(2) in the rules of T, replace each subformula $f(\mathbf{t}) = u$ with $p(\mathbf{t}, u)$;

(3) add the rules

$$(\exists y)p(\mathbf{x}, y) \Leftarrow \top \tag{4}$$

and

$$\neg p(\mathbf{x}, y) \vee \neg p(\mathbf{x}, z) \Leftarrow y \neq z, \tag{5}$$

where **x** is a tuple of variables, and the variables **x**, y, z are pairwise distinct.

Rule (5) expresses, in the language of causal logic, the uniqueness of y such that $p(\mathbf{x}, y)$.

THEOREM 1. *The sentence*

$$\forall \mathbf{x} y(p(\mathbf{x}, y) \leftrightarrow f(\mathbf{x}) = y) \tag{6}$$

entails $T \leftrightarrow T'$.

EXAMPLE 1, CONTINUED. The result T_1' of applying this transformation to T_1 and to the object constant c as f is the causal theory

$$\begin{aligned}
\bot &\Leftarrow a = b,\\
p(a) &\Leftarrow p(a),\\
p(b) &\Leftarrow q,\\
(\exists y)p(y) &\Leftarrow \top,\\
\neg p(y) \vee \neg p(z) &\Leftarrow y \neq z
\end{aligned}$$

with the explainable symbol p. According to Theorem 1, the equivalence between T_1 and T_1' is entailed by the sentence

$$\forall y(p(y) \leftrightarrow c = y). \tag{7}$$

EXAMPLE 2, CONTINUED. The result T_2' of applying this transformation to T_2 and to the function constant *loc* as f is the causal theory

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

with the explainable symbol *at*. According to Theorem 1, the equivalence between T_2 and T_2' is entailed by the sentence

$$\forall xty(\mathit{at}(x, t, y) \leftrightarrow \mathit{loc}(x, t) = y) \tag{8}$$

By repeated applications of this process we can eliminate all explainable function symbols, provided that T is f-plain for each explainable symbol f.

The following corollary shows that there is a simple 1–1 correspondence between models of T and models of T'. Recall that the signature of T' is obtained from the signature of T by replacing f with p. For any interpretation I of the signature of T, by I^f_p we denote the interpretation of the signature of T' obtained from I by replacing the function f^I with the set p^I that consists of the tuples

$$\langle \xi_1, \dots, \xi_n, f^I(\xi_1, \dots, \xi_n) \rangle$$

for all $\xi_1, \dots, \xi_n$ from the universe of I.

COROLLARY 2. *(a) An interpretation I of the signature of T is a model of T iff I^f_p is a model of T'. (b) An interpretation J of the signature of T' is a model of T' iff $J = I^f_p$ for some model I of T.*

Part (a) follows from the fact that the "union" of I and I^f_p satisfies (6). To show that any model of T' can be represented in the form I^f_p for some interpretation I, note that T' contains rules (4), (5) and consequently entails

$$\forall \mathbf{x}(\exists! y) p(\mathbf{x}, y). \tag{9}$$

5 Definite Elimination

Unfortunately, the elimination process described in the previous section does not help us turn a causal theory with explainable function symbols into a logic program. The problem is that the translation from causal logic into logic programming described in [Ferraris, Lee, Lierler, Lifschitz, and Yang 2011, Section 5] does not apply to causal rules with an existential quantifier in the head, such as (4). We will now describe an alternative elimination process, which is limited to causal theories of a special form but does not add rules containing quantifiers.

Consider an f-plain causal theory T, where f is an explainable function constant f satisfying the following condition: the head of any rule of T either does not contain f or has the form $f(\mathbf{t}) = u$, where **t** is a tuple of terms not containing explainable symbols, and u is a term not containing explainable symbols. The causal theory T'' is obtained from T as follows:

(1) in the signature of T, replace f with a new explainable predicate constant p of arity $n + 1$, where n is the arity of f;

(2) in the rules of T, replace each subformula $f(\mathbf{t}) = u$ with $p(\mathbf{t}, u)$;

(3$'$) add the rule

$$\neg p(\mathbf{x}, y) \Leftarrow \neg p(\mathbf{x}, y), \tag{10}$$

where **x** is a tuple of variables, and the variables **x**, y are pairwise distinct.

Rule (10) expresses the "closed world assumption" for p: by default, $p(\mathbf{x}, y)$ is false.[5]

We call this process "definite elimination" because all new causal rules that it introduces are definite in the sense of [Lifschitz 1997, Section 5]. Definite elimination is applicable to the causal theories from Examples 1 and 2, but it cannot be applied, for instance, to a theory containing a rule with the head of the form $f(x) = y \vee f(x) = z$ or $\neg(f(x) = y)$.

