Turner's Logic of Universal Causation, Propositional Logic, and Logic Programming

Jianmin Ji[1] and Fangzhen Lin[2]

[1] School of Computer Science and Technology
University of Science and Technology of China, Hefei, China
jianmin@ustc.edu.cn
[2] Department of Computer Science and Engineering
The Hong Kong University of Science and Technology, Hong Kong
flin@cs.ust.hk

Abstract. Turner's logic of universal causation is a general logic for nonmonotonic reasoning. It has its origin in McCain and Turner's causal action theories which have been translated to propositional logic and logic programming with nested expressions. In this paper, we propose to do the same for Turner's logic, and show thatTurner's logic can actually be mapped to McCain and Turner's causal theories. These results can be used to construct a system for reasoning in Turner's logic.

1 Introduction

Turner's logic of universal causation [17], called UCL, is a nonmonotonic modal logic that generalizes McCain and Turner's causal action theories [15]. The idea is to use the modal operator $\mathbf{C}$ to specify the statement that a proposition is "caused". For instance, $\psi \supset \mathbf{C}\phi$ says that ϕ is caused whenever ψ obtains.

McCain and Turner's causal action theories have been the basis for the semantics of several expressive action languages, such as $\mathcal{C}$ and $\mathcal{C}+$ [11,5]. They have been translated to propositional logic and logic programming. Ferraris [2] provided a translation from causal theories to disjunctive logic programs. Lee [9] proposed a conversion from causal theories to propositional logic. In this paper, we consider UCL, and show that UCL theories can be converted to propositional theories. We also show that they can be converted to logic programs with nested expressions in polynomial size with polynomial number of new variables. This result improves and generalizes Turner's linear and modular translation from a fragment of UCL to disjunctive logic programs [17]. Furthermore we show that both Ferraris and Lee's translations are special cases of our translations, just as McCain and Turner's causal theories are special theories in UCL. Our motivation for this work is to use the translations to implement a system for computing UCL theories via SAT solvers or ASP solvers, like the system CCalc[3] for causal theories.

[3] http://www.cs.utexas.edu/~tag/ccalc/.

This paper is organized as follows. Section 2 reviews UCL and logic programming. Section 3 shows how Turner's logic can be mapped to propositional logic. Section 4 considers mapping UCL theories to logic programs with nested expressions. Section 5 outlines how the translations here are related to Ferraris and Lee's translations. Finally, Section 6 concludes this paper.

2 Preliminaries

2.1 Propositional languages

We assume a propositional language with two zero-place logical connectives $\top$ for tautology and $\bot$ for contradiction. We denote by *Atom* the set of atoms, the signature of our language, and *Lit* the set of literals: $Lit = Atom \cup \{\neg a \mid a \in Atom\}$. A set I of literals is called *complete* if for each atom a, exactly one of $\{a, \neg a\}$ is in I. Given a literal l, the *complement* of l, written $\bar{l}$ below, is $\neg a$ if l is a and a if l is $\neg a$, where a is an atom. For a set L of literals, we let $\overline{L} = \{\, \bar{l} \mid l \in L\,\}$.

In this paper, we identify an interpretation with a complete set of literals. If I is a complete set of literals, we use it as an interpretation when we say that it is a model of a formula, and we use it as a set of literals when we say that it entails a formula.

2.2 Turner's logic of universal causation

The language of Turner's logic of universal causation (UCL) [17] is a modal propositional language with a modal operator $\mathbf{C}$. UCL *formulas* are propositional formulas with unary modal operator $\mathbf{C}$. A UCL *theory* is a set of UCL formulas.

The semantics of UCL is defined through causally explained interpretations. A UCL *structure* is a pair $(I, \mathcal{S})$ such that I is an interpretation, and $\mathcal{S}$ is a set of interpretations to which I belongs. The truth of a UCL sentence in a UCL structure is defined by the standard recursions over the propositional connectives, plus the following two conditions:

$$(I, \mathcal{S}) \models a \text{ iff } I \models a \quad \text{(for any atom } a)$$
$$(I, \mathcal{S}) \models \mathbf{C}\phi \text{ iff for all } I' \in \mathcal{S},\ (I', \mathcal{S}) \models \phi$$

Given a UCL theory T, we write $(I, \mathcal{S}) \models T$ to mean that $(I, \mathcal{S}) \models \phi$, for every $\phi \in T$. In this case, we say that $(I, \mathcal{S})$ is a *model* of T. We also say that $(I, \mathcal{S})$ is an I*-model* of T, emphasizing the distinguished interpretation I.

Let T be a UCL theory. An interpretation I is *causally explained* by T if $(I, \{I\})$ is the unique I-model of T.

Note that, if there is a nested occurrence of $\mathbf{C}$, the $\mathbf{C}$ that occurs in the range of another $\mathbf{C}$ can be equivalently[4] removed [17]. In the paper, we only consider UCL formulas with no nested occurrences of $\mathbf{C}$. A formula of the form $\mathbf{C}\phi$, where ϕ is a propositional formula, is called a $\mathbf{C}$*-atom*. Then these UCL formulas are constructed from $\mathbf{C}$-atoms, propositional atoms and connectives.

[4] In the sense that, two formulas have the same set of UCL models.

2.3 Logic Programming

A *nested expression* is built from literals using the 0-place connectives $\top$ and $\bot$, the unary connective "*not*" and the binary connective "," and ";".

A *logic program* with nested expressions is a finite set of rules of the form $F \leftarrow G$, where F and G are nested expressions.

The *answer set* of a logic program with nested expressions is defined as in [12]. Given a nested expression F and a set S of literals, we define when S satisfies F, written $S \models F$ below, recursively as follows (l is a literal and G is a nested expression):

- $S \models l$ if $l \in S$,
- $S \models \top$ and $S \not\models \bot$,
- $S \models not\, F$ if $S \not\models F$,
- $S \models F, G$ if $S \models F$ and $S \models G$, and
- $S \models F; G$ if $S \models F$ or $S \models G$.

S satisfies a rule $F \leftarrow G$ if $S \models F$ whenever $S \models G$. S satisfies a logic program P, written $S \models P$, if S satisfies all rules in P.

The *reduct* P^S of P related to S is the result of replacing every maximal subexpression of P that has the form $not\, F$ with $\bot$ if $S \models F$, and with $\top$ otherwise.

Let P be a logic program without *not*, the *answer set* of P is any minimal consistent subset S of Lit that satisfies P. We use $\Gamma_P(S)$ to denote the set of answer sets of P^S. Now a consistent set S of literals is an *answer set* of P iff $S \in \Gamma_P(S)$.

Every logic program with nested expressions can be equivalently translated to disjunctive logic programs with disjunctive rules of the form

$$l_1 \vee \cdots \vee l_k \leftarrow l_{k+1}, \ldots, l_t, not\, l_{t+1}, \ldots, not\, l_m, not\, not\, l_{m+1}, \ldots, not\, not\, l_n,$$

where $n \geq m \geq t \geq k \geq 0$ and $l_1, \ldots, l_n$ are propositional literals. A disjunctive logic program can be computed by disjunctive ASP solvers such as claspD [1], DLV [10], GNT [7] and cmodels [6].

3 From Turner's Logic of Universal Causation to Propositional Logic

Before presenting the translation, we provide some notations. Given a UCL formula F, let $Atom_{\mathbf{C}}(F) = \{\, \phi \mid \mathbf{C}\phi \text{ is a } \mathbf{C}\text{-atom occurring in } F \,\}$. Given a UCL theory T, we let $Atom_{\mathbf{C}}(T) = \bigcup_{F \in T} Atom_{\mathbf{C}}(F)$.

We use $tr_p(F)$ to denote the propositional formula obtained from the UCL formula F by replacing each occurrence of a $\mathbf{C}$-atom $\mathbf{C}\phi$ by a new propositional atom a_ϕ w.r.t. ϕ.

Given two propositional formulas ϕ and ψ, we use ϕ^ψ to denote the propositional formula obtained from ϕ by replacing each occurrence of an atom a with a new atom a^ψ w.r.t. ψ.

The following proposition provides a specification of the propositional formula whose models are related to models of a UCL theory.

Proposition 1. *Let T be a UCL theory. A UCL structure $(I, \mathcal{S})$ is a model of T if and only if there exists a model I^* of the propositional formula*

$$\bigwedge_{F\in T} tr_p(F) \wedge \bigwedge_{\phi\in Atom_{\mathbf{C}}(T)} (a_\phi \supset \phi)$$
$$\wedge \bigwedge_{\psi\in Atom_{\mathbf{C}}(T)} \left((\neg a_\psi \wedge \psi) \supset \left(\bigwedge_{\phi\in Atom_{\mathbf{C}}(T)} (a_\phi \supset \phi^\psi) \wedge \neg\psi^\psi \right) \right), \quad (1)$$

such that $I^* \cap Lit = I$ and for each $\phi \in Atom_{\mathbf{C}}(T)$, $a_\phi \in I^*$ iff $\mathcal{S} \models \phi$.

Proof. "$\Rightarrow$" $(I, \mathcal{S})$ is a model of T, then $I \in \mathcal{S}$. If $\mathcal{S} \not\models \psi$ and $I \models \psi$, then there exists another interpretation $I' \in \mathcal{S}$ such that $I' \models \neg\psi$. Thus, we can create an interpretation I^* such that

$$I^* = I \cup \{a_\phi \mid \phi \in Atom_{\mathbf{C}}(T) \text{ and } \mathcal{S} \models \phi\} \cup \{\neg a_\phi \mid \phi \in Atom_{\mathbf{C}}(T) \text{ and } \mathcal{S} \not\models \phi\}$$
$$\cup \bigcup_{\psi\in Atom_{\mathbf{C}}(T), \mathcal{S}\models\psi} \{l^\psi \mid l \in I\} \cup \bigcup_{\psi\in Atom_{\mathbf{C}}(T), \mathcal{S}\not\models\psi, \exists I'. I'\in\mathcal{S}, I'\models\neg\psi} \{l^\psi \mid l \in I'\}.$$

Clearly, $I^* \models (1)$.

"$\Leftarrow$" $I^* \models T$. Let $I = I^* \cap Lit$ and $\mathcal{S} = \{I' \mid \text{if } I^* \models a_\phi \text{ for some } \phi \in Atom_{\mathbf{C}}(T), \text{ then } I' \models \phi\}$. Note that, $I^* \models \bigwedge_{\phi\in Atom_{\mathbf{C}}(T)} (a_\phi \supset \phi)$, then $I \in \mathcal{S}$. For each $\phi \in Atom_{\mathbf{C}}(T)$, if $I^* \models a_\phi$, then $\mathcal{S} \models \phi$; from (1), if $I^* \models \neg a_\phi$, then there exists an interpretation I' such that $I' \models \neg\psi$ and for each $\psi \in Atom_{\mathbf{C}}(T)$, $I^* \models a_\psi$ implies $I' \models \psi$, thus $I' \in \mathcal{S}$ and $\mathcal{S} \not\models \phi$. Clearly, $(I, \mathcal{S}) \models T$.

Intuitively, the formula

$$\bigwedge_{\psi\in Atom_{\mathbf{C}}(T)} \left((\neg a_\psi \wedge \psi) \supset \left(\bigwedge_{\phi\in Atom_{\mathbf{C}}(T)} (a_\phi \supset \phi^\psi) \wedge \neg\psi^\psi \right) \right)$$

specifies that for each UCL structure $(I, \mathcal{S})$, if $I \models \psi$ and $\mathcal{S} \models \neg\mathbf{C}\psi$, then there exists an interpretation $I' \in \mathcal{S}$ such that $I' \models \neg\psi$.

In the following, we construct propositional formulas whose models are related to causally explained interpretations. First, we consider how to specify the unique model of a propositional formula.

Given a propositional formula ϕ and a nonempty consistent set K of literals, we denote by $\phi|_{K\to\bot}$ the result of replacing each occurrence of an atom a in ϕ by $\bot$ if $a \in K$ and $\top$ if $\neg a \in K$.

Lemma 1. *Let ϕ be a propositional formula, K a nonempty consistent set of literals, and an interpretation $I \supseteq K$. $I \not\models \bigwedge_{l\in K} l \supset \neg\phi|_{K\to\bot}$ if and only if the interpretation $(I \setminus K) \cup \overline{K} \models \phi$.*

Proof. Let $I' = (I \setminus K) \cup \overline{K}$.

"$\Rightarrow$" $I \not\models \bigwedge_{l \in K} l \supset \neg\phi|_{K\to\bot}$, then $I \models \phi|_{K\to\bot}$. Note that, atoms occurring in K do not occur in $\phi|_{K\to\bot}$, then $I' \models \phi|_{K\to\bot}$, furthermore, $I' \models \overline{K}$, thus $I' \models \phi$.

"$\Leftarrow$" $I' \models \phi$ and $I' \models \overline{K}$, then $I' \models \phi|_{K\to\bot}$, thus $I \models \phi|_{K\to\bot}$. Note that $K \subseteq I$, then $I \not\models \bigwedge_{l\in K} l \supset \neg\phi|_{K\to\bot}$.

To avoid influence of auxiliary atoms, we introduce the notion of forgetting provided by Lin and Reiter [14].

Definition 1. *Let ϕ be a propositional formula and S a set of atoms. $forget(\phi; S)$ is the formula inductively defined as follows:*

- $forget(\phi; \emptyset) = \phi$,
- $forget(\phi; \{a\}) = \phi|_{\{a\}\to\bot} \vee \phi|_{\{\neg a\}\to\bot}$,
- $forget(\phi; \{a\} \cup S) = forget(forget(\phi; S), \{a\})$.

Lemma 2 (Theorem 4 in [14]). *Let ϕ be a propositional formula and S a set of atoms. An interpretation $I \models forget(\phi; S)$ if and only if there exists an interpretation $I' \models \phi$ such that $I \setminus S \cup \overline{S} = I' \setminus S \cup \overline{S}$.*

Directly from Lemma 1 and 2, we have the following lemma.

Lemma 3. *Let ϕ be a propositional formula, K a nonempty consistent set of literals, S a set of atoms, and an interpretation $I \supseteq K$. $I \not\models \bigwedge_{l\in K} l \supset \neg forget(\phi; S)|_{K\to\bot}$ if and only if there exists an interpretation $I' \models \phi$ such that $\big((I \setminus S \cup \overline{S}) \setminus K\big) \cup \overline{K} = I' \setminus S \cup \overline{S}$.*

Given a propositional formula ϕ, we use $\widehat{\phi}$ to denote the propositional formula obtained from ϕ by replacing each occurrence of an atom a in ϕ by a new atom $\hat{a}$. For a set L of literals, we let $\widehat{L} = \{\hat{l} \mid l \in L\}$. We use Lit_a to denote the set of literals formed from new atoms of the form a_ϕ and $Atom^*$ the set of atoms of the form a^ψ w.r.t. ψ in (1).

Theorem 1. *Let T be a UCL theory. An interpretation I is causally explained by T if and only if there exists a model I^* of the propositional formula*

$$
\begin{aligned}
&(1) \wedge \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (a_\phi \supset \widehat{\phi}) \\
&\wedge \bigwedge_{\substack{A \subseteq Lit_a \\ A \text{ is nonempty and consistent}}} \left(\bigwedge_{l_a \in A} l_a \supset \neg\, forget\big((1); Atom^*\big)\Big|_{A\to\bot} \right) \\
&\wedge \bigwedge_{\substack{K \subseteq Lit \\ K \text{ is nonempty and consistent}}} \left(\bigwedge_{l \in K} \hat{l} \supset \neg \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (a_\phi \supset \widehat{\phi})\Big|_{\widehat{K}\to\bot} \right), \qquad (2)
\end{aligned}
$$

such that $I^* \cap Lit = I$.

Proof. "⇒" I is causally explained by T, then $(I, \{I\})$ is the unique I-model of T. We can create an interpretation I^* such that

$$I^* = I \cup \{a_\phi \mid \phi \in Atom_{\mathbf{C}}(T) \text{ and } I \models \phi\} \cup \{\neg a_\phi \mid \phi \in Atom_{\mathbf{C}}(T) \text{ and } I \not\models \phi\} \cup \bigcup_{\psi \in Atom_{\mathbf{C}}(T)} \{l^\psi \mid l \in I\} \cup \{\hat{l} \mid l \in I\}.$$

From Proposition 1, $I^* \models (1) \wedge \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (a_\phi \supset \widehat{\phi})$.

If $I^* \not\models \bigwedge_{l_a \in A} l_a \supset \neg\, forget\big((1); Atom^*\big)\big|_{A \to \bot}$ for some nonempty consistent set $A \subseteq Lit_a$, similar to the proof of Lemma 3, then these exists another interpretation $I^{*\prime}$ such that $((I^* \setminus Atom^* \cup \overline{Atom^*}) \setminus A) \cup \overline{A} = I^{*\prime} \setminus Atom^* \cup \overline{Atom^*}$ and $I^{*\prime} \models (1)$. From Proposition 1, there exists another set $\mathcal{S}'$ of interpretations such that $\mathcal{S}' \neq \{I\}$, $I \in \mathcal{S}'$ and $(I, \mathcal{S}')$ is an I-model of T, which conflicts to the condition that $(I, \{I\})$ is the unique I-model of T.

If $I^* \not\models \bigwedge_{l \in K} \hat{l} \supset \neg \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (a_\phi \supset \widehat{\phi})\Big|_{\widehat{K} \to \bot}$ for some nonempty consistent set $K \subseteq Lit$, similar to the proof of Lemma 1, then there exists another interpretation $I^{*\prime}$ such that $I^{*\prime} = (I^* \setminus \widehat{K}) \cup \overline{\widehat{K}}$ and $I^{*\prime} \models (1) \wedge \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (a_\phi \supset \widehat{\phi})$. From Proposition 1, there exists another interpretation I' such that $(I, \{I, I'\}) \models T$, which conflicts to the condition that $(I, \{I\})$ is the unique I-model of T, thus $I^* \models (2)$.

"⇐" $I^* \models (2)$. Let $I = I^* \cap Lit$, if there exists another UCL structure $(I, \mathcal{S})$ such that $(I, \mathcal{S}) \models T$ and $\mathcal{S} \neq \{I\}$, then there are two cases: 1. there exists $\phi \in Atom_{\mathbf{C}}(T)$ such that $I \models \phi$ and $\mathcal{S} \not\models \phi$; 2. for each $\phi \in Atom_{\mathbf{C}}(T)$, $I \models \phi$ if and only if $\mathcal{S} \models \phi$.

For case 1, let $A = \{a_\phi \mid \phi \in Atom_{\mathbf{C}}(T), I \models \phi, \mathcal{S} \not\models \phi\}$, then $I^* \models \bigwedge_{l_a \in A} l_a$ and $I^* \models forget\big((1); Atom^*\big)\big|_{A \to \bot}$, which conflicts to the condition that $I^* \models (2)$, thus it is impossible.

For case 2, let $I' \in \mathcal{S}$ and $I' \neq I$, then for each $\phi \in Atom_{\mathbf{C}}(T)$, $I^* \models a_\phi$ implies $I' \models \phi$, thus there exists $K = I \setminus I'$ such that $I^* \models \bigwedge_{l \in K} \hat{l}$ and $I^* \models \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (a_\phi \supset \widehat{\phi})\Big|_{\widehat{K} \to \bot}$, which conflicts to the condition that $I^* \models (2)$. So I is the only interpretation that satisfies $\{\phi \in Atom_{\mathbf{C}}(T) \mid I^* \models a_\phi\}$, then $(I, \{I\})$ is the unique I-model of T.

Note that, the size of formula (2) is exponential increased from T, as the number of all possible nonempty consistent sets of literals is 3^n, where n is the number of atoms. In fact, we only need to consider a subset of these sets. Details are proposed in Section 5.3.

As a simple example, given the UCL theory $T = \{\mathbf{C}(p \vee q),\ \mathbf{C}p \supset \mathbf{C}q,\ \mathbf{C}q \supset \mathbf{C}p\}$, from the definition of (1), we obtain the following propositional formula:

$$a_{p \vee q} \wedge (a_p \equiv a_q) \wedge (p \vee q) \wedge (a_p \supset p) \wedge (a_q \supset q) \wedge \big((\neg a_p \wedge p) \supset (\neg p^2 \wedge q^2)\big) \wedge \big((\neg a_q \wedge q) \supset (\neg q^3 \wedge p^3)\big) \quad (3)$$

From the definition of (2), we obtain the following formula[5]

$$
\begin{aligned}
(3) \wedge \big(a_{p\vee q} \supset (\hat{p} \vee \hat{q})\big) \wedge (a_p \supset \hat{p}) \wedge (a_q \supset \hat{q}) \\
\wedge \big(a_p \supset \neg(\neg a_q)\big) \wedge \big(a_q \supset \neg(\neg a_p)\big) \wedge \big((a_p \wedge a_q) \supset \bot\big) \\
\wedge \big(\neg a_p \supset \neg(a_q \wedge p \wedge q)\big) \wedge \big(\neg a_q \supset \neg(a_p \wedge p \wedge q)\big) \wedge \big(\neg a_p \wedge \neg a_q \supset \neg(p \wedge q)\big) \\
\wedge \big(\hat{p} \supset \neg((a_{p\vee q} \supset \hat{q}) \wedge \neg a_p \wedge (a_q \supset \hat{q}))\big) \wedge \big(\hat{q} \supset \neg((a_{p\vee q} \supset \hat{p}) \wedge (a_p \supset \hat{p}) \wedge \neg a_q)\big) \\
\wedge \big(\neg\hat{p} \supset \neg(a_q \supset \hat{q})\big) \wedge \big(\neg\hat{q} \supset \neg(a_p \supset \hat{p})\big)
\end{aligned}
$$

where $\hat{p}$ and $\hat{q}$ are new atoms. The formula implies that

$$a_{p\vee q} \wedge \hat{p} \wedge \hat{q} \wedge a_p \wedge a_q \wedge p \wedge q \wedge ((a_p \wedge a_q) \supset \bot)$$

which is inconsistent. From Theorem 1, there does not exist an interpretation I such that I is causally explained by the UCL theory T.

4 From Turner's Logic of Universal Causation to Logic Programming

Formula (2) in propositional logic is complex, as it needs to include constraints to make it satisfied by a "unique model". The problem becomes easier when we consider logic programming. Based on the propositional formula (1), we can translate a UCL theory T to a logic program with nested expressions.

Note that, every propositional formula ϕ can be equivalently translated to CNF as

$$(l_1^1 \vee \cdots \vee l_{n^1}^1) \wedge \cdots \wedge (l_1^m \vee \cdots \vee l_{n^m}^m), \tag{4}$$

where $l_1^1, \ldots, l_{n^m}^m$ are literals.

For any propositional formula, we can convert it to the nested expression by replacing each $\wedge$ with a comma, each $\vee$ with a semicolon and $\neg$ with *not*.

Given a UCL theory T, we use $tr_{ne}(T)$ to denote the nested expression obtained from (1). We use $Atom'$ to denote the set of atoms that occur in (1) but not in $Atom$. Now we define $tr_{lp}(T)$ to be the logic program containing $\bot \leftarrow not\, tr_{ne}(T)$, the following rules for each $\phi \in Atom_{\mathbf{C}}(T)$ whose CNF is in the form of (4)

$$
\begin{aligned}
l_1^1; \ldots; l_{n^1}^1 &\leftarrow not\, not\, a_\phi, (\bar{l}_1^1; not\, \bar{l}_1^1), \ldots, (\bar{l}_{n^1}^1; not\, \bar{l}_{n^1}^1), \\
&\cdots \\
l_1^m; \ldots; l_{n^m}^m &\leftarrow not\, not\, a_\phi, (\bar{l}_1^m; not\, \bar{l}_1^m), \ldots, (\bar{l}_{n^m}^m; not\, \bar{l}_{n^m}^m),
\end{aligned}
$$

and

$$a'; \neg a' \leftarrow \top, \qquad \text{(for each } a' \in Atom'\text{)}.$$

[5] The formula is simplified due to Theorem 5 in Section 5.3.

Lemma 4. *Let T be a UCL theory and I and J two interpretations. $(I, \{I, J\}) \models T$ if and only if there exists a set S of literals occurring in $tr_{lp}(T)$ such that $S \models (tr_{lp}(T))^{S \cup I}$ and $S \cap Lit = I \cap J$.*

Proof. "⇒" $(I, \{I, J\}) \models T$. Similar to the proof of Proposition 1, we can create an interpretation I^* such that $I^* \cap Lit = I$ and $I^* \models (1)$. Note that, $(tr_{lp}(T))^{I^*}$ contains rules of the form

$$l_1; \ldots; l_n \leftarrow \mathop{,}_{l \in \{l_1, \ldots, l_n\}, \bar{l} \in I} \bar{l}, \tag{5}$$

where $I^* \models a_\phi$ for corresponding $\phi \in Atom_{\mathbf{C}}(T)$.

Note that, $\{I, J\} \models \phi$, $I \models l_1 \vee \cdots \vee l_n$ and $J \models l_1 \vee \cdots \vee l_n$. Consider the case, for each literal $l \in \{l_1, \ldots, l_n\}$, $\bar{l} \in I$ implies $\bar{l} \in I \cap J$, then there exists literal $l \in \{l_1, \ldots, l_n\}$ and $l \in J$ such that $l \in I$ (if not, $\bar{l} \in I$ which implies $\bar{l} \in J$), thus $(I \cap J) \models (5)$.

We denote $S = (I^* \setminus I) \cup (I \cap J)$. Clearly, $S \models (tr_{lp}(T))^{S \cup I}$.

"⇐" $S \models (tr_{lp}(T))^{S \cup I}$ and $S \cap Lit = I \cap J$. $(tr_{lp}(T))^{S \cup I}$ contains rules of (5) and $S \models a_\phi$ for corresponding $\phi \in Atom_{\mathbf{C}}(T)$.

Note that $I \cap J \models (5)$, then $I \cap J \models l_1 \vee \cdots \vee l_n$ whenever $\bar{l} \in I \cap J$ for all $\bar{l} \in I$ and $l \in \{l_1, \ldots, l_n\}$. If $J \not\models l_1 \vee \cdots \vee l_n$, then there exists $\bar{l} \in I$ and $\bar{l} \notin I \cap J$, thus $\bar{l} \notin J$ and $l \in J$ which conflicts to $J \not\models l_1 \vee \cdots \vee l_n$. So $J \models \phi$, from Proposition 1, $(I, \{I, J\}) \models T$.

Theorem 2. *Let T be a UCL theory. An interpretation I is causally explained by T if and only if there exists an answer set S of the logic program $tr_{lp}(T) \cup \{\bot \leftarrow not\, a, not\, \neg a \mid a \in Atom\}$, such that $S \cap Lit = I$.*

Proof. I is causally explained by T means that $(I, \{I\})$ is the unique I-model of T. From Lemma 4, this is equivalent to the condition, for every set S of literals occurring in $tr_{lp}(T)$ and interpretation J such that $S \models (tr_{lp}(T))^{S \cup I}$, $S \cap Lit = I \cap J$ iff $J = I$. This means that there exists an answer set S of $tr_{lp}(T) \cup \{\bot \leftarrow not\, a, not\, \neg a \mid a \in Atom\}$ such that $S \cap Lit = I$.

5 Related Work

5.1 Turner's Conversion from a Fragment of UCL to Disjunctive Logic Programming

Turner [17] proposed a simple translation from a subset of UCL theories to disjunctive logic programs [3] via disjunctive default logic [4].

Turner's translation considers the UCL formula of the form

$$\mathbf{C}(l_1 \wedge \cdots \wedge l_k) \wedge l_{k+1} \wedge \cdots \wedge l_m \supset \mathbf{C}l_{m+1} \vee \cdots \mathbf{C}l_n, \tag{6}$$

where $l_1, \ldots, l_n$ are literals.

A UCL formula of the form (6) is translated to the disjunctive rule

$$l_{m+1} \vee \cdots \vee l_n \leftarrow l_1, \ldots, l_k, not\, \bar{l}_{k+1}, \ldots, not\, \bar{l}_m.$$

It has been proved that, given a set T of UCL formulas in the form (6), an interpretation I is an answer set of the corresponding disjunctive logic program if and only if I is causally explained by T.

When every formula in range of $\mathbf{C}$ is a literal, our translation seems more complex than Turner's translation. However, some steps in the translation can also be simplified. Consider the following proposition proposed in [2].

Proposition 2 (Proposition 1 in [2]). *For any literal l and any nested expression F, the one-rule logic program*

$$l \leftarrow F, (\bar{l}; not\, \bar{l})$$

is strongly equivalent to $l \leftarrow F$.

5.2 Ferraris's Translation from Causal Theories to Logic Programs

Ferraris [2] proposed a translation from McCain and Turner's causal theories [15] to logic programs with nested expressions. As causal theories can be easily converted into UCL, we show that Ferraris's translation is a special case of our translation proposed in Section 4. First, we briefly review causal theories and Ferraris's translation, then we consider the relation to our translation.

A causal theory according to McCain and Turner [15] is a set of *causal laws* of the following form

$$\psi \Rightarrow \phi, \tag{7}$$

where ϕ and ψ are propositional formulas.

Ferraris's translation converts the causal law

$$\psi \Rightarrow l_1 \vee \cdots \vee l_n, \tag{8}$$

to the rule

$$l_1; \ldots ; l_n \leftarrow not\, not\, \psi^{ne}, (\bar{l}_1; not\, \bar{l}_1), \ldots, (\bar{l}_n; not\, \bar{l}_n),$$

where ψ^{ne} stands for the nested expression of ψ. Theorem 1 in [2] proved that models of a set of causal laws in the form (8) are identical to complete answer sets of the corresponding logic programs.

According to Turner [17], a causal law of the form (7) can be translated to his logic as

$$\psi \supset \mathbf{C}\phi. \tag{9}$$

Thus given our translation from Turner's logic to logic programming, we have a translation from McCain and Turner's causal theory to logic programming as well.

A UCL formula of the form (9) is called *regular*. A *regular* UCL theory is a set of regular UCL formulas.

Note that, when T is a regular UCL theory, formula (1) in Proposition 1 can be simplified to

$$\bigwedge_{F \in T} tr_p(F) \wedge \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (a_\phi \supset \phi). \tag{10}$$

Proposition 3. *Let T be a regular UCL theory. A UCL structure $(I, \mathcal{S})$ is a model of T if and only if there exists a model I^* of formula* (10) *such that $I^* \cap Lit = I$ and for each $\phi \in Atom_{\mathbf{C}}(T)$, $a_\phi \in I^*$ iff $\mathcal{S} \models \phi$.*

Based on Proposition 3, the translation in Section 4 can also be simplified. Given a regular UCL theory T, we use $tr'_{ne}(T)$ to denote the nested expression obtained from (10). We define $tr'_{lp}(T)$ the same as $tr_{lp}(T)$ except $tr_{ne}(T)$ is replaced by $tr'_{ne}(T)$.

Theorem 3. *Let T be a regular UCL theory. An interpretation I is causally explained by T if and only if there exists an answer set S of the logic program $tr'_{lp}(T) \cup \{\bot \leftarrow not\, a, not\, \neg a \mid a \in Atom\}$, such that $S \cap Lit = I$.*

It is easy to find out that, for regular UCL theory T, $tr'_{lp}(T)$ is equivalent to the result of Ferraris's translation.

Our translation in Section 4 can also be specified by Ferraris's translation. First, a UCL theory can be converted to a regular UCL theory.

Theorem 4. *Let T be a UCL theory. An interpretation I is causally explained by T if and only if I is causally explained by the regular UCL theory with following formulas*

$$
\begin{array}{ll}
(1), & \\
a_\phi \supset \mathbf{C}\phi, & (\textit{for each } \phi \in Atom_{\mathbf{C}}(T)) \\
a' \supset \mathbf{C}a',\ \neg a' \supset \mathbf{C}\neg a'. & (\textit{for each } a' \in Atom')
\end{array}
$$

Then we can use Ferraris's translation turning the regular UCL theory in the above theorem to a logic program with nested expressions.

5.3 Lee's Translation from Causal Theories to Propositional Theories with Loop Formulas

Lee [9] proposed a translation from McCain and Turner's causal theories to propositional theories with loop formulas. In this section, we show that we can also define so called "loop formulas" for our translation in Section 3 and Lee's translation would be a special case.

Given a set Π of propositional clauses, i.e. disjunctions of literals, the *dependency graph* of Π is the directed graph G_Π such that

- the vertices of G_Π are literals in Π, and
- for any two vertices l_1, l_2, there is an edge from l_1 to l_2 if there is a clause $C \in \Pi$ such that l_1 and $\overline{l_2}$ are in C.

A nonempty consistent set L of literals is called a *loop* of Π if for any literals l_1 and l_2 in L, there is a path from l_1 to l_2 in G_Π such that all the vertices in the path are in L, i.e. the L-induced subgraph of G_Π is strongly connected. Specially, the singleton set $\{l\}$ for every literal $l \in Lit$ is a loop. We use $Loop(\Pi)$ to denote the set of all loops of Π.

The *loop formula* associated with a loop L under a set Π of propositional clauses, denoted by $LF(\Pi, L)$, is a sentence of the form:

$$\bigwedge_{l \in L} l \supset \neg \bigwedge_{C \in \Pi} C|_{L \to \bot} .$$

We can simplify the translation from UCL to propositional logic by loops.

Proposition 4. *Let Π be a set of propositional clauses,*

$$\bigwedge_{C \in \Pi} C \wedge \bigwedge_{L \in Loop(\Pi)} LF(\Pi, L) \supset \bigwedge_{C \in \Pi} C \wedge \bigwedge_{\substack{K \subseteq Lit \\ K \text{ is nonempty} \\ \text{and consistent}}} \left(\bigwedge_{l \in K} l \supset \neg \bigwedge_{C \in \Pi} C|_{K \to \bot} \right).$$

Proof. Let $L \in Loop(\Pi)$, K a nonempty consistent set of literals s.t. $L \subseteq K$, and there does not exist an edge of G_Π from a literal in L to a literal in $K \setminus L$.

There does not exist an edge of G_Π from a literal in L to a literal in $K \setminus L$, then there does not exist a clause C in Π of the form

$$l_1 \vee \cdots \vee l_n$$

such that $l_i \in L$ and $l_j \in K \setminus L$ for some $1 \leq i, j \leq n$. Thus, if $C \in \Pi$ and $C \cap L \neq \emptyset$, then $C \cap (K \setminus L) = \emptyset$.

For each clause $C \in \Pi$, as $L \subseteq K$, there are three different cases.

Case 1, $L \cap C = \emptyset$. If $L \cap \overline{C} \neq \emptyset$, then $K \cap \overline{C} \neq \emptyset$, thus $\neg C|_{L \to \bot} \equiv \neg C|_{K \to \bot} \equiv \bot$. If $L \cap \overline{C} = \emptyset$, then $\neg C|_{L \to \bot} \equiv \neg C$, thus $C \wedge LF(\{C\}, L) \supset C \wedge (\bigwedge_{l \in K} l \supset \neg C|_{K \to \bot})$.

Case 2, $L \cap C \neq \emptyset$, $L \cap \overline{C} = \emptyset$, and $K \cap \overline{C} = \emptyset$. From the above condition, $C \cap (K \setminus L) = \emptyset$, then $\neg C|_{L \to \bot} \equiv \neg C|_{K \to \bot}$.

Case 3, $K \cap \overline{C} \neq \emptyset$. Then $\neg C|_{K \to \bot} \equiv \bot$, thus $C \wedge LF(\{C\}, L) \supset C \wedge (\bigwedge_{l \in K} l \supset \neg C|_{K \to \bot})$.

Based on the above results,

$$\bigwedge_{C \in \Pi} C \wedge LF(\Pi, L) \supset \bigwedge_{C \in \Pi} C \wedge \left(\bigwedge_{l \in K} l \supset \neg \bigwedge_{C \in \Pi} C|_{K \to \bot} \right).$$

In addition, for every nonempty consistent set K of literals, there always exists a loop $L \subseteq K$ such that there does not exist an edge of G_Π from a formula in L to a formula in $K \setminus L$. So the proposition is proved.

Given a UCL theory T, with a slight abuse of notations, we use $Loop(T)$ to the set of loops of the set of clauses which are in CNF of $\phi \in Atom_{\mathbf{C}}(T)$. Similarly, we use $Loop_a(T)$ to the set of loops of the set of clauses which are in CNF of (1).

Theorem 5. *Let T be a UCL theory. An interpretation I is causally explained by T if and only if there exists a model I^* of the propositional formula*

$$
\begin{aligned}
&(1) \wedge \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (\, a_\phi \supset \widehat{\phi}\,) \\
&\quad \wedge \bigwedge_{A \subseteq Lit_a,\, A \in Loop_a(T)} \left(\bigwedge_{l_a \in A} l_a \supset \neg\, forget\big((1); Atom^*\big)\big|_{A \to \bot} \right) \\
&\qquad \wedge \bigwedge_{L \in Loop(T)} \left(\bigwedge_{l \in L} \hat{l} \supset \neg \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (a_\phi \supset \widehat{\phi})\Big|_{\widehat{L} \to \bot} \right), \quad (11)
\end{aligned}
$$

such that $I^* \cap Lit = I$.

Similar to the discussion in the previous section, when T is a regular UCL theory, the above theorem can be simplified.

Theorem 6. *Let T be a regular UCL theory. An interpretation I is causally explained by T if and only if there exists a model I^* of the propositional formula*

$$
\begin{aligned}
&\bigwedge_{F \in T} tr_p(F) \wedge \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (\, a_\phi \supset \phi\,) \wedge \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (\, a_\phi \supset \widehat{\phi}\,) \\
&\qquad \wedge \bigwedge_{L \in Loop(T)} \left(\bigwedge_{l \in L} \hat{l} \supset \neg \bigwedge_{\phi \in Atom_{\mathbf{C}}(T)} (a_\phi \supset \widehat{\phi})\Big|_{\widehat{L} \to \bot} \right), \quad (12)
\end{aligned}
$$

such that $I^* \cap Lit = I$.

When each formula in the range of $\mathbf{C}$ is a clause in the regular UCL theory T, comparing the above theorem with Theorem 1 in [9], it is easy to find out that formula (12) corresponds to $DR(T) \cup CLC(T)$ in Lee's Theorem.

6 Conclusion

We have provided translations from Turner's logic of universal causation to propositional logic and logic programming. These translations generalize the respective translations by Ferraris and Lee for McCain and Turner's causal theories. Our next step is to use these results to implement Turner's logic using SAT and ASP solvers.

It is worth mentioning here that our results in this paper can also be used to map Turner's logic to fixed-point nonmonotonic logics such as default logic [16] and Lin and Shoham's logic of GK [13,8].

Acknowledgments. This work had been supported by the National Hi-Tech Project of China under grant 2008AA01Z150, the Natural Science Foundation of China under grant 60745002 and 61175057, the USTC Key Direction Project, the Fundamental Research Funds for the Central Universities, the Youth Innovation Fund of USTC, and HK RGC GRF 616909 . We thank the anonymous reviewers for their valuable comments on an earlier version of the paper.

References

1. Drescher, C., Gebser, M., Grote, T., Kaufmann, B., König, A., Ostrowski, M., Schaub, T.: Conflict-Driven Disjunctive Answer Set Solving. In: Brewka, G., Lang, J. (eds.) Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR-08). pp. 422–432. AAAI Press, Menlo Park, CA, USA (2008)
2. Ferraris, P.: A logic program characterization of causal theories. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07). pp. 366–371 (2007)
3. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Comput. 9(3/4), 365–386 (1991)
4. Gelfond, M., Lifschitz, V., Przymusińska, H., Truszczyński, M.: Disjunctive Defaults. In: Allen, J.F., Fikes, R., Sandewall, E. (eds.) Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR-91). pp. 230–237. Morgan Kaufmann, San Fransisco, CA, USA (1991)
5. Giunchiglia, E., Lee, J., Lifschitz, V., McCain, N., Turner, H.: Nonmonotonic causal theories. Artificial Intelligence 153(1-2), 49–104 (2004)
6. Giunchiglia, E., Lierler, Y., Maratea, M.: Answer Set Programming Based on Propositional Satisfiability. J. Autom. Reasoning 36(4), 345–377 (2006)
7. Janhunen, T., Niemelä, I.: GnT – A Solver for Disjunctive Logic Programs. In: Lifschitz, V., Niemelä, I. (eds.) Proceedings of the 7th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-04). pp. 331–335. Springer-Verlag, Berlin, Heidelberg (2004), lNAI 2923
8. Ji, J., Lin, F.: From Turner's Logic of Universal Causation to the Logic of GK. In: Erdem, E., Lee, J., Lierler, Y., Pearce, D. (eds.) Correct Reasoning. Lecture Notes in Computer Science, vol. 7265, pp. 380–385. Springer (2012)
9. Lee, J.: Nondefinite vs. definite causal theories. In: Proceedings of the 7th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-04). pp. 141–153 (2004)
10. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. ACM Transactions on Computational Logic 7(3), 499–562 (2006)
11. Lifschitz, V.: Action languages, answer sets and planning. In: The Logic Programming Paradigm: a 25-Year Perspective, pp. 357–373. Springer Verlag (1999)
12. Lifschitz, V., Tang, L.R., Turner, H.: Nested expressions in logic programs. Annals of Mathematics and Artificial Intelligence 25(3-4), 369–389 (1999)
13. Lin, F., Shoham, Y.: A logic of knowledge and justified assumptions. Artificial Intelligence 57(2-3), 271–289 (1992)
14. Lin, F., Reiter, R.: Forget it. In: Working Notes of AAAI Fall Symposium on Relevance. pp. 154–159 (1994)
15. McCain, N., Turner, H.: Causal theories of action and change. In: Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97). pp. 460–465 (1997)
16. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13, 81–132 (1980)
17. Turner, H.: Logic of universal causation. Artificial Intelligence 113(1), 87–123 (1999)