An Abductive Counterfactual Reasoning Approach in Logic Programming

Luís Moniz Pereira\textsuperscript{1}\textsuperscript{*}
Emmanuelle-Anna Dietz\textsuperscript{1,2}\textsuperscript{†}
Steffen Hölldobler\textsuperscript{2}\textsuperscript{†}

\textsuperscript{1}NOVA Laboratory for Computer Science and Informatics (NOVA LINCS), Universidade Nova de Lisboa, Portugal
\textsuperscript{2}International Center for Computational Logic TU Dresden, Germany

Abstract
We construct a counterfactual statement when we reason conjecturally about an event which did or did not occur in the past: If an event had occurred, what would have happened? Would it be relevant? Real world examples, as studied by Byrne, Rescher and many others, show that these conditionals involve a complex reasoning process. An intuitive and elegant approach to evaluate counterfactuals, without deep revision mechanisms, is proposed by Pearl. His \textit{Do}-Calculus identifies causal relations in a Bayesian network resorting to counterfactuals. Though leaving out probabilities, we adopt Pearl’s stance, and its prior epistemological justification to counterfactuals in causal Bayesian networks, but for programs. Logic programming seems a suitable environment for several reasons. First, its inferential arrow is adept at expressing causal direction and conditional reasoning. Secondly, together with its other functionalities such as abduction, integrity constraints, revision, updating and debugging (a form of counterfactual reasoning), it proffers a wide range of expressibility itself. We show here how programs under the weak completion semantics in an abductive framework, comprising the integrity constraints, can smoothly and uniformly capture well-known and off-the-shelf counterfactual problems and conundrums, taken from the psychological and philosophical literature. Our approach is adroitly reconstructable in other three-valued LP semantics, or restricted to two-valued ones.

Keywords: Counterfactual Reasoning, Pearl’s Do Calculus, Logic Programming, Abduction, Weak Completion Semantics.

\textsuperscript{*}lmp@fct.unl.pt
\textsuperscript{†}\{dietz,sh\}@iccl.tu-dresden.de
1 Introduction

Counterfactual and causal reasoning has been widely studied in linguistics, in psychology as well as in philosophy, and in the logic programming (LP) field [1–4]. One of the first elaborate analysis was carried out by Lewis, who employed a possible world semantics for counterfactuals [5]. Counterfactuals capture the process of reasoning about a past event that did not occur, namely what would have happened, had this event occurred, or, vice-versa, to reason about an event that did occur but what if it had not. Consider the example from Byrne in [6, pp. 107 – 108]:

Lightning hits a forest and a devastating forest fire breaks out.
The forest was dry after a long hot summer and many acres were destroyed.

Let us express a possible causal relation by the following conditional:

\[
\text{If there is a lightning and the leaves are dry, then there is a forest fire.} \quad (C_1)
\]

A counterfactual we might think of could be as follows:

\[
\text{If only there had not been so many dry leaves on the forest floor, then the forest fire wouldn’t have occurred.} \quad (\text{CF}_{\text{dry,fire}})
\]

Similarly to [7], we extend this scenario with another possible reason for a forest fire, represented by the following conditional:

\[
\text{If there is a fire-raising\textsuperscript{1}, then there is a forest fire.} \quad (C_2)
\]

Note that, different to \(C_1\), for a fire-raising to be successful, i.e., to cause a forest fire, it is not necessary that the leaves be dry. We assume an arsonist (the one who raises the fire) would use some aggressive substances to make sure that the fire spreads independently of conditions on the forest ground.

If counterfactuals are understood as usual material conditionals, then they are trivially true, as the antecedents are patently false. In order to evaluate a counterfactual adequately, it is necessary to go ‘a step back’ and assume that the antecedent is actually true. For instance, \(\text{CF}_{\text{dry,fire}}\) implies that the leaves were dry, thus for the evaluation of this counterfactual, we need to assume that the leaves were conjecturally not dry. Suppose this assumption together with the information that there is a lightning and that we only know about \(\text{CF}_{\text{dry,fire}}\). Then, when evaluating \(\text{CF}_{\text{dry,fire}}\) we would preferably come to the conclusion that the forest fire would not occur, and thus, that \(\text{CF}_{\text{dry,fire}}\) is valid. However, if additionally to \(\text{CF}_{\text{dry,fire}}\) we

\textsuperscript{1}The act of intentionally burning something
also know about $C_2$, we do not want to conclude that $\text{CF}_{\text{dry,fire}}$ is valid anymore. In this case, a forest fire still could have occurred because of the fire-raising.

At first glance, it seems that counterfactual reasoning requires some involved belief-revision procedure; however, there might be a more convenient approach. In [8], Pearl presents a theory for employing counterfactuals including conjectures and Bayesian networks, extensively spelled out and exemplified in [9]. His main idea is to accept a counterfactual if its consequent is true after adding the antecedent hypothetically to the beliefs and making the minimal required adjustments to maintain consistency of the model. This is achieved by isolating the antecedent node from its parent nodes in the network whilst forcefully imposing it to be true, and subsequently computing the corresponding network model to evaluate whether the consequent then follows. Probabilistic counterfactuals in Bayesian Networks, simulating Pearl’s approach, are captured in LP by Baral et al. with their system P-log [10][11]. P-log has been used in probabilistic moral reasoning [12] and the authors intend to employ counterfactuals in moral reasoning, as part of ongoing work using LP [13][14].

In the sequel, before our formal preliminaries in Section 3, we discuss related work. In Section 4 we present the main contribution of this paper, a counterfactual abductive framework employing logic programming. We illustrate this approach with examples and discuss its formal properties.

2 Related Work

There are three main prototypical alternatives to counterfactual analysis: Ramsey’s maximal belief-retention approach [15]; Lewis’s maximal world-similarity one [5]; and Rescher’s systematic reconstruction of the belief system, using principles of saliency and prioritization [16].