[5]It is somewhat similar to the second rule of Example 1 (Section 2).

THEOREM 3. *The sentences (6) and*

$$\exists xy(x \neq y) \tag{11}$$

entail $T \leftrightarrow T''$.

Formula (11) expresses that the universe contains at least two elements. Without this assumption, the statement of Theorem 3 would be incorrect. Indeed, consider the causal theory T_3 with an explainable function symbol f that consists of the single rule $\top \Leftarrow \top$. It is easy to check that T_3 is equivalent to $\forall xy(x = y)$. On the other hand, T''_3 consists of the rules

$$\begin{aligned} \top &\Leftarrow \top, \\ \neg p(\mathbf{x}, y) &\Leftarrow \neg p(\mathbf{x}, y), \end{aligned}$$

and is equivalent to $\forall \mathbf{x} y \neg p(\mathbf{x}, y)$. The interpretation with a singleton universe that makes p identically true satisfies (6) and is a model of T_3, but it is not a model of T''_3.

COROLLARY 4. *If T contains a constraint of the form $\bot \Leftarrow t_1 = t_2$, where t_1, t_2 don't contain explainable function symbols, then (6) entails $T \leftrightarrow T''$.*

Indeed, if T contains the constraint $\bot \Leftarrow a = b$ then T'' contains it also, so that (11) is entailed both by T and by T''.

EXAMPLE 1, CONTINUED. The result T''_1 of applying definite elimination to T_1 and to the object symbol c is the theory

$$\begin{aligned} \bot &\Leftarrow a = b, \\ p(a) &\Leftarrow p(a), \\ p(b) &\Leftarrow q, \\ \neg p(y) &\Leftarrow \neg p(y) \end{aligned} \tag{12}$$

with the explainable symbol p. By Corollary 4, the equivalence between T_1 and T''_1 is entailed by sentence (7). Using the completion theorem from [Lifschitz 1997, Section 5], it is easy to check that causal theory (12) is equivalent to the first-order sentence

$$a \neq b \wedge (q \leftrightarrow p(b)) \wedge \forall y(p(y) \leftrightarrow (y = a \vee y = b)).$$

Under assumption (9), which can be written in this case as

$$(\exists! y) p(y), \tag{13}$$

this sentence can be equivalently transformed into a formula conveying the same information as (3):

$$a \neq b \wedge (q \rightarrow p(b)) \wedge (\neg q \rightarrow p(a)). \tag{14}$$

EXAMPLE 2, CONTINUED. The result T''_2 of applying definite elimination to theory T_2 and to function symbol *loc* is the theory

$$\begin{aligned} \bot &\Leftarrow 0 = 1, \\ \bot &\Leftarrow 0 = \mathit{none}, \\ \bot &\Leftarrow 1 = \mathit{none}, \\ \mathit{at}(x, 0, y) &\Leftarrow \mathit{at}(x, 0, y) \wedge \mathit{obj}(x) \wedge \mathit{place}(y), \\ \mathit{at}(x, 1, y) &\Leftarrow \mathit{move}(x, y) \wedge \mathit{obj}(x) \wedge \mathit{place}(y), \\ \mathit{at}(x, 1, y) &\Leftarrow \mathit{at}(x, 0, y) \wedge \mathit{at}(x, 1, y) \wedge \mathit{obj}(x) \wedge \mathit{place}(y), \\ \mathit{at}(x, t, \mathit{none}) &\Leftarrow \neg \mathit{obj}(x), \\ \mathit{at}(x, t, \mathit{none}) &\Leftarrow t \neq 0 \wedge t \neq 1, \\ \neg \mathit{at}(x, t, y) &\Leftarrow \neg \mathit{at}(x, t, y) \end{aligned} \tag{15}$$

with the explainable symbol *at*. By Corollary 4, the equivalence between T_2 and T_2'' is entailed by sentence (8). Using the completion theorem from [Lifschitz 1997] we can show that under assumption (9), which can be written in this case as

$$\forall xt(\exists! y)at(x,t,y), \tag{16}$$

theory (15) can be equivalently transformed into the conjunction of the universal closures of the formulas

