Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Mapping Action Language BC to Logic Programs: A Characterization by Postulates

Haodi Zhang and Fangzhen Lin
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong

Abstract

We have earlier shown that the standard mappings from action languages $\mathcal{B}$ and $\mathcal{C}$ to logic programs under answer set semantics can be captured by sets of properties on transition systems. In this paper, we consider action language $\mathcal{BC}$ and show that a standard mapping from $\mathcal{BC}$ action descriptions to logic programs can be similarly captured when the action rules in the descriptions do not have consistency conditions.

Introduction

Logical formalization of the effects of actions has been a main concern in knowledge representation (e.g. (McCarthy and Hayes 1969; Fikes and Nilsson 1971; McCarthy 1977; Finger 1987; McCarthy 1980; Reiter 1980)). There been many approaches proposed (e.g. (McCarthy and Hayes 1969; McCarthy 1980; Lifschitz 1987; Reiter 1991; Shanahan 1995; Lin 1995; McCain and Turner 1995; Thielscher 1998; Giunchiglia et al. 2004)). In this paper we consider various action languages (Gelfond and Lifschitz 1993; 1998; Giunchiglia and Lifschitz 1998; Lee, Lifschitz, and Yang 2013).

Recently Zhang and Lin (2015) proposed a set of properties for characterizing action languages $\mathcal{B}$ (Gelfond and Lifschitz 1998) and $\mathcal{C}$ (Giunchiglia and Lifschitz 1998). They considered static causal rules of the form (l, G), where l is a fluent literal and G a set of fluent literals. What these rules mean depends on the underlying semantics, and is best understood by a mapping to logic programs under the answer set semantics (Gelfond and Lifschitz 1988; 1991). When these rules are interpreted under the $\mathcal{B}$ semantics, a rule like (l, G) corresponds to the logic program rule

$$l' \leftarrow G', \tag{1}$$

where the primed propositions represent the truth values of the corresponding fluents in the successor states. If they are interpreted under the $\mathcal{C}$ semantics, a rule like (l, G) corresponds to the logic program rule

$$l' \leftarrow \mathit{not}\ \overline{G'}, \tag{2}$$

where $\overline{G'}$ denotes the set of complements of the literals in G'. The differences between these two types of logic program rules under the answer set semantics are well-known. The use of the default negation operator in (2) makes it stronger than (1) in the sense that (2) can be applied even when (1) cannot. For instance, the singleton program $\{p \leftarrow q\}$ has the empty set as its only answer set, but the singleton program $\{p \leftarrow \text{ not } \neg q\}$ has $\{p\}$ as its only answer set.

While these rules can be interpreted by either $\mathcal{B}$'s semantics or $\mathcal{C}$'s semantics, they cannot be mixed. This is reflected in incompatible sets of postulates that we used for capturing $\mathcal{B}$ and $\mathcal{C}$ in (Zhang and Lin 2015). However, this presents a problem if one tries to capture the language $\mathcal{BC}$ (Lee, Lifschitz, and Yang 2013). In $\mathcal{BC}$, a static causal rule has the form:

$$l \textbf{ if } G \textbf{ ifcons } H, \tag{3}$$

and means that in every state, if G and it is consistent to assume H, then l. This is like a default rule (Reiter 1980) with G as premises and H justifications. In terms of logic programs, such a rule is translated as

$$l \leftarrow G, \mathit{not\ not}\ H$$

or

$$l \leftarrow G, \mathit{not}\ \overline{H}$$

using strong negation. Thus the semantics of $\mathcal{BC}$ is a combination of $\mathcal{B}$ and $\mathcal{C}$: G is translated according to $\mathcal{B}$ while H according to $\mathcal{C}$. However, given that a static causal rule in (Zhang and Lin 2015) is a pair (l, G), the semantics of $\mathcal{BC}$ cannot be captured in their framework.

To address this problem, we propose to extend static causal rules in (Zhang and Lin 2015) to triples (l, G, H), with G as premises and H justifications. We'll show with this extension, the mixed semantics of $\mathcal{BC}$ on static causal rules can be similarly characterized.

The rest of this paper is organized as follows. First we introduce the syntax of our causal action theories. We then relate these theories to $\mathcal{BC}$ action descriptions, and this provides the $\mathcal{BC}$ semantics to these theories. We then formulate four properties on transition systems and show that they are all satisfied by the $\mathcal{BC}$ semantics. We then show that these four properties provide a unique characterization of the $\mathcal{BC}$ semantics under the so-called permissible mappings. We then relate our result to those in (Zhang and Lin 2015) for action languages $\mathcal{B}$ and $\mathcal{C}$, and conclude the paper.

Causal action theories

Lee, Lifschitz and Yang (2013) introduced action language $\mathcal{BC}$ as a generalization of both languages $\mathcal{B}$ and $\mathcal{C}$. They defined the semantics for their language by a mapping to logic programs under answer set semantics. Our main objective of this paper is to show that this mapping[1] can be characterized by a set of postulates (properties).