Different from our LP least weak completion model approach, and its revision of logic rules negating the counterfactual premise, Ginsberg [2] employs a possible worlds approach to evaluate counterfactuals, defining the closest worlds as those obtained by minimally removing logic clauses such that no contradiction is obtained when enforcing the counterfactual premise. Pereira and Aparício [17] improve on Ginsberg’s approach by imposing the requisite of relevance of the counterfactual premise for its consequent. They also addressed the irrelevance issue in the treatment of even-if counterfactuals. For a belief revision characterization of counterfactuals in LP through a possible worlds stance see [1]. [14][18] inspired by our work, have employed and implemented a well-founded semantics approach to LP counterfactual reasoning with applications to morality.
2.1 Pearl’s Do-Calculus

Pearl [8] proposes a structural theory of counterfactuals in Bayesian networks which determines the probability of a counterfactual. We briefly sketch the main idea of Pearl’s well-known theory: Assume a structural model $M$ that consists of two sets of variables $U$ and $V$, and a set of functions $F$ that decides how values are assigned to each variable $V_i \in V$. The variables in $U$ are exogenous, that means they are some given background knowledge for which there is no explanatory mechanism, i.e. parent nodes, encoded in the model. The probability function of every (endogenous) variable in $V$ is uniquely determined by the instantiated background variable $U = u$. Let us consider the following statement:

*Given $e$, what is the probability that $Y$ had happened, had we done $X$?*

The probability of the counterfactual sentence $Y$ would be $y$ had $X$ been $x$ in situation $U = u$, can be computed in a three step process: Let us assume that $P(u)$ denotes the probability of $u$ and $e$ is any current evidence:

**Step 1 (abduction)** Update the probability of $P(u)$ to obtain $P(u | e)$.

**Step 2 (action)** Replace equations corresponding to variables in $X$ by $X = x$.

**Step 3 (prediction)** Compute the probability of $Y = y$ in the modified model.

Step 1 updates the past circumstances ($U$) with the additional current evidence $e$; step 2 changes the past sufficiently enough by the hypothetical condition $X = x$ and step 3 predicts the future ($Y$) with the modifications done in step 2, while in keeping with the newly determined $U$ context. In a nutshell, intervene to impose $x$ and determine probability of $y$, other things being equal given evidence $e$ about $u$.

2.2 Rescher’s Systematic Reconstruction of Belief

Rescher’s semantically pragmatic approach does not put unrealizable demands on reasoning - like surveying whole possible worlds, recasting entire belief systems - but, to the contrary, only requires scrutinizing immediately relevant beliefs. Likewise, the crux of our counterfactual analysis is not an issue of scrutinizing the situation at hand at other possible worlds, or of reformulating a whole web of beliefs, but rather of comparatively prioritizing the present relevant epistemic beliefs regarding the actual world and incidental to the case at hand. Rescher discusses the weakest link principle, whose goal is to restore consistency by breaking the chain of inconsistency at its weakest link(s) [19, p. 99]. In our LP context, this corresponds to that we aim at just a ‘counterfactualized’ clauses or surface revision, and not at a deep
clausal revision (revising clausal subgoals) that puts into question clausal knowledge, inasmuch it can involve more side consequences. We suppose people normally do just that, as deep counterfactuals are unwieldy, costly, and non-deterministic.

2.3 CP-Logic

In [20,21], the authors show how Pearl’s intervention can be represented in CP-Logic. CP-Logic is a logic of causal Probabilistic Events. Their logic programs contain causal probabilistic laws which state the cause and possible effects of a particular event or class of events. They have the following form: \( \forall x (A_1 : \alpha_1) \lor \cdots \lor (A_i : \alpha_i) \leftarrow \delta \)
where \( \delta \) is a first-order formula and \( A_i \) are atoms, the \( \alpha_i \) are non-zero probabilities, and the tuple of variable \( z \) contains free variablies in \( \delta \) and the \( A_i \). They can be read as follows “for each \( x \), \( \delta \) causes an event whose effect is that at most one of the \( A_i \) becomes true; for each \( i \), the probability of \( A_i \) being the effect of this event is \( \alpha_i \).” [20] Their semantics is based on Shafer’s probability trees [22], where each node corresponds to an interpretation for a given vocabulary. Each node in the tree is a state where the parents represent an event that causess a probabilistic transition to one of its children, which is represented by a probability distribution \( \pi(T)(l) \) for each leave \( l \). Following Pearl, they define an intervention as a pair \((R, A)\) where \( R \) is a subset of a set of CP-laws \( C \) and \( A \) is a set of CP-laws, not in \( C \). The result of performing \((R, A)\) on \( C \), is the CP-theory \( (C \min R) \cup A \). Accordingly, if they intend to block an effect, they simply exclude the CP-law in the new set and when they intend to force the outcome of an event, they can impose a CP-law by adding it to the set without specifying the probability (that is, with probability 1). Different than in our approach, they do not encounter the issue of conflicting laws that might overwrite other ones, because first they the imposed CP-law has a higher probability as all possibly conflicting ones and second, they exclude the old CP-laws from the new program.

3 Preliminaries

In this section we introduce the general notation and terminology that will be used throughout the paper, based on [23,24].