$$\begin{array}{l}
\neg(0=1),\ \neg(0=\mathit{none}),\ \neg(1=\mathit{none}),\\
\forall xy(\mathit{at}(x,0,y)\land\neg\mathit{obj}(x)\to y=\mathit{none}),\\
\forall xy(\mathit{at}(x,0,y)\land\mathit{obj}(x)\to\mathit{place}(y)),\\
\forall xy(\mathit{at}(x,1,y)\leftrightarrow((\mathit{move}(x,y)\land\mathit{obj}(x)\land\mathit{place}(y))\\
\qquad\qquad\qquad\lor(\mathit{at}(x,0,y)\land\mathit{obj}(x)\land\mathit{place}(y)\land\neg\exists w(\mathit{move}(x,w)\land\mathit{place}(w)))\\
\qquad\qquad\qquad\lor(y=\mathit{none}\land\neg\mathit{obj}(x)))),\\
\forall xyt((t\neq 0\land t\neq 1)\to(\mathit{at}(x,t,y)\leftrightarrow y=\mathit{none})).
\end{array}$$

The equivalence with the left-hand side $at(x,1,y)$ is similar to successor state axioms in the sense of [Reiter 1991].

By repeated applications of this process we can eliminate all explainable function symbols provided that

- T is f-plain for each explainable function symbol f, and
- the head of each rule of T containing an explainable function symbol f has the form $f(\mathbf{t})=u$, where $\mathbf{t}$ is a tuple of terms not containing explainable symbols, and u is a term not containing explainable symbols.

We saw in Section 4 that the mapping $I\mapsto I^f_p$ is a 1–1 correspondence between the class of models of T and the class of models of T'. For definite elimination, this mapping establishes a 1–1 correspondence between the models of T with the universe of cardinality >1 and the models of T'' with the universe of cardinality >1 that satisfy additional condition (9); that condition, generally, is not entailed by T''. By $T^{\exists!}$ we denote the causal theory obtained from T'' by adding the constraint

$$\bot \Leftarrow (\exists! y)p(\mathbf{x},y). \tag{17}$$

It is clear that $T^{\exists!}$ is equivalent to the conjunction of T'' with (9).

COROLLARY 5. *(a) An interpretation I of the signature of T with the universe of cardinality >1 is a model of T iff I^f_p is a model of $T^{\exists!}$. (b) An interpretation J of the signature of $T^{\exists!}$ with the universe of cardinality >1 is a model of $T^{\exists!}$ iff $J=I^f_p$ for some model I of T.*

For instance, the mapping $I\mapsto I^c_p$ establishes a 1–1 correspondence between the models of T_1 and the models of $T_1^{\exists!}$. Similarly, the mapping $I\mapsto I^{loc}_{at}$ establishes a 1–1 correspondence between the models of T_2 and the models of $T_2^{\exists!}$.

As discussed at the beginning of this section, the definite elimination process is limited to causal rules satisfying an additional syntactic restriction: if the head of a rule of T

contains f then it should be an atomic formula. Without this restriction, the assertion of Theorem 3 would be incorrect. Consider, for instance, the causal theory T_4 with the rules

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

and explainable c. It is easy to check that T_4 is inconsistent. On the other hand, T_4'' is

$$\begin{aligned} \bot &\Leftarrow a = b, \\ p(a) \lor p(b) &\Leftarrow \top, \\ \neg p(x) &\Leftarrow \neg p(x). \end{aligned}$$

This causal theory is equivalent to

$$a \neq b \land (\forall x(p(x) \leftrightarrow x = a) \lor \forall x(p(x) \leftrightarrow x = b)).$$

The interpretation with the universe $\{a, b\}$ that interprets c as a and p as $\{a\}$ satisfies (6), (11), and T_4'', but does not satisfy T_4.

Another restriction imposed at the beginning of this section is that in formulas $f(\mathbf{t}) = u$ in the heads of rules, **t** and u don't contain explainable symbols. Without this restriction, the assertion of Theorem 3 would be incorrect. Let T_5 be the causal theory obtained from T_4 by adding the rule

$$d = c \Leftarrow \top,$$

with both c and d explainable. It is easy to check that T_5 is inconsistent. Consider the result T_5'' of applying definite elimination to T_5 and d:

$$\begin{aligned} \bot &\Leftarrow a = b, \\ c = a \lor c = b &\Leftarrow \top, \\ p(c) &\Leftarrow \top, \\ \neg p(x) &\Leftarrow \neg p(x), \end{aligned}$$

with p and c explainable. This causal theory is equivalent to

$$(a \neq b) \land (\forall x(p(x) \leftrightarrow x = a) \lor \forall x(p(x) \leftrightarrow x = b)) \land p(c).$$

The interpretation with the universe $\{a, b\}$ that interprets c as a, d as a, and p as $\{a\}$ satisfies (6), (11), and T_5'', but does not satisfy T_5.

6 Modified Definite Elimination

As discussed in Section 5, rule (10) expresses the closed world assumption for p. In "modified definite elimination," (10) is replaced by a definite counterpart of the uniqueness rule (5):

$$\neg p(\mathbf{x}, y) \Leftarrow p(\mathbf{x}, z) \land y \neq z$$

(**x** is a tuple of variables, and the variables **x**, y, z are pairwise distinct). We will denote the result of applying the modified definite elimination process to T by T'''. For instance, T_1''' is

$$\begin{aligned} \bot &\Leftarrow a = b, \\ p(a) &\Leftarrow p(a), \\ p(b) &\Leftarrow q, \\ \neg p(y) &\Leftarrow p(z) \land y \neq z. \end{aligned}$$

Causal theories T'' and T''' are essentially equivalent to each other. To be precise, formula (6) entails $T'' \leftrightarrow T'''$. Indeed, $(T''')^\dagger$ can be obtained from $(T'')^\dagger$ by replacing

$$\forall \mathbf{x} y(\neg p(\mathbf{x}, y) \rightarrow \neg \upsilon p(\mathbf{x}, y)) \tag{18}$$

with

$$\forall \mathbf{x} yz(p(\mathbf{x}, z) \wedge y \neq z \rightarrow \neg \upsilon p(\mathbf{x}, y)). \tag{19}$$

Formula (19) can be rewritten as

$$\forall \mathbf{x} y(\exists z(p(\mathbf{x}, z) \wedge y \neq z) \rightarrow \neg \upsilon p(\mathbf{x}, y)).$$

In the presence of (6), the antecedent of this formula is equivalent to the antecedent of (18). Consequently $(T'')^\dagger \leftrightarrow (T''')^\dagger$, so that $T'' \leftrightarrow T'''$.

This fact implies that the assertions of Theorem 3 and Corollary 4 will remain valid if we replace T'' in their statements with T'''.

7 From Causal Logic to ASP

7.1 Turning Causal Theories into Logic Programs

The version of the stable model semantics that we will refer to in this section is defined in [Ferraris, Lee, and Lifschitz 2011]. That paper defines, for any first-order sentence F and any tuple $\mathbf{p}$ of predicate constants, which models of F are $\mathbf{p}$-*stable*. The predicate symbols from $\mathbf{p}$ are called intensional, and the other predicate symbols are extensional. (Intensional predicates are somewhat similar to minimized predicates in circumscription and to explainable symbols in causal logic.)

The translation defined in [Ferraris, Lee, Lierler, Lifschitz, and Yang 2011, Section 5] transforms a causal theory T satisfying some syntactic conditions into a first-order sentence F that has "the same meaning under the stable model semantics" as the theory T. (One of the conditions on T is that its explainable symbols are predicate constants, not function constants.) To be more precise, if the explainable symbols of T, along with the auxiliary predicate symbols introduced by the translation, are taken to be intensional then the stable models of F are identical to the models of T, provided that the interpretations of the auxiliary predicate symbols are "forgotten." In many cases, this translation can be applied to theories obtained by the definite elimination process described above.

EXAMPLE 1, CONTINUED. Consider the causal theory $T_1^{\exists!}$, which, as we have seen, is "isomorphic" to T_1. It consists of the rules

$$\begin{aligned}
\bot &\Leftarrow a = b,\\
p(a) &\Leftarrow p(a),\\
p(b) &\Leftarrow q,\\
\neg p(y) &\Leftarrow \neg p(y),\\
\bot &\Leftarrow \neg(\exists! y)p(y).
\end{aligned}$$

The result of applying the translation from [Ferraris, Lee, Lierler, Lifschitz, and Yang 2011, Section 5.2] to this theory is the conjunction of the universal closures of the formulas

$$\begin{array}{r l}
a \neq b, & \\
\neg\neg p(a) & \rightarrow p(a), \\
\neg\neg q & \rightarrow p(b), \\
\neg\neg\neg p(x) & \rightarrow \widehat{p}(x), \\
\neg\neg(\exists! y)p(y), & \\
\neg(p(x) \wedge \widehat{p}(x)), & \\
\neg(\neg p(x) \wedge \neg\widehat{p}(x)), &
\end{array} \tag{20}$$

where $\widehat{p}$ is an auxiliary predicate.[6] Theorem 5 from [Ferraris, Lee, and Lifschitz 2011] shows that these formulas can be rewritten as