To simplify our presentation, instead of considering $\mathcal{BC}$ descriptions, we follow Zhang and Lin (2015) and use a notion of causal action theories as discussed in the introduction. This choice is not material but we made it for several reasons: it allows us to concentrate on the issues regarding possible semantics to static causal rules without explicitly introducing actions, and it makes it easier to compare our result to those in Zhang and Lin (2015). All the definitions and results in this paper can be restated in terms of a class of $\mathcal{BC}$ descriptions that are about one action and have no justifications in dynamic causal rules.

We assume a propositional language $\mathcal{F}$ and call elements in $\mathcal{F}$ *fluents*. A fluent literal is then either a fluent or negation of a fluent.

Our notion of causal action theories is the same as that from (Zhang and Lin 2015) except that now static causal rules are triples instead of pairs, as we mentioned in the introduction. In particular, we also assume that there is just one action whose effects we are interested in. A dynamic causal rule will be a pair (l, G), meaning that if initially G holds, then after the (unnamed) action is performed, l becomes true.

Formally, a causal action theory is a pair (S, D), where S is a set of *static causal rules*, and D a set of *dynamic causal rules*. A static causal rule is a triple of the form (l, G, H), where l is a fluent literal, and G and H are sets of fluent literals such that $G \cup H$ is consistent, i.e. it does not contain both a fluent and its negation, for any fluent. A dynamic causal rule is a pair of the form (l, G), where l is a fluent literal, and G a consistent set of fluent literals.

Intuitively, a dynamic rule (l, G) means that if initially all formulas in G are true, then the (unnamed) action will cause l to be true. As we discussed in the introduction, the static causal rule (l, G, H) is modelled after the static causal rule (3) in $\mathcal{BC}$, and in the following we call l the head, G the premise and H the justification of the rule. Similarly, if (l, G) is a dynamic causal rule, then we call l the head and G the premise of the rule.

Notice that the premise G (in both types of rules) and the justification H are sets of fluent literals. This means that the orders of fluent literals in them are not material. In the following, to make our formula easier to read, when a set of literals occurs in a formula, it stands for the conjunction of literals in the set. So $\neg G \vee H$ stands for

$$\neg[\bigwedge_{l \in G} l] \vee \bigwedge_{l \in H} l.$$

For the special case when G is $\emptyset$, it is taken to be $\top$ (tautology) when it occurs in a formula. We also use $\bot$ to denote a contradiction.

[1] They actually provided two equivalent mappings, one using double default negation *not not* a and the other using strong negation *not* $\neg a$.

The semantics of a causal theory is defined by a set of transitions. The following is basically from (Zhang and Lin 2015).

Definition 1 *A state is a set of fluent atoms, and a transition is a pair of states. A semantic function δ is a mapping from causal action theories to sets of transitions.*

In the following, we identify a state s with a truth assignment: a fluent f is true in s iff $f \in s$. For a formula φ, we write $s \models \varphi$ if φ is true in the truth assignment s.

Intuitively, a transition (s, s') means that the (unnamed) action can be successfully executed in s to yield s'. Thus given a semantic function δ and a causal theory T, $\delta(T)$ is the set of transitions that are possible under the theory T for the action. In particular, if for some state s, there are two states s_1 and s_2 such that both (s, s_1) and (s, s_2) are in $\delta(T)$, then the action is nondeterministic when performed in s according to T under δ.

We say that two causal action theories T_1 and T_2 are equivalent under δ if they have the same transitions: $\delta(T_1) = \delta(T_2)$.

The purpose of this paper is not to propose any specific semantic function, but to study properties that a reasonable semantic function should satisfy. In particular, we are interested in a set of properties that will capture the semantic function given by the semantics of the action language $\mathcal{BC}$ in (Lee, Lifschitz, and Yang 2013) under a natural embedding of our causal theories to $\mathcal{BC}$, which we now describe.

Embedding causal action theories in $\mathcal{BC}$

We now describe a natural mapping from our causal theories to $\mathcal{BC}$ that will provide the standard $\mathcal{BC}$ semantic function that we will then try to capture by postulates.

Recall that our language does not have terms for actions. A causal action theory in our formalism is supposed to axiomatize the action that is of interests to the user. To map our causal theories to $\mathcal{BC}$ descriptions, we need a name for the action, which we denote by A below.

Given a causal action theory $T = (S, D)$, consider the following action description $T_{\mathcal{BC}}$ in language $\mathcal{BC}$: if (l, G, H) is in S, then the static causal rule

$$l \textbf{ if } G \textbf{ ifcons } H$$

is in $T_{\mathcal{BC}}$, and if (l, G) is in D, then the action rule

$$l \textbf{ after } G \textbf{ ifcons } \top$$

is in $T_{\mathcal{BC}}$.

First thing to notice about this mapping is that when a dynamic rule is translated to $\mathcal{BC}$, the **ifcons** part, which represents the consistency condition or justification for the rule to be applied, is empty. While both static causal rules and action rules in $\mathcal{BC}$ can have non-empty **ifcons** parts, we consider only those in static causal rules. Hence in our definition of causal theories, dynamic rules are pairs while static causal rules triples. We did this in order to focus on static causal rules. We believe the results in this paper can be extended to include action rules with non-empty **ifcons** parts as well. We leave this as future work.

This mapping induces a semantic function δ for our causal action theories: $(s, s') \in \delta(T)$ iff (s, A, s') is a transition of $T_{\mathcal{BC}}$ according to $\mathcal{BC}$. In the following, we call this semantic function the $\mathcal{BC}$ semantics. Thus we say that (s, s') is a transition of T under $\mathcal{BC}$ semantics if (s, A, s') is a transition of $T_{\mathcal{BC}}$ according to $\mathcal{BC}$.

Properties

We now consider how the $\mathcal{BC}$ semantics can be captured by a set of postulates on semantic functions.

The following properties on semantic functions are adapted from (Zhang and Lin 2015).

Property 1 below says that states in a transition must satisfy all static causal rules.

Property 1 (Static causal rules are state constraints) *If (s, s') is a transition of a causal action theory (S, D), then*

$$s \models G \wedge H \supset l, \tag{4}$$

$$s' \models G \wedge H \supset l \tag{5}$$

for every static causal rule (l, G, H) in S.

Notice that this property is stated in terms of a given semantic function δ. What the above property really means is that a semantic function δ satisfies Property 1 iff for every causal action theory (S, D), if $(s, s') \in \delta(T)$, then (4) and (5) hold for every static causal rule (l, G, H) in S.

All the properties below are to be read the same way.

The next property says that if the premise or the justification of a static causal rule contains the negation of its head, then this rule is basically a constraint according to Property 1.

Property 2 (Immediate negative loops can be eliminated) *For any causal action theory (S, D), any static rule $r = (l, \{\neg l\} \cup G, H) \in S$ or $r = (l, G, \{\neg l\} \cup H) \in S$ and any states s and s' that satisfy (4) and (5) for all rules in S (i.e. Property 1 holds for S), we have that (s, s') is a transition of (S, D) iff (s, s') is a transition of $(S \setminus \{r\}, D)$.*

Before giving our next property we first define the *dependency graph* of a causal action theory. Given a causal action theory $T = (S, D)$, its dependency graph is the directed graph such that

- its vertices are arbitrary fluent literals, and
- it has a P-edge from l_1 to l_2 iff S contains a static causal rule (l_1, G, H) such that $l_2 \in G$.
- it has a J-edge from l_1 to l_2 iff S contains a static causal rule (l_1, G, H) such that $l_2 \in H$.

We call a loop L in a dependency graph a *persistent loop* if all edges in L are P-edges, and a *non-persistent loop* if there is at least one J-edge in L. In a dependency graph, a fluent literal l is *supported by* a loop L if there is a directed path from l to some fluent literal in L. In particular, a fluent literal in a loop is supported by the loop itself. For a causal action theory T, we define the *persistent set* of T, denoted by $\Delta(T)$, the set of all fluents f such that neither f_i nor $\neg f_i$ is supported by any non-persistent loop.

Our next property says that if there are no possible interactions between the static and dynamic causal rules, then the values of the fluents in the persistent set will not change. In other words, there are no ramifications for them.

Definition 2 *A causal action theory $T = (S, D)$ is said to have no overlap between static and dynamic causal rules if whenever $(l, G) \in D$ and $(l', G', H') \in S$, the fluent in l does not occur in $G' \cup H' \cup \{l'\}$.*

Property 3 (No ramification for persistent set when there is no overlap between static and dynamic causal rules) *For every causal action theory $T = (S, D)$ that has no overlap between static and dynamic causal rules, if a pair (s, s') of states is a transition of T, then*

1. *$s^+ \subseteq s'$ and $s^- \cap s' = \emptyset$, where $s^+ = \{f \mid (f, G) \in D, s \models G\}$ and $s^- = \{f \mid (\neg f, G) \in D, s \models G\}$,*
2. *for all fluent f_i such that $f_i \in \Delta(T)$ and $f_i \notin s^+ \cup s^-$, we have that $f_i \in s$ iff $f_i \in s'$,*
3. *s and s' satisfy (4) and (5), respectively, for every rule in S (i.e. Property 1 is satisfied).*

Furthermore, if $s^- \cap s^+ = \emptyset$, then $(s, (s \setminus s^-) \cup s^+)$ is a transition of T provided that Property 1 is satisfied.

The next property is about how justifications can be combined.