3.1 Logic Programs

We restrict ourselves to datalog programs, i.e. the set of terms consists only of constants and variables. A logic program \( P \) is a finite set of clauses of the form

\[
A \leftarrow L_1 \land \ldots \land L_n,
\]
where $n \geq 0$ with finite $n$. $A$ is an atom and $L_i$, $1 \leq i \leq n$, are literals. $A$ is called head of the clause and the subformula to the right of the implication sign is called body of the clause. If a clause only contains atoms in the body, then it is definite. If a program contains only definite clauses, then it is a definite program. If the clause contains variables, then they are implicitly universally quantified within the scope of the entire clause. A clause that does not contain variables, is called a ground clause. In case $n = 0$, then the clause is a positive fact and denoted as

$$A \leftarrow \top.$$  

A negative fact is denoted by

$$A \leftarrow \bot,$$

where true, $\top$, and false, $\bot$, are truth-value constants. The notion of falsehood appears counterintuitive at first sight, but programs will be interpreted under their (weak) completion where we replace the implication by the equivalence sign. In the sequel, we assume $P$ to be ground, containing all the ground instances of its clauses. To refer to the positive and negative part of a body, we introduce the following notation: If $F$ is a conjunction of literals, then $\text{pos}(F)$ (resp.) denotes the conjunction of all positive (resp.) literals occurring in $F$. An empty conjunction is always true, therefore, if $F$ does not contain any literal, $\text{pos}(F) = \text{neg}(F) = \top$. According to this, it holds that $\text{pos}(\top) = \text{neg}(\top) = \top$. To let pos and neg also be applicable to bodies of negative facts, we define additionally $\text{pos}(\bot) := \text{neg}(\bot) := \top$.

If $P$ is a program, then $\text{atoms}(P)$ denotes the set of all atoms occurring in $P$. The set of all clauses with head $A$ in $P$ is called the definition of $A$ in $P$. If this set is nonempty, the atom $A$ is said to be defined in $P$, otherwise $A$ is said to be undefined in $P$. The set of all atoms that are defined in $P$ is denoted by $\text{def}(P)$. The set of all atoms that are undefined in $P$, that is, $\text{atoms}(P) \setminus \text{def}(P)$, is denoted by $\text{undef}(P)$. Consider $P$:

$$p \leftarrow q \land \overline{r} \land s, \quad q \leftarrow t, \quad r \leftarrow \bot, \quad s \leftarrow \top,$$

where the third clause is a negative and the fourth clause is a positive fact. Applying pos and neg to the body of the first clause gives the following result:

$$\text{pos}(q \land \overline{r} \land s) = q \land s, \quad \text{neg}(q \land \overline{r} \land s) = \overline{r}.$$

The sets of atoms, the set of defined and the set of undefined atoms are:

$$\text{atoms}(P) = \{p, q, r, s, t\}, \quad \text{def}(P) = \{p, q, r, s\}, \quad \text{undef}(P) = \{t\}.$$
Let $\mathcal{P}$ be a program and $p, q \in \text{atoms}(\mathcal{P})$. $p$ depends negatively on $q$ wrt $\mathcal{P}$ iff $\mathcal{P}$ contains a clause of the form $p \leftarrow \text{Body}$ and $q$ is in $\text{neg}(\text{Body})$. $p$ depends positively on $q$ wrt $\mathcal{P}$ iff $p$ does not depend negatively on $q$ and $\mathcal{P}$ contains a clause of the form $p \leftarrow \text{Body}$ and $q$ is in $\text{pos}(\text{Body})$. $p$ depends on $q$ wrt $\mathcal{P}$ iff $p$ depends positively or negatively on $q$ wrt $\mathcal{P}$. Additionally, dependency is transitive, thus, if $p$ depends on $q$ and $q$ depends on $t$, then $p$ depends on $t$. One negative dependency is enough to define the whole dependency as negative.

### 3.2 Three-Valued Łukasiewicz Semantics

Logic programs are normally evaluated under two-valued semantics, where a two-valued interpretation $I$ of $\mathcal{P}$ is a mapping of the Herbrand base $\mathcal{B}_P$ to $\{\top, \bot\}$. A two-valued model $\mathcal{M}$ for $\mathcal{P}$ is a two-valued interpretation which maps each clause occurring in $\mathcal{P}$ to $\top$.

We extend two-valued semantics to three-valued Łukasiewicz Semantics [25], for which the corresponding truth values are $\top$, $\bot$ and $\mathsf{U}$, which mean true, false and unknown, respectively. A three-valued interpretation $I$ is a mapping from formulas to a set of truth values $\{\top, \bot, \mathsf{U}\}$. The truth value of a given formula under $I$ is determined according to the truth tables in Table 1. We represent an interpretation as a pair $I = \langle I^\top, I^\bot \rangle$ of disjoint sets of atoms where $I^\top$ is the set of all atoms that are mapped to $\top$ by $I$, and $I^\bot$ is the set of all atoms that are mapped to $\bot$ by $I$. Atoms which do not occur in $I^\top \cup I^\bot$, are mapped to $\mathsf{U}$. $\mathcal{I}(F) = \top$ means that a formula $F$ is mapped to true under $I$. $\mathcal{M}$ is a three-valued model of $\mathcal{P}$ if it is a three-valued interpretation, which maps each clause occurring in $\mathcal{P}$ to $\top$.

In the sequel, we implicitly assume all interpretations and models to be three-valued. If we mean two-valued interpretations and two-valued models, we explicitly write it.
3.3 Reasoning with Respect to Least Models

Least models can often be computed as least fixed points of an appropriate semantic operator \[26\]. For instance, the least fixed point of the \(T_P\) operator (lfp \(T_P\)) corresponds to the least two-valued model of a definite program \(P\). Given a definite program \(P\) and an atom \(A\), \(P \models A\) iff \(A \in \text{lfp} T_P\). For any definite program there exists a least model. However this does not hold for programs that are not definite. Hölldobler and Kencana Ramli \[27\] proposed an alternative approach for all programs, the weak completion semantics which extends the two-valued semantics to three-valued Łukasiewicz semantics, and which guarantees a least fixed point for every program. It seems to adequately model some famous human reasoning tasks from cognitive science \[28–30\]. In contrast to well-founded semantics, this approach seems to be easier computable and understandable by people, and its treatment of positive loops appears more in line with psychological experiments \[31\]. Consider following transformation for \(P\):

1. Replace all clauses in \(P\) with the same head \(A \leftarrow \text{Body}_1, A \leftarrow \text{Body}_2, \ldots\) by the single expression \(A \leftarrow \text{Body}_1 \lor \text{Body}_2, \lor \ldots\).