$$\begin{array}{r l}
a \neq b, & \\
\neg\widehat{p}(a) & \rightarrow p(a), \\
q & \rightarrow p(b), \\
\neg p(x) & \rightarrow \widehat{p}(x), \\
\neg\neg(\exists! y)p(y), & \\
\neg(p(x) \wedge \widehat{p}(x)), & \\
\neg(\neg p(x) \wedge \neg\widehat{p}(x)), &
\end{array} \tag{21}$$

without changing the class of stable models. Thus the stable models of (the conjunction of the universal closures of) formulas (21) turn into the models of $T_1^{\exists!}$ as soon as the interpretation of $\widehat{p}$ is dropped. It follows that the class of stable models of (21) is "isomorphic" to the class of models of T_1.

EXAMPLE 2, CONTINUED. The result of applying the translation from [Lifschitz and Yang 2010] to $T_2^{\exists!}$ becomes, and simplifications,

$$\begin{array}{r l}
0 \neq 1,\ 0 \neq \mathit{none},\ 1 \neq \mathit{none}, & \\
\neg\widehat{at}(x,0,y) \wedge \mathit{obj}(x) \wedge \mathit{place}(y) & \rightarrow \mathit{at}(x,0,y), \\
\mathit{move}(x,y) \wedge \mathit{obj}(x) \wedge \mathit{place}(y) & \rightarrow \mathit{at}(x,1,y), \\
\neg\widehat{at}(x,0,y) \wedge \neg\widehat{at}(x,1,y) \wedge \mathit{obj}(x) \wedge \mathit{place}(y) & \rightarrow \mathit{at}(x,1,y), \\
\neg \mathit{at}(x,t,y) & \rightarrow \widehat{at}(x,t,y), \\
\neg \mathit{obj}(x) & \rightarrow \mathit{at}(x,t,\mathit{none}), \\
t \neq 0 \wedge t \neq 1 & \rightarrow \mathit{at}(x,t,\mathit{none}), \\
\neg\neg(\forall x t \exists! y)\mathit{at}(x,t,y), & \\
\neg(\mathit{at}(x,t,y) \wedge \widehat{at}(x,t,y)), & \\
\neg(\neg \mathit{at}(x,t,y) \wedge \neg\widehat{at}(x,t,y)). &
\end{array} \tag{22}$$

The class of stable models of (22) is "isomorphic" to the class of models of T_2.

7.2 Turning Causal Theories into Executable Code

In many cases, answer set solvers such as CLINGO[7] allow us to generate the Herbrand stable models of a given sentence. Consequently they can be sometimes used to generate models of causal theories.

[6]This predicate is analogous to the classical negation of p in the sense of [Gelfond and Lifschitz 1991]. The last of formulas (20) is a consistency and completeness condition.

[7]`http://potassco/sourceforge.net/`

EXAMPLE 1, CONTINUED. We would like to find all models of T_1 with the universe $\{a, b\}$ in which the constants a, b represent themselves. These models correspond to the Herbrand stable models of (21). We can find them by running CLINGO on the following input:

```
u(a;b).  #domain u(X).
{q}.
p(a) :- not -p(a).
p(b) :- q.
-p(X) :- not p(X).
:- not 1{p(Z):u(Z)}1.
:- not p(X), not -p(X).
```

The first line expresses that the universe `u` consists of `a` and `b`, and that `X` is a variable for arbitrary elements of `u`. The choice rule in the second line says that *q* can be assigned an arbitrary value. The other lines correspond to five of the formulas (21), with the classical negation `-p` representing $\widehat{p}$. There is no need to include a constraint corresponding to $a \neq b$, because the unique name assumption is true in all Herbrand models and thus is taken by CLINGO for granted. A constraint corresponding to $\neg(p(x) \wedge \widehat{p}(x))$ would be redundant as well, since $\widehat{p}$ is represented by classical negation.

Given this input, CLINGO generates two stable models: one containing `q` and `p(b)`, the other containing `p(a)`. Consequently T_1 has two models of the kind that we are interested in: in one of them *q* is true and the value of c is *b*; in the other, *q* is false and the value of c is a.

EXAMPLE 2, CONTINUED. Consider the dynamic domain consisting of two objects o_1, o_2 that can be located in any of two places l_1, l_2. What are the possible locations of the objects after moving o_1 to l_2, for each possible initial state? To answer this question, we will find the models of T_2 with the universe

$$\{o_1, o_2, l_1, l_2, 0, 1, \mathit{none}\}$$

such that

- each of the constants 0, 1, *none* represents itself,
- the extent of *obj* is $\{o_1, o_2\}$,
- the extent of *place* is $\{l_1, l_2\}$, and
- the extent of *move* is $\{\langle o_1, l_2 \rangle\}$.

To this end, we will find the stable models of (22) that satisfy all these conditions.

This computational task is equivalent to finding the Herbrand models of the sentence obtained by conjoining the universal closures of formulas (22) with the formulas

$$\mathit{obj}(o_1),\ \mathit{obj}(o_2),\ \mathit{place}(l_1),\ \mathit{place}(l_2),\ \mathit{move}(o_1, l_2)$$

(o_1, o_2, l_1, l_2 are new object constants), with *obj*, *place* and *move* included in the list of intensional predicates along with *at* and $\widehat{at}$.[8] To find these models, we run CLINGO on the following input:

[8]This claim can be justified using the splitting theorem. See [Ferraris, Lee, Lifschitz, and Palla 2009, Sections 6, 7].

```
u(o1;o2;l1;l2;0;1;none).
#domain u(X).  #domain u(T).  #domain u(Y).
at(X,0,Y) :- not -at(X,0,Y), obj(X), place(Y).
at(X,1,Y) :- move(X,Y), obj(X), place(Y).
at(X,1,Y) :- not -at(X,0,Y), not -at(X,1,Y), obj(X), place(Y).
-at(X,T,Y) :- not at(X,T,Y).
at(X,T,none) :- not obj(X).
at(X,T,none) :- T!=0, T!=1.
:- not 1{at(X,T,Z):u(Z)}1.
:- not at(X,T,Y), not -at(X,T,Y).
obj(o1;o2).  place(l1;l2).  move(o1,l2).
```

CLINGO generates 4 stable models, one for each possible combination of the locations of `o1` and `o2` at time 0. In every stable model, at time 1 object `o1` is at `l2`, and object `o2` is at the same place where it was at time 0.

8 Synonymity Rules

In this section we extend the definite elimination process (Section 5) to the case when several explainable function constants are eliminated in favor of predicate constants simultaneously, and the causal theory may contain rules of the form

$$f_1(\mathbf{t}^1) = f_2(\mathbf{t}^2) \Leftarrow G,$$

where f_1 and f_2 are two of the symbols that are eliminated. This rule expresses that there is a cause for $f_1(\mathbf{t}^1)$ to be “synonymous” to $f_2(\mathbf{t}^2)$ under condition G. Such “synonymity rules” play an important role in reasoning about actions [Erdoğan and Lifschitz 2006; Lifschitz and Ren 2006; Lee, Lierler, Lifschitz, and Yang 2010].

Consider a causal theory T and a tuple **f** of explainable function constants such that the bodies of the rules of T are f-plain for all members f of **f**, and the head of any rule of T

- does not contain members of **f**, or
- has the form $f(\mathbf{t}) = u$, where f is a member of **f**, **t** is a tuple of terms not containing explainable symbols, and u is a term not containing explainable symbols, or
- has the form $f_1(\mathbf{t}^1) = f_2(\mathbf{t}^2)$, where f_1, f_2 are members of **f**, and $\mathbf{t}^1$, $\mathbf{t}^2$ are tuples of terms not containing explainable symbols.

The causal theory T'' is obtained from T as follows:

(1) in the signature of T, replace each member f of **f** with a new explainable predicate constant p of arity $n + 1$, where n is the arity of f;

(2a) in the rules of T, replace each subformula $f(\mathbf{t}) = u$ such that f is a member of **f** and u doesn't contain members of **f**, with $p(\mathbf{t}, u)$;

(2b) in the heads of rules of T, replace each formula $f_1(\mathbf{t}^1) = f_2(\mathbf{t}^2)$ such that f_1, f_2 are members of **f**, with $p_1(\mathbf{t}^1, y) \leftrightarrow p_2(\mathbf{t}^2, y)$, where y is a new variable;

(3′) add, for every new predicate p, the rule

$$\neg p(\mathbf{x}, y) \Leftarrow \neg p(\mathbf{x}, y),$$

where **x** is a tuple of variables, and the variables **x**, y are pairwise distinct.

THEOREM 6. *Sentences (6) for all f from* **f** *and sentence (11) entail $T \leftrightarrow T''$.*

EXAMPLE 3. Consider the causal theory T_6

$$\begin{aligned} f(x) = y &\Leftarrow a(x,y), \\ g(x) = y &\Leftarrow b(x,y), \\ f(x) = g(x) &\Leftarrow c(x) \end{aligned}$$

with the explainable symbols f, g. Its translation T_6'' is

$$\begin{aligned} p(x,y) &\Leftarrow a(x,y), \\ q(x,y) &\Leftarrow b(x,y), \\ p(x,y) \leftrightarrow q(x,y) &\Leftarrow c(x), \\ \neg p(x,y) &\Leftarrow \neg p(x,y), \\ \neg q(x,y) &\Leftarrow \neg q(x,y) \end{aligned} \tag{23}$$

with the explainable symbols p, q.

Theorem 6 turns into Theorem 3 in the special case when **f** is a single function symbol and T does not contain synonymity rules.

For the class of causal theories studied in Section 5, the mapping $I \mapsto I_p^f$ establishes a 1–1 correspondence between the non-singleton models of T and the non-singleton models of $T^{\exists!}$ (Corollary 5). This assertion can be generalized to theories with synonymity rules as follows.

Recall that the signature of T'' is obtained from the signature of T by replacing the members of the tuple **f** of function constants with new predicate constants; we will denote the list of these predicate constants by **p**. For any interpretation I of the signature of T, by $I_{\mathbf{p}}^{\mathbf{f}}$ we denote the interpretation of the signature of T'' obtained from I by replacing, for each $f \in$ **f**, the function f^I with the set p^I that consists of the tuples

$$\langle \xi_1, \dots, \xi_n, f^I(\xi_1, \dots, \xi_n) \rangle$$

for all $\xi_1, \dots, \xi_n$ from the universe of I.

By $T^{\exists!}$ we denote the causal theory obtained from T'' by adding constraints (17) for all members p of **p**. It is clear that $T^{\exists!}$ is equivalent to the conjunction of T'' with formulas (9) for all $p \in$ **p**.

COROLLARY 7. *(a) An interpretation I of the signature of T with the universe of cardinality > 1 is a model of T iff $I_{\mathbf{p}}^{\mathbf{f}}$ is a model of $T^{\exists!}$. (b) An interpretation J of the signature of $T^{\exists!}$ with the universe of cardinality > 1 is a model of $T^{\exists!}$ iff $J = I_{\mathbf{p}}^{\mathbf{f}}$ for some model I of T.*

EXAMPLE 3, CONTINUED. $T_6^{\exists!}$ consists of the rules

$$\begin{aligned} p(x,y) &\Leftarrow a(x,y), \\ q(x,y) &\Leftarrow b(x,y), \\ p(x,y) \leftrightarrow q(x,y) &\Leftarrow c(x), \\ \neg p(x,y) &\Leftarrow \neg p(x,y), \\ \neg q(x,y) &\Leftarrow \neg q(x,y), \\ \bot &\Leftarrow \neg(\exists! y) p(x,y), \\ \bot &\Leftarrow \neg(\exists! y) q(x,y). \end{aligned}$$

The mapping $I \mapsto I^{fg}_{pq}$ establishes a 1–1 correspondence between the non-singleton models of T_6 and the non-singleton models of $T_6^{\exists!}$.

The (simplified) result of applying the transformation from [Ferraris, Lee, Lierler, Lifschitz, and Yang 2011, Section 5.3] to $T_6^{\exists!}$ is

$$\begin{array}{rl}
a(x,y) & \rightarrow p(x,y),\\
b(x,y) & \rightarrow q(x,y),\\
c(x) \wedge p(x,y) & \rightarrow q(x,y),\\
c(x) \wedge q(x,y) & \rightarrow p(x,y),\\
c(x) \wedge \widehat{p}(x,y) & \rightarrow \widehat{q}(x,y),\\
c(x) \wedge \widehat{q}(x,y) & \rightarrow \widehat{p}(x,y),\\
\neg p(x,y) & \rightarrow \widehat{p}(x,y),\\
\neg q(x,y) & \rightarrow \widehat{q}(x,y),\\
\neg\neg(\exists! y)p(x,y), & \\
\neg\neg(\exists! y)q(x,y), & \\
\neg(p(x) \wedge \widehat{p}(x)), & \\
\neg(\neg p(x) \wedge \neg\widehat{p}(x)), & \\
\neg(q(x) \wedge \widehat{q}(x)), & \\
\neg(\neg q(x) \wedge \neg\widehat{q}(x)). &
\end{array}$$

The non-singleton stable models of the conjunction of the universal closures of these formulas, with the intensional predicates p, q, $\widehat{p}$, $\widehat{q}$, turn into the non-singleton models of $T_6^{\exists!}$ as soon as the interpretations of the auxiliary predicates $\widehat{p}$ and $\widehat{q}$ are dropped.

9 Related Work

The problem addressed in this paper is similar to the problem of eliminating multi-valued propositional constants from a multi-valued causal theory [Giunchiglia, Lee, Lifschitz, McCain, and Turner 2004]. In this sense, our general elimination and modified definite elimination are similar to the elimination methods proposed in [Lee 2005, Section 6.4.2]. On the other hand, modified definite elimination does not introduce rules similar to constraint (6.26) from [Lee 2005], and our proofs (not included in this note) are entirely different: the semantics of multi-valued propositional constants is based on a fixpoint construction and does not refer to syntactic transformations.

Eliminating function constants in the framework of a different nonmonotonic formalism—a version of the stable model semantics—is discussed in [Lin and Wang 2008].

10 Conclusion

In this paper we investigated some of the cases when an explainable function symbol can be eliminated from a first-order causal theory in favor of a predicate symbol. This is a step towards the goal of creating a compiler from the modular action language MAD [Lifschitz and Ren 2006] into answer set programming. It will differ from the current version of COALA [Gebser, Grote, and Schaub 2010] in that it will be applicable to action descriptions that involve non-Boolean fluents and synonymity rules.

Acknowledgments: We are grateful to Joohyung Lee for useful comments. This research was supported by the National Science Foundation under grant IIS-0712113.

References

Doğandağ, S., F. N. Alpaslan, and V. Akman [2001]. Using stable model semantics (SMODELS) in the Causal Calculator (CCALC). In *Proceedings 10th Turkish Symposium on Artificial Intelligence and Neural Networks*, pp. 312–321.

Doğandağ, S., P. Ferraris, and V. Lifschitz [2004]. Almost definite causal theories. In *Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pp. 74–86.

Erdoğan, S. T. and V. Lifschitz [2006]. Actions as special cases. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pp. 377–387.

Ferraris, P., J. Lee, Y. Lierler, V. Lifschitz, and F. Yang [2011]. Representing first-order causal theories by logic programs. *Theory and Practice of Logic Programming*. To appear.

Ferraris, P., J. Lee, and V. Lifschitz [2007]. A new perspective on stable models. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 372–379.

Ferraris, P., J. Lee, and V. Lifschitz [2011]. Stable models and circumscription. *Artificial Intelligence 175*, 236–263.

Ferraris, P., J. Lee, V. Lifschitz, and R. Palla [2009]. Symmetric splitting in the general theory of stable models. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 797–803.

Gebser, M., T. Grote, and T. Schaub [2010]. Coala: a compiler from action languages to ASP. In *Proceedings of European Conference on Logics in Artificial Intelligence (JELIA)*, pp. 169–181.

Gelfond, M. and V. Lifschitz [1988]. The stable model semantics for logic programming. In R. Kowalski and K. Bowen (Eds.), *Proceedings of International Logic Programming Conference and Symposium*, pp. 1070–1080. MIT Press.

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

Giunchiglia, E., J. Lee, V. Lifschitz, N. McCain, and H. Turner [2004]. Nonmonotonic causal theories. *Artificial Intelligence 153(1–2)*, 49–104.

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

Lee, J. [2005]. *Automated Reasoning about Actions*[9]. Ph.D. thesis, University of Texas at Austin.

Lee, J., Y. Lierler, V. Lifschitz, and F. Yang [2010]. Representing synonymity in causal logic and in logic programming[10]. In *Proceedings of International Workshop on Nonmonotonic Reasoning (NMR)*.

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

[9]`http://peace.eas.asu.edu/joolee/papers/dissertation.pdf`
[10]`http://userweb.cs.utexas.edu/users/vl/papers/syn.pdf`

Lifschitz, V. [2002]. Answer set programming and plan generation. *Artificial Intelligence 138*, 39–54.

Lifschitz, V. [2008]. What is answer set programming? In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1594–1597. MIT Press.

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

Lifschitz, V. and W. Ren [2007]. The semantics of variables in action descriptions. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, pp. 1025–1030.

Lifschitz, V. and F. Yang [2010]. Translating first-order causal theories into answer set programming. In *Proceedings of the European Conference on Logics in Artificial Intelligence (JELIA)*, pp. 247–259.

Lin, F. and Y. Wang [2008]. Answer set programming with functions. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pp. 454–465.

Marek, V. and M. Truszczyński [1999]. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm: a 25-Year Perspective*, pp. 375–398. Springer Verlag.

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

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

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

Niemelä, I. [1999]. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence 25*, 241–273.

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

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