Property 4 (Justifications of static causal rules can be combined) *Let $T = (S, D)$ be a causal action theory. If both*

$$r_1 = (l, G, H \cup \{l_1\})$$

and

$$r_2 = (l, G \setminus \{l_1\}, (H \setminus \{l_1\}) \cup \{\neg l_1\})$$

are in S, and $\{l_1\} \cup G \cup H$ is consistent, then T and $T' = (S', D)$ are equivalent, where

$$S' = (S \setminus \{r_1, r_2\}) \cup \{(l, G \setminus \{l_1\}, H)\}.$$

As expected, the $\mathcal{BC}$ semantics satisfies all the properties.

Proposition 1 *Let $\delta_{\mathcal{BC}}$ be the semantic function induced by the $\mathcal{BC}$ according to the mapping from causal action theories to $\mathcal{BC}$ descriptions given in the last section. $\delta_{\mathcal{BC}}$ satisfies Properties 1, 2, 3 and 4.*

We now show that in a sense, these are also characterizing properties for $\mathcal{BC}$.

Permissible mappings from causal action theories to logic programs

Following Zhang and Lin (2015), we consider characterizing an action semantics by a set of properties that uniquely determines a mapping from causal theories to logic programs with answer set semantics. This fits $\mathcal{BC}$ well as its semantics was given by a mapping to logic programs (Lee, Lifschitz, and Yang 2013). Briefly, we extend the notion of permissible mappings in (Zhang and Lin 2015) to our causal action theories and show that there is a unique such permissible mapping that satisfies the properties in the last section and this permissible mapping yields exactly the $\mathcal{BC}$ semantics. Given that $\mathcal{BC}$ is an extension of both $\mathcal{B}$ and $\mathcal{C}$, we will discuss how this result relates to similar characterizations of $\mathcal{B}$ and $\mathcal{C}$ in (Zhang and Lin 2015).

The following definitions are adapted from those in (Zhang and Lin 2015), by taking into account our new notion of causal action theories.

Given a language consisting of a set $\mathcal{F}$ of fluents, we map causal action theories to logic programs (with constraints and classical negation) in the language

$$\mathcal{L} = \mathcal{F} \cup \neg\mathcal{F} \cup \mathcal{F}' \cup \neg\mathcal{F}',$$

where $\neg\mathcal{F} = \{\neg f \mid f \in \mathcal{F}\}$, $\mathcal{F}' = \{f' \mid f \in \mathcal{F}\}$, and similarly for $\neg\mathcal{F}'$. We assume that for each $f \in \mathcal{F}$, f' is a new atom.

Definition 3 *Let ξ be a mapping from causal action theories to logic programs. The induced semantic function by ξ is defined as follows: a pair (s, s') of states in $\mathcal{F}$ is a transition of a causal action theory T iff there is an answer set A of $\xi(T)$ such that*

$$s = \{f \mid f \in \mathcal{F}, f \in A\}, \tag{6}$$
$$s' = \{f \mid f' \in \mathcal{F}', f' \in A\}. \tag{7}$$

In the following, we say a mapping from causal action theories to logic programs satisfies a property if the induced semantic function of this mapping satisfies the property. Similarly, we say that (s, s') is a transition of T under such a mapping if (s, s') is a transition of T under the semantic function induced by this mapping.

In addition to satisfying some semantic properties, we also want a mapping to be "modular", and in the case of logic programs, to make use of a "standard way" of encoding the frame axioms. This is the intuitions behind the following definition of *permissible* mappings.

Definition 4 *A mapping ξ from causal action theories to logic programs is said to be* permissible *if it is*

1. ***(Compositional)*** *For each $T = (S, D)$,*

$$\xi(T) = \bigcup_{r \in S} \xi^s(r) \cup \bigcup_{r \in D} \xi^d(r) \cup B, \tag{8}$$

where

- *$\xi^s(r)$ is the translation on static causal rules, and for a rule (l, G, H), $\xi^s(l, G, H)$ is a logic program consisting of the following constraints:*

$$\leftarrow \bar{l}, G, H \tag{9}$$
$$\leftarrow \overline{l'}, G', H' \tag{10}$$

and some rules of form

$$l' \leftarrow F_1',\ not\ F_2'$$

where $\bar{l}$ is the complement of l: $\overline{f} = \neg f$, and $\overline{\neg f} = f$, $W' = l_1', ..., l_n'$ if $W = \{l_1, \cdots, l_n\}$, and F_1' and F_2' are sets of atoms in $\mathcal{F}'_{G\cup H} \cup \neg\mathcal{F}'_{G\cup H}$, where $\mathcal{F}_{G\cup H}$ are all fluents that appear in $G \cup H$, and $not\ \{l_1', ..., l_k'\}$ is $not\ l_1', ..., not\ l_k'$;

- *$\xi^d(r)$ is the translation on dynamic causal rules, and for a rule (l, G), $\xi^d(l, G)$ is a set of rules of the form*

$$l' \leftarrow F,$$

where F is a set of fluent literals in $\mathcal{F} \cup \neg\mathcal{F}$;

- *B, the base, is the following set of rules that are independent of the given theory T: for each $f \in \mathcal{F}$,*

$$f \leftarrow\ not\ \neg f, \tag{11}$$
$$\neg f \leftarrow\ not\ f. \tag{12}$$
$$f' \leftarrow f,\ not\ \neg f', \tag{13}$$
$$\neg f' \leftarrow \neg f,\ not\ f', \tag{14}$$

2. ***(Uniform)*** *Both ξ^s and ξ^d are homomorphic under any permutation σ on the set $\mathcal{F}$ of fluents: $\xi^s(r^\sigma) = (\xi^s(r))^\sigma$, and $\xi^d(r^\sigma) = (\xi^d(r))^\sigma$, where r^σ is obtained from r by a fluent substitution according to σ, and for any logic program P, P^σ is the program obtained from P by a fluent substitution according to σ.*
And Both ξ^s and ξ^d are also homomorphic under permutation of any fluent $f \in \mathcal{F}$ with its negation: $\xi^s(r^f) = (\xi^s(r))^f$, and $\xi^d(r^f) = (\xi^d(r))^f$, where r^f is the causal rule obtained from r by swapping f and $\neg f$, and for any logic program P, P^f is the program obtained from P by swapping f and $\neg f$.

We can now state the main result of our paper, which basically says that there is exactly one permissible mapping that satisfies the semantic properties given in the last section, and this permissible mapping yields exactly the $\mathcal{BC}$ semantics.

Theorem 1 *Let $\xi_{\mathcal{BC}}$ be the following mapping from causal action theories to logic programs: for each $T = (S, D)$,*

$$\xi_{\mathcal{BC}}(T) = \bigcup_{r \in S} \xi^s_{\mathcal{BC}}(r) \cup \bigcup_{r \in D} \xi^d_{\mathcal{BC}}(r) \cup B, \tag{15}$$

where

- *B is the base,*
- *for each static causal rule (l, G, H), $\xi^s_{\mathcal{BC}}(l, G, H)$ is the set of rules consisting of constraints (9) and (10) and the following rule*

$$l' \leftarrow G',\ not\ \overline{H'}. \tag{16}$$

- *for each dynamic causal rule (l, G), $\xi^d_{\mathcal{BC}}(l, G)$ is the singleton set consisting of the following rule*

$$l' \leftarrow G. \tag{17}$$

We have

1. *The mapping $\xi_{\mathcal{BC}}$ is permissible and satisfies Properties 1, 2, 3 and 4. Furthermore, for any causal action theory T, (s, s') is a transition of T under $\xi_{\mathcal{BC}}$ iff it is a transition of T under the $\mathcal{BC}$ semantics.*
2. *If ξ is a permissible mapping that satisfies Properties 1, 2, 3 and 4, then ξ is strongly equivalent to $\xi_{\mathcal{BC}}$ in the following sense: for any static causal rule (l, G, H), $\xi^s(l, G, H)$ and $\xi^s_{\mathcal{BC}}(l, G, H)$ are strongly equivalent under the base B, and for any dynamic causal rule (l, G), $\xi^d(l, G)$ and $\xi^d_{\mathcal{BC}}(l, G)$ are strongly equivalent under the base B as well.*

See the detailed proof in our supplemental material. One can see that the mapping $\xi_{\mathcal{BC}}$ is essentially the same as the one given in Section 8 of (Lee, Lifschitz, and Yang 2013) using strong negation. They also gave a mapping using double default negation (*not not p*) in Section 4 of their paper.

While these two mappings yield the same semantics for language $\mathcal{BC}$, they are not strongly equivalent. Thus one can see that our uniqueness result holds only because we do not have double default negation. On the other hand, if double default negation is allowed, one can add more constraints and rules to the base B so that the two mappings given in (Lee, Lifschitz, and Yang 2013) become strongly equivalent under the new base.

The $\mathcal{B}$ and $\mathcal{C}$

The action language $\mathcal{BC}$ includes $\mathcal{B}$ and $\mathcal{C}$ as special cases: if an action description in $\mathcal{BC}$ includes only static causal rules of the form l **if** G **ifcons** $\top$ (those with the empty justification), then it is essentially a $\mathcal{B}$ action description, and similarly if it only includes static causal rules of the form l **if** $\top$ **ifcons** H (those with the empty premise), then it is essentially a $\mathcal{C}$ action description. We now discuss what our result suggests for $\mathcal{B}$ and $\mathcal{C}$, and how they compared to the results in (Zhang and Lin 2015).

Definition 5 *A causal action theory (S, D) is called a $\mathcal{B}$-theory if for all $(l, G, H) \in S$, $H = \emptyset$. It is called a $\mathcal{C}$-theory if for all $(l, G, H) \in S$, $G = \emptyset$.*

For $\mathcal{B}$-theories, Property 4 trivially holds for every semantic function. Furthermore, Property 3 can be much simplified.

Proposition 2 *For $\mathcal{B}$-theories, Property 3 is equivalent to the following property:*

Property 5 **(No ramification when there is no overlap between static and dynamic causal rules)** *For every $\mathcal{B}$-theory $T = (S, D)$ that has no overlap between static and dynamic causal rules, then a pair (s, s') of states is a transition of T iff*

- $s' = (s \setminus s^-) \cup s^+$,
- $s^+ \cap s^- = \emptyset$,
- *both s and s' satisfy the static causal rules in S (Property 1),*

where $s^- = \{f \mid (\neg f, G) \in D, s \models G\}$ and $s^+ = \{f \mid (f, G) \in D, s \models G\}$.

Proof: If T is a $\mathcal{B}$-theory, then its dependency graph has no J-edges. Thus its persistent set is the entire set of fluents. ■

Thus Theorem 1 suggests that for $\mathcal{B}$-theories, the Properties 1, 2, and 5 would uniquely determine a permissible mapping that would yield a semantics that is the same as the one for $\mathcal{B}$ language. Indeed, we have the following result that corresponds to Theorem 1 in (Zhang and Lin 2015).

Theorem 2 *Let $\xi_{\mathcal{B}}$ be the following mapping from $\mathcal{B}$-theories to logic programs: for each $T = (S, D)$,*

$$\xi_{\mathcal{B}}(T) = \bigcup_{r \in S} \xi^s_{\mathcal{B}}(r) \cup \bigcup_{r \in D} \xi^d_{\mathcal{B}}(r) \cup B, \tag{18}$$

where

- *B is the base as in the definition of permissible mappings,*
- *for each static causal rule $(l, G, \emptyset)$, $\xi^s_{\mathcal{B}}(l, G, \emptyset)$ is the set of rules consisting of constraints (9) and (10) (with H and H' deleted) and the following rule*

$$l' \leftarrow G'. \tag{19}$$

- *for each dynamic causal rule (l, G), $\xi^d_{\mathcal{B}}(l, G)$ is the singleton set consisting of the following rule*

$$l' \leftarrow G. \tag{20}$$

We have

1. *The mapping $\xi_{\mathcal{B}}$ is permissible and satisfies Properties 1, 2 and 5.*
2. *If ξ is a permissible mapping that satisfies Properties 1, 2, and 5, then ξ is strongly equivalent to $\xi_{\mathcal{B}}$ on $\mathcal{B}$-theories in the following sense: for any static causal rule $(l, G, \emptyset)$, $\xi^s(l, G, \emptyset)$ and $\xi^s_{\mathcal{B}}(l, G, \emptyset)$ are strongly equivalent under the base B, and for any dynamic causal rule (l, G), $\xi^d(l, G)$ and $\xi^d_{\mathcal{B}}(l, G)$ are strongly equivalent under the base B as well.*

Proof:[$Sketch.$] We use the results in (Zhang and Lin 2015) to prove. We call their causal action theories *simple causal action theories*, to distinguish from those in this paper. And we denote Property i in their paper as Pi. Let M be following mapping from simple causal action theories to $\mathcal{B}$-theories: for $T = (S, D)$, $M(T) = (\{M(r) \mid r \in S\}, D)$ where for a static rule (l, G) in S, $M(l, G) = (l, G, \emptyset)$. We call a mapping from $\mathcal{B}$-theories to logic programs a *$\mathcal{B}$-mapping*, and a mapping from simple causal action theories to logic programs a *simple mapping*. And let η the following mapping from a $\mathcal{B}$-mapping to a simple mapping: for any simple causal theory $T = (S, D)$,

$$\eta(\xi)(T) = \bigcup_{r \in S} \eta(\xi)^s(r) \cup \bigcup_{r \in D} \eta(\xi)^d(r) \cup B, \tag{21}$$

where $\eta(\xi)^s(r) = \xi^s(M(r))$ for a static rule r, and $\eta(\xi)^d(r) = \xi^d(r)$ for a dynamic rule r. It can be checked that, considering only $\mathcal{B}$-theories, ξ is a permissible mapping iff $\eta(\xi)$ is also permissible according to the definition of permissible mappings in (Zhang and Lin 2015). And Property 1 is essentially equivalent with P2, in the sense that a permissible mapping ξ satisfies Property 1 iff $\eta(\xi)$ satisfies P2. And Property 2 is essentially equivalent with P3, and Property 5 is essentially equivalent with P4. According to Theorem 1 in (Zhang and Lin 2015), we have

1. $\eta(\xi_{\mathcal{B}})$ is permissible and satisfies P2, P3 and P4. So $\xi_{\mathcal{B}}$ is permissible and satisfies Properties 1, 2 and 5.
2. If ξ is a permissible mapping that satisfies Properties 1, 2 and 5, then $\eta(\xi)$ satisfies P2, P3 and P4, so $\eta(\xi)^s(l, G)$ is strongly equivalent with $\eta(\xi_{\mathcal{B}})^s(l, G)$ under $B \cup Q$ where Q is the constraints for (l, G), i.e. $\xi^s(l, G, \emptyset)$ is strongly equivalent with $\xi^s_{\mathcal{B}}(l, G, \emptyset)$ under $B \cup Q_M$ where Q_M is the constraints for $(l, G, \emptyset)$, which is the same with Q, and $\xi^d(l, G)$ is strongly equivalent with $\xi^d_{\mathcal{B}}(l, G)$ for a dynamic rule (l, G).

■

For $\mathcal{C}$ language, the set of properties in (Zhang and Lin 2015) includes Property 1, 2, 4, and the following property which is weaker than Property 3:

Property 6 (No ramification when there is no loop and no overlap between static and dynamic causal rules) *For every $\mathcal{C}$-theory $T = (S, D)$ that has no overlap between static and dynamic causal rules, and there is no loop in the dependency graph, a pair (s, s') of states is a transition of T iff*

- $s' = (s \setminus s^-) \cup s^+$,
- $s^+ \cap s^- = \emptyset$, *and*
- *both s and s' satisfy the static causal rules in S (Property 1),*

where $s^- = \{f \mid (\neg f, G) \in D, s \models G\}$ and $s^+ = \{f \mid (f, G) \in D, s \models G\}$.

Proposition 3 *For $\mathcal{C}$-theories, Property 3 implies Property 6*

Proof: If T is a $\mathcal{C}$-theory whose dependency graph is acyclic, then its persistent set is the entire set of fluents. ■

The following result corresponds to Theorem 2 in (Zhang and Lin 2015)

Theorem 3 *Let $\xi_{\mathcal{C}}$ be the following mapping from $\mathcal{C}$-theories to logic programs: for each $T = (S, D)$,*

$$\xi_{\mathcal{C}}(T) = \bigcup_{r \in S} \xi^s_{\mathcal{C}}(r) \cup \bigcup_{r \in D} \xi^d_{\mathcal{C}}(r) \cup B, \tag{22}$$

where

- *B is the base as in the definition of permissible mappings,*
- *for each static causal rule $(l, \emptyset, H)$, $\xi^s_{\mathcal{C}}(l, \emptyset, H)$ is the set of rules consisting of constraints (9) and (10) (with G and G' deleted) and the following rule*

$$l' \leftarrow \ not\ \overline{H'}. \tag{23}$$

- *for each dynamic causal rule (l, G), $\xi^d_{\mathcal{C}}(l, G)$ is the singleton set consisting of the following rule*

$$l' \leftarrow G. \tag{24}$$

We have

1. *The mapping $\xi_{\mathcal{C}}$ is permissible and satisfies Properties 1, 2, 4 and 6.*
2. *If ξ is a permissible mapping that satisfies Properties 1, 2, 4 and 6, then ξ is strongly equivalent to $\xi_{\mathcal{C}}$ on $\mathcal{C}$-theories in the following sense: for any static causal rule $(l, \emptyset, H)$, $\xi^s(l, \emptyset, H)$ and $\xi^s_{\mathcal{C}}(l, \emptyset, H)$ are strongly equivalent under the base B, and for any dynamic causal rule (l, G), $\xi^d(l, G)$ and $\xi^d_{\mathcal{C}}(l, G)$ are strongly equivalent under the base B as well.*

Proof:[$Sketch.$] Let M be following mapping from simple causal action theories to $\mathcal{C}$-theories: for $T = (S, D)$, $M(T) = (\{M(r) \mid r \in S\}, D)$ where for a static rule (l, H) in S, $M(l, H) = (l, \emptyset, H)$. We call a mapping from $\mathcal{C}$-theories to logic programs a *$\mathcal{C}$-mapping*. And let η the following mapping from a $\mathcal{C}$-mapping to a simple mapping: for any simple causal theory $T = (S, D)$,

$$\eta(\xi)(T) = \bigcup_{r \in S} \eta(\xi)^s(r) \cup \bigcup_{r \in D} \eta(\xi)^d(r) \cup B, \tag{25}$$

where $\eta(\xi)^s(r) = \xi^s(M(r))$ for a static rule r, and $\eta(\xi)^d(r) = \xi^d(r)$ for a dynamic rule r. ξ is permissible iff $\eta(\xi)$ is also permissible according to the definition of permissible mappings in (Zhang and Lin 2015). As ξ is a permissible mapping, $\eta(\xi)$ satisfies P1. And as ξ satisfies Properties 1, 2, 4 and 6, $\eta(\xi)$ satisfies P2, P3, P4$'$ and P5 in (Zhang and Lin 2015). According to Theorem 2 in (Zhang and Lin 2015), we have

1. As $\eta(\xi_{\mathcal{C}})$ is permissible and satisfies P1, P2, P3, P4$'$ and P5, $\xi_{\mathcal{C}}$ is permissible and satisfies Properties 1, 2, 4 and 5.
2. If ξ is a permissible mapping that satisfies Properties 1, 2, 4 and 5, then $\eta(\xi)$ satisfies P1, P2, P3, P4$'$ and P5. So $\eta(\xi)^s(l, H)$ is strongly equivalent with $\eta(\xi_{\mathcal{C}})^s(l, H)$ under $B \cup Q$ where Q is the constraints for (l, H), i.e. $\xi^s(l, \emptyset, H)$ is strongly equivalent with $\xi^s_{\mathcal{C}}(l, \emptyset, H)$ under $B \cup Q_M$ where Q_M is the constraints for $(l, \emptyset, H)$, which is the same with Q, and $\xi^d(l, G)$ is strongly equivalent with $\xi^d_{\mathcal{C}}(l, G)$ for a dynamic rule (l, G).

■

Concluding remarks

We considered the action language $\mathcal{BC}$, which is recently proposed by Lee *et al.* (2013) as a generalization of both $\mathcal{B}$ and $\mathcal{C}$. By extending Zhang and Lin's results for $\mathcal{B}$ and $\mathcal{C}$, we showed that the mapping from $\mathcal{BC}$ action descriptions to logic programs can be uniquely captured up to a notion of strong equivalence by four properties, provided that the action rules in these action descriptions do not have non-empty consistency conditions. We believe these results can be extended to include action rules with non-empty consistency conditions by allowing dynamic causal rules to be triples like static causal rules, and then extending corresponding cancepts like permissible mappings and dependency graphs. We leave this as a future work.

Besides providing an alternative characterization for the semantics of action language $\mathcal{BC}$, this work serves as another evidence that the approach advocated by Zhang and Lin (2015) to study causal action theories by postulates is fruitful and further work is needed for a systematic study of these postulates as a whole, not just as a means to capture some known action languages.

References

Fikes, R. E., and Nilsson, N. J. 1971. Strips: A new approach to the application of theorem proving to problem solving. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI–71)*, 608–620.

Finger, J. J. 1987. *Exploiting Constraints in Design Synthesis*. Ph.D. Dissertation, Stanford University, Stanford, CA, USA. UMI Order No. GAX87-20386.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proceeding of Fifth International Conference and Symposium on Logic Programming (ICLP–88)*, 1070–1080.

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

Gelfond, M., and Lifschitz, V. 1993. Representing action and change by logic programs. *Journal of Logic Programming* 17:301–322.

Gelfond, M., and Lifschitz, V. 1998. Action languages. *Electronic Transactions on Artificial Intelligence* 3:195–210.

Giunchiglia, E., and Lifschitz, V. 1998. An action language based on causal explanation: preliminary report. In *Proceedings of National Conference on Artificial Intelligence (AAAI–98)*, 623–630.

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

Lee, J.; Lifschitz, V.; and Yang, F. 2013. Action language BC: preliminary report. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI–13), Beijing, China, August 3-9, 2013*.

Lifschitz, V. 1987. Formal theories of action (preliminary report). In *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI–87). Milan, Italy, August 1987*, 966–972.

Lin, F. 1995. Embracing causality in specifying the indirect effects of actions. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI–95), Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, 1985–1993.

McCain, N., and Turner, H. 1995. A causal theory of ramifications and qualifications. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI–95), Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, 1978–1984.

McCarthy, J., and Hayes, P. 1969. Some Philosophical Problems From the Standpoint of Artificial Intelligence. In Meltzer, B., and Donald Michie., eds., *Machine Intelligence 4*. 463–502.

McCarthy, J. 1977. Epistemological problems of artificial intelligence. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI–77) - Volume 2*, 1038–1044.

McCarthy, J. 1980. Circumscription: A form of non-monotonic reasoning. *Artificial Intelligence* 13(1-2):27–39. Special Issue on Non-Monotonic Logic.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13(1-2):81–132.

Reiter, R. 1991. The frame problem in situation the calculus: A simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., ed., *Artificial Intelligence and Mathematical Theory of Computation*. San Diego, CA, USA: Academic Press Professional, Inc. 359–380.

Shanahan, M. 1995. A circumscriptive calculus of events. *Artificial Intelligence* 77(2):249–284.

Thielscher, M. 1998. Introduction to the fluent calculus. *Electronic Transactions on Artificial Intelligence* 2:179–192.

Zhang, H., and Lin, F. 2015. Characterizing causal action theories and their implementations in answer set programming: Action languages b, c, and beyond. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI–15), Buenos Aires, Argentina, July 25-31, 2015*, 3285–3291.