2. If \(A\) is undefined in \(P\), then add \(A \leftarrow \bot\).

3. Replace all occurrences of \(\leftarrow\) by \(\leftrightarrow\).

The resulting set of equivalences is called the completion of \(P\) \[32\]. If Step 2 is omitted, then the resulting set is called the weak completion of \(P\) (wc \(P\)). In contrast to completed programs, the model intersection property holds for weakly completed programs \[33\]. This guarantees the existence of a least model for every program. Stenning and van Lambalgen \[34\] devised such an operator which has been generalized for first-order programs by \[27\]: Let \(I\) be an interpretation in \(\Phi_{SvL,P}(I) = \langle J^\top, J^\bot \rangle\), where

\[
\begin{align*}
J^\top &= \{A \mid \text{there exists a clause } A \leftarrow \text{Body} \in P \text{ with } I(\text{Body}) = \top\}, \\
J^\bot &= \{A \mid \text{there exists a clause } A \leftarrow \text{Body} \in P \text{ and for all clauses } A \leftarrow \text{Body} \in P \text{ we find } I(\text{Body}) = \bot\}.
\end{align*}
\]

As shown in \[27\] the least fixed point of \(\Phi_{SvL,P}\) is identical to the least model of the weak completion of \(P\) (\(\text{lm}_w\text{wc} P\)) under three-valued Łukasiewicz semantics. From \(I = \langle \emptyset, \emptyset \rangle\), \(\text{lm}_w\text{wc} P\) is computed by iterating \(\Phi_{SvL,P}\). Given a program \(P\) and a formula \(F\) \(P \models F\) iff \(\text{lm}_w\text{wc} P(F) = \top\) for formula \(F\).
Abductive Counterfactual Reasoning

Table 2: The $\text{lm}_{wc}$ and the well-founded models of some program examples.

<table>
<thead>
<tr>
<th>Program $\mathcal{P}$</th>
<th>$\text{lm}_{wc} \mathcal{P}$</th>
<th>Well-founded Model of $\mathcal{P}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{P}_1 = { p \leftarrow q }$</td>
<td>$\langle \emptyset, \emptyset \rangle$</td>
<td>$\langle \emptyset, { p, q } \rangle$</td>
</tr>
<tr>
<td>$\mathcal{P}_2 = { p \leftarrow \neg q, q \leftarrow \neg p }$</td>
<td>$\langle \emptyset, \emptyset \rangle$</td>
<td>$\langle \emptyset, \emptyset \rangle$</td>
</tr>
<tr>
<td>$\mathcal{P}_3 = { p \leftarrow q, q \leftarrow p }$</td>
<td>$\langle \emptyset, \emptyset \rangle$</td>
<td>$\langle \emptyset, { p, q } \rangle$</td>
</tr>
<tr>
<td>$\mathcal{P}_4 = { p \leftarrow \neg p }$</td>
<td>$\langle \emptyset, \emptyset \rangle$</td>
<td>$\langle \emptyset, \emptyset \rangle$</td>
</tr>
</tbody>
</table>

Proposition 1. Given a definite $\mathcal{P}$ and an atom $A$, the following holds:

$$\mathcal{P} \models A \iff \mathcal{P} \models_{lmwc} A.$$ 

Proof. $\mathcal{P} \models A \iff A \in \text{lfp} \Phi_{S_{\mathcal{T}} \mathcal{P}}$. By the definition of the $\Phi_{S_{\mathcal{T}} \mathcal{P}}$ operator, where $\text{lfp} \Phi_{S_{\mathcal{T}} \mathcal{P}} = \langle I^\top, I^\perp \rangle$, $A \in I^\top$, therefore $\mathcal{P} \models_{lmwc} A$. 

Notice that $\Phi_{S_{\mathcal{T}}}$ differs in a subtle way from the well-known Fitting operator $\Phi_F$, introduced in [35]: The definition of $\Phi_F$ is like that of $\Phi_{S_{\mathcal{T}}}$, except that in the specification of $J^\perp$ the first line “there exists a clause $A \leftarrow \text{Body} \in \mathcal{P}$ and” is dropped. The least fixed point of $\Phi_{F, \mathcal{P}}$ corresponds to the least model of the completion of $\mathcal{P}$. If an atom $A$ is undefined in the program $\mathcal{P}$, then, for arbitrary interpretations $I$ it holds that $A \in J^\perp$ in $\Phi_{F, \mathcal{P}}(I) = \langle J^\top, J^\perp \rangle$, whereas if $\Phi_{S_{\mathcal{T}}}$ is applied instead of $\Phi_F$, this does not hold for any interpretation $I$.

The correspondence between weak completion semantics and well-founded semantics [36] for tight programs, i.e. those without positive loops, is shown in [37]. Table 2 shows some examples which give an intuitive idea of their similarities and differences.

3.4 Integrity Constraints

A set of integrity constraints $\mathcal{IC}$ comprises clauses of the form $\bot \leftarrow \text{Body}$, where $\text{Body}$ is a conjunction of literals. $\mathcal{P}$ satisfies $\mathcal{IC}$ iff $\mathcal{P} \cup \mathcal{IC}$ are satisfiable. Under two-valued semantics this means that there exists a two-valued model for $\mathcal{P} \cup \mathcal{IC}$. This unambiguously implies that for each clause in $\mathcal{IC}$, $\text{Body}$ is false under this model. Under three-valued semantics, there are several ways on how to understand integrity constraints [38], two of them being the theoremhood view and the consistency view. Consider $\mathcal{IC}$:

$$\bot \leftarrow \neg p \land q.$$
The theoremhood view requires that a model only satisfies the set of integrity constraints if for all its clauses, \( \text{Body} \) is false under this model. In this example, this is only the case if \( p \) is true or if \( q \) is false in the model. In the consistency view, the set of integrity constraints is satisfied by the model if \( \text{Body} \) is unknown or false in it. In this example, that means, a model satisfies \( \mathcal{IC} \) already if either \( p \) or \( q \) are unknown in it.

We follow the consistency view extended for three-valued semantics, like the one presented by Pereira, Aparício and Alferes [39] for well-founded semantics. Given a program \( \mathcal{P} \) and a set of integrity constraints \( \mathcal{IC} \), \( \mathcal{P} \) satisfies \( \mathcal{IC} \) iff there exists an interpretation \( I \), which is a model for \( \mathcal{P} \), and for each \( \bot \leftarrow \text{Body} \in \mathcal{IC} \), we find that \( I(\text{Body}) \in \{ \bot, U \} \).

### 3.5 Abduction

Before we introduce three-valued abduction, let us first recall two-valued abduction [38]. A two-valued abductive framework is a quadruple \( \langle \mathcal{P}, A_2, \mathcal{P}, \mathcal{IC}, \models \rangle \), consisting of a definite program \( \mathcal{P} \) as knowledge base, a collection of abducibles \( A_2, \mathcal{P} \), a set of integrity constraints \( \mathcal{IC} \), and the logical consequence relation \( \models \). \( A_2, \mathcal{P} \) is:

\[
\{ A \leftarrow \top | A \in \text{undef}(\mathcal{P}) \}.
\]

An observation \( O \) is a non-empty set of atoms. This is analogous to a query whose explanation is desired, not necessarily (though possibly) something actually observed.

**Definition 1.** Let \( \langle \mathcal{P}, A_2, \mathcal{P}, \mathcal{IC}, \models \rangle \) be a two-valued abductive framework, \( \mathcal{E} \subseteq A_2, \mathcal{P} \) and \( O \) an observation.

\( O \) is two-explained by \( \mathcal{E} \) given \( \mathcal{P} \) and \( \mathcal{IC} \) iff \( \mathcal{P} \not\models O \), \( \mathcal{P} \cup \mathcal{E} \models O \) and \( \mathcal{P} \cup \mathcal{E} \) satisfies \( \mathcal{IC} \).

\( O \) is two-explained given \( \mathcal{P} \) and \( \mathcal{IC} \) iff there exists an \( \mathcal{E} \) such that \( O \) is explained by \( \mathcal{E} \) given \( \mathcal{P} \) and \( \mathcal{IC} \).

We assume henceforth that explanations are minimal, that means, there is no other explanation \( \mathcal{E}' \subset \mathcal{E} \) for \( O \). When we deal with three-valued semantics, abducibles may not only be positive but can also take the form of negative facts. Therefore, we extend the set of abducibles \( A_\mathcal{P} \) as follows:

\[
\{ A \leftarrow \top | A \in \text{undef}(\mathcal{P}) \} \cup \{ A \leftarrow \bot | A \in \text{undef}(\mathcal{P}) \}.
\]

**Proposition 2.** Given a definite program \( \mathcal{P} \), the following holds:

If \( A \leftarrow \top \in A_2, \mathcal{P} \), then \( \{ A \leftarrow \top, A \leftarrow \bot \} \subseteq A_\mathcal{P} \).
Abductive Counterfactual Reasoning

Proof. This follows immediately from the definitions for $A_{2,P}$ and $A_P$. □

Note, for a three-valued abductive framework, $O$ is now a non-empty set of literals.

Definition 2. Let $\langle P, A_P, IC, \models_L^{lmwc} \rangle$ be an abductive framework, $E \subset A_P$ and $O$ an observation.

$O$ is explained by $E$ given $P$ and $IC$ iff

$$\begin{align*}
P \not\models_L^{lmwc} O, \\ P \cup E \models_L^{lmwc} O \text{ and } \text{lm}_{cw}(P \cup E) \text{ satisfies } IC.
\end{align*}$$

$O$ is explained given $P$ and $IC$ iff there exists an $E$ such that $O$ is explained by $E$ given $P$ and $IC$.

In case abducibles are not abduced as positive or negative facts, they stay unknown in the least model of the weak completion.

Proposition 3. Given a two-valued abductive framework $\langle P, A_{2,P}, IC, \models \rangle$, an abductive framework $\langle P, A_P, IC, \models_L^{lmwc} \rangle$, where $P$ is definite, $E \subseteq A_{2,P}$ and observation $O$ a non-empty set of atoms. The following holds:

1. If $E$ is a two-explanation for $O$ given $P$ and $IC$, then $E$ is an explanation for $O$ given $P$ and $IC$.

2. If $O$ is two-explained given $P$ and $IC$ then $O$ is explained given $P$ and $IC$.

The proof is in Appendix A. The other direction does not hold. Consider $P$:

$$p \leftarrow q,$$

and observation $O = \{\overline{p}\}$. Its explanation is $E = \{q \leftarrow \perp\}$, where $E \in A_P$. However, $E \not\in A_{2,P}$ and therefore it is not a two-explanation for $O$.

4 Abductive Counterfactual Reasoning

Based on Pearl’s theory, Sloman extensively clarifies and discusses the distinction between causal and counterfactual reasoning in [40]. In both cases, a specific relation of cause and effect is described. With the following two examples we want to clarify how abduction models causal relations on the one hand, and why abduction alone is not adequate to model counterfactual reasoning, on the other hand. Let’s consider $P$:

$$b \leftarrow r,$$
whose least model of the weak completion is \(\langle \emptyset, \emptyset \rangle\), because \(r\) is unknown. We conjecture the following counterfactual

\[ \text{She would have gone to the beach (}b\text{), if it wouldn’t have rained (}r\text{). (}\text{CF}_{b,r}\text{)} \]

Let us impose the hypothetical truth value of the antecedent of \(\text{CF}_{b,r}\) by simply adding it as a negative fact, \(r \leftarrow \bot\), to \(\mathcal{P}\):

\[
\begin{align*}
    b & \leftarrow \overline{r}, \\
    r & \leftarrow \bot.
\end{align*}
\]

The corresponding least model of the weak completion is \(\langle \{b\}, \{r\} \rangle\). Indeed \(b\) is true and thus \(\text{CF}_{b,r}\) seems to be valid wrt \(\mathcal{P}\). Assume that initially \(r\) is actually true.

\[
\begin{align*}
    b & \leftarrow \overline{r}, \\
    r & \leftarrow \top.
\end{align*}
\]

Imposing the (hypothetical) negative fact \(r \leftarrow \bot\), leads to

\[
\begin{align*}
    b & \leftarrow \overline{r}, \\
    r & \leftarrow \top, \\
    r & \leftarrow \bot.
\end{align*}
\]

Consider its weak completion

\[
\begin{align*}
    b & \leftrightarrow \overline{r}, \\
    r & \leftrightarrow \top \lor \bot.
\end{align*}
\]

The corresponding least model of the weak completion is \(\langle \{r\}, \{b\} \rangle\) and does not imply \(b\): Simply imposing hypothetically \(r \leftarrow \bot\) didn’t have any effect at all.

In the following we present an approach that allows for real interventions and permits for adequate counterfactual evaluation. Our approach is epistemologically akin to Rescher’s, inspired by Pearl’s theory [8], technically extended to LP semantics by weak completion and abduction, and scaffolded, if needed, in declarative debugging and belief revision methods [41–44].

### 4.1 Program Transformation

As described in Pearl’s structural theory [8], a counterfactual requires a hypothetical modification of the current situation. Analogously to Pearl’s three step procedure discussed in Section 2.1 we apply his idea to logic programs, but leaving out probabilities. Pearl’s starting point is a model \(M\) which consists of a set of background (or exogenous) variables \(U\) whose values are given, like in an experiment, or else they depend on current observations or evidences \(e\), but are not causally explained by \(M\), as they have no parent nodes.

Let’s consider a program \(\mathcal{P}\), a set of integrity constraints \(\mathcal{IC}\) and a set of abducibles \(\mathcal{AP}\) establishing and anchoring an explanation \(\mathcal{E}\) of a current observation \(\mathcal{O}\):
Explanation $E$ corresponds to Pearl’s definition for background variables $U$ as there is no preceding causal explanatory mechanism encoded in the program to define them. They are abducibles and hence have no causes and no rules concluding them.

Observation $O$ corresponds to Pearl’s definition for evidence $e$, it being a set of facts assumed to be true or false. $O$ needs explanations in terms of $E$, by means of abductive explanations for those facts in $O$. As we don’t contemplate probabilities in our LP approach, $O$ may require some abducibles in $E$ just to be true or false, that is, as if having probability 0 or 1.

The least model $\text{lm}_{wc}(P \cup E)$ which explains $O$ and satisfies $IC$, corresponds to Pearl’s model $M$ updated according to $U$ and $e$.

Pearl’s first step of abduction in Bayesian networks can be translated into our abductive logic programming approach, when leaving out probabilities. Yet, we haven’t discussed the correspondence to Pearl’s second and third step, i.e., how action and predication should be formalized accordingly, in order to evaluate counterfactuals. Different than in abduction, in counterfactual reasoning the precondition might already have a truth value in $P$, which means that we cannot ensure that it is in $A_P$. For this reason, we introduce a reserved abducibles constructor $\text{mk}(A)$ for all atoms $A \in P$, with no rules in $P$ and hence undefined. The set of $\text{mk}$-abducibles $A_{pk}\text{m}$ is:

$$
\{ \text{mk}(A) \leftarrow \top \mid P \not\models_{lm_{wc}} A \} \cup \{ \text{mk}(A) \leftarrow \bot \mid P \not\models_{lm_{wc}} \overline{A} \}.
$$

If the precondition states that if $A$ had been true or if $A$ had been false, we abduce $\text{mk}(A) \leftarrow \top$ or $\text{mk}(A) \leftarrow \bot$, about which we say that we $\text{mk}$-abduce $A \leftarrow \top$ or $A \leftarrow \bot$, respectively. Note that, we can only abduce the truth about something when it is not the case already. In other words, we cannot abduce $A$ being false (or true), if it is false (or true) in $P$. This complies with the natural language understanding of counterfactuals [45].

**Proposition 4.** Given a program $P$, the sets $A_P$, and $A_{pk}\text{m}$, the following holds:

If $\{ A \leftarrow \top, A \leftarrow \bot \} \subseteq A_P$ then $\{ \text{mk}(A) \leftarrow \top, \text{mk}(A) \leftarrow \bot \} \subseteq A_{pk}\text{m}$.

**Proof.** This follows immediately from the definitions for $A_P$ and $A_{pk}\text{m}$. \hfill $\square$

The other direction does not hold. Consider $P = \{ q \leftarrow \top \}$ where $\text{lm}_{wc}P = \langle \{q\}, \emptyset \rangle$. There is no atom in $P$ that is undefined, therefore $\text{undef}(P)$ is empty and thus the
set of abducibles $A_P$ is empty. The set of mk-abducibles, $A_{P}^{mk}$, contains exactly one (abducible) clause:

$$\text{mk}(q) \leftarrow \bot.$$  

Let us consider the following transformation for $P$ and a given set of literals $S$:

For all $L \in S$, where $L = A$ or $L = \overline{A}$, consider $\text{mk}(A)$ and do as follows:

- **Conjoin** $\text{mk}(A)$ to the body of all clauses in the definition of $A$.
- **Add** $A \leftarrow \text{mk}(A)$.

The new program is denoted $P^{mk}(S)$. The idea is that subsequently $\text{mk}(A)$ can defeat all the rules for $A$ if made false, and can impose $A$ if made true (cf. Proposition 6). Let us exemplify the transformation with the simple program for $q$, given above. Given that $S = \{q\}$, $P^{mk}(S)$ is:

$$q \leftarrow T \land \text{mk}(q), \quad q \leftarrow \text{mk}(q).$$  

The $\text{lm}_{L\text{wc}}P^{mk}(S)$ is empty. However, $\text{lm}_{L\text{wc}}P^{mk}(S) \cup \{\text{mk}(q) \leftarrow \bot\}$ is:

$$\langle \emptyset, \{\text{mk}(q), q\}\rangle.$$

**Proposition 5.** Given a program $P$ and a set of literals $S$, the following holds:

Every dependency in $P$ is also a dependency in $P^{mk}(S)$.

*Proof.* By transforming $P$ to $P^{mk}(S)$, no clauses or literals are eliminated but only added. Therefore, every dependency that was previously in $P$ is also in $P^{mk}(S)$. $\Box$

**Proposition 6.** Given a program $P$ and a set of literals $S$, the following holds:

Every $L \in S$ is unknown in $\text{lm}_{L\text{wc}}P^{mk}(S)$.

The proof is in Appendix A.

**Proposition 7.** Given a program $P$ and a set of literals $S$, the following holds:

$$\text{lm}_{L\text{wc}}P^{mk}(S) \subseteq \text{lm}_{L\text{wc}}P.$$  

*Sketch of proof.* After the transformation no new facts are added, and, consequently no new facts can be retrieved. The possible subset relation is because according to Proposition 6, every $L \in S$ is unknown in $\text{lm}_{L\text{wc}}P^{mk}(S)$. Consequently, every retrieved fact in $\text{lm}_{L\text{wc}}P$ that depends on $L$ possibly becomes unknown in $\text{lm}_{L\text{wc}}P^{mk}(S)$. $\Box$
The following example shows the intuition behind Proposition 7. 

\[\begin{align*}
    r & \leftarrow p \land q, & p & \leftarrow \top, \\
    s & \leftarrow p \land \neg q, & q & \leftarrow \top.
\end{align*}\]

\(\text{Im}_{\text{wc}} \mathcal{P}\) is \(\langle \{p, q, r\}, \{s\} \rangle\). After program transformation wrt \(S = \{p\}\), \(\mathcal{P}^\text{mk}(S)\) is:

\[\begin{align*}
    r & \leftarrow p \land q, & p & \leftarrow \top \land \text{mk}(p), & p & \leftarrow \text{mk}(p), \\
    s & \leftarrow p \land \neg q, & q & \leftarrow \top.
\end{align*}\]

Its weak completion is

\[\begin{align*}
    r & \leftrightarrow q \land p, & p & \leftrightarrow \text{mk}(p) \lor (\top \land \text{mk}(p)), \\
    s & \leftrightarrow \neg q \land p, & q & \leftrightarrow \top.
\end{align*}\]

\(\text{Im}_{\text{wc}} \mathcal{P}^\text{mk}(S)\) is \(\langle \{q\}, \{s\} \rangle\), where \(p\) and \(r\) are now unknown. An \(\text{mk}\)-explanation is similar to \(\mathcal{E}\) in abduction, a set of facts. We use the following notation wrt a set \(S\):

\[\begin{align*}
    \mathcal{E}_{\text{mk}(S)} & = \bigcup_{L \in S} \mathcal{E}_{\text{mk}(L)} \quad \text{where} \quad \mathcal{E}_{\text{mk}(L)} = \begin{cases} 
        \{\text{mk}(A) \leftarrow \top\} & \text{if } L = A, \\
        \{\text{mk}(A) \leftarrow \bot\} & \text{if } L = \overline{A}.
    \end{cases}
\end{align*}\]

Similarly, we denote an explanation \(\mathcal{E}\) wrt a set \(S\) as follows:

\[\begin{align*}
    \mathcal{E}_S & = \bigcup_{L \in S} \mathcal{E}_L \quad \text{where} \quad \mathcal{E}_L = \begin{cases} 
        \{A \leftarrow \top\} & \text{if } L = A, \\
        \{A \leftarrow \bot\} & \text{if } L = \overline{A}.
    \end{cases}
\end{align*}\]

**Proposition 8.** Given a program \(\mathcal{P}\), a consistent set of literals \(S\) and \(\mathcal{E}_{\text{mk}(S)} \subseteq \mathcal{A}_\mathcal{P}^\text{mk}\), the following holds:

\[\forall L \in S : \mathcal{P}^\text{mk}(S) \cup \mathcal{E}_{\text{mk}(S)} \models_{\text{Im}_{\text{wc}}} L.\]

**Proposition 9.** Given a program \(\mathcal{P}\), a consistent set of literals \(S\), \(\mathcal{E}_{\text{mk}(S)} \subseteq \mathcal{A}_\mathcal{P}^\text{mk}\) and \(\mathcal{E}_S \subset \mathcal{A}_\mathcal{P}\), the following holds:

\[\text{Im}_{\text{wc}}(\mathcal{P} \cup \mathcal{E}_S) \subset \text{Im}_{\text{wc}}(\mathcal{P}^\text{mk}(S) \cup \mathcal{E}_{\text{mk}(S)}).\]

**Proposition 10.** Given a program \(\mathcal{P}\), a consistent set of literals \(S\), \(\mathcal{E}_{\text{mk}(S)} \subseteq \mathcal{A}_\mathcal{P}^\text{mk}\) and \(\mathcal{E}_S \subset \mathcal{A}_\mathcal{P}\), the following holds:

\[\text{Im}_{\text{wc}}(\mathcal{P} \cup \mathcal{E}_S) = \text{Im}_{\text{wc}}(\mathcal{P}^\text{mk}(S) \cup \mathcal{E}_{\text{mk}(S)}) \setminus \{\text{mk}(A) \mid \text{mk}(A) \in \mathcal{A}_\mathcal{P}^\text{mk}\}.\]

The proofs for Proposition 8, 9 and 10 are in Appendix A.
4.2 Evaluating Counterfactuals

A counterfactual statement is of the form

\[ \text{Conc would have happened, if } S \text{ had been the case.} \]  

\((\text{CF}_{S,\text{Conc}})\)

where \(S\) and \(\text{Conc}\) are each a set of literals.

**Definition 3.** Let \(\langle P, \mathcal{A}_P, \text{CF}_{S,\text{Conc}}, \mathcal{I}C, \models_{\text{lmc}} \rangle\) be a counterfactual framework where \(\mathcal{E}_{\text{mk}(S)} \subset \mathcal{A}_P\).

\(\text{CF}_{S,\text{Conc}}\) is a valid counterfactual given \(P\) and \(\mathcal{I}C\) iff

\[ \mathcal{P} \not\models_{\text{lmc}} \text{Conc}, \mathcal{P}\text{mk}(S) \cup \mathcal{E}_{\text{mk}(S)} \models_{\text{lmc}} \text{Conc}, \text{ and } \]

\[ \text{lm}_{\text{lwc}}(\mathcal{P}\text{mk}(S) \cup \mathcal{E}_{\text{mk}(S)}) \text{ satisfies } \mathcal{I}C. \]

**Theorem 11.** Given an abductive framework \(\langle P, \mathcal{A}_P, \mathcal{I}C, \models_{\text{lmc}} \rangle\) and a counterfactual framework \(\langle P, \mathcal{A}_P, \text{CF}_{S,\text{Conc}}, \mathcal{I}C, \models_{\text{lmc}} \rangle\), where observation \(O = \text{Conc}\). The following holds:

If \(\mathcal{E}_S\) is an explanation for \(O\) given \(P\) and \(\mathcal{I}C\),
then \(\text{CF}_{S,\text{Conc}}\) is a valid counterfactual given \(P\) and \(\mathcal{I}C\).

**Proof.** Let us assume that \(\mathcal{E}_S\) is an explanation for \(O\) given \(P\) and \(\mathcal{I}C\), that is,

\[ \mathcal{P} \not\models_{\text{lmc}} O, \mathcal{P} \cup \mathcal{E}_S \models_{\text{lmc}} O \text{ and } \text{lm}_{\text{lwc}}(\mathcal{P} \cup \mathcal{E}_S) \text{ satisfies } \mathcal{I}C. \]

To show: \(\mathcal{P}\text{mk}(S) \cup \mathcal{E}_{\text{mk}(S)} \models_{\text{lmc}} O\), and \(\text{lm}_{\text{lwc}}(\mathcal{P}\text{mk}(S) \cup \mathcal{E}_{\text{mk}(S)}) \text{ satisfies } \mathcal{I}C. \)

1. \(\mathcal{P}\text{mk}(S) \cup \mathcal{E}_{\text{mk}(S)} \models_{\text{lmc}} O\) follows from \(\mathcal{P} \cup \mathcal{E}_S \models_{\text{lmc}} O\) and Proposition 9.

2. \(\text{lm}_{\text{lwc}}(\mathcal{P}\text{mk}(S) \cup \mathcal{E}_{\text{mk}(S)}) \text{ satisfies } \mathcal{I}C, \) follows from \(\text{lm}_{\text{lwc}}(\mathcal{P} \cup \mathcal{E}_S) \text{ satisfies } \mathcal{I}C, \)
by Proposition 10 and because \(\mathcal{I}C\) does not include mk-predicates.

Let us assume a counterfactual statement in the context of some observation:

\[ \text{Given } O, \text{ Conc would have happened, if } S \text{ had been the case.} \]  

\((\text{CF}^O_{S,\text{Conc}})\)

\(O\) is an observation, which is to be explained by some abducible wrt \(P\) and \(\mathcal{I}C\). We extend Definition 3 accordingly by integrating usual abduction in the framework.
Definition 4. Let \( \langle P, \mathcal{A}_P, A^\mk_P, CF^O_{S, Conc}, IC, \models_{lmwc} \rangle \) be a counterfactual abductive framework, where \( \mathcal{E} \subset \mathcal{A}_P \), \( \mathcal{E}_{\mk(S)} \subset A^\mk_P \) and \( \mathcal{O} \) is an observation.

\( CF^O_{S, Conc} \) is a valid counterfactual given \( \mathcal{O}, P \) and \( IC \) iff

\[
P \models_{lmwc} Conc, \mathcal{E} \text{ explains } \mathcal{O} \text{ given } IC \text{ and } P^{\mk(S)} \cup \mathcal{E}_{\mk(S)}, P^{\mk(S)} \cup \mathcal{E}_{\mk(S)} \cup \mathcal{E} \models_{lmwc} Conc, \text{ and } \text{lmcwc}(P^{\mk(S)} \cup \mathcal{E}_{\mk(S)} \cup \mathcal{E}) \text{ satisfies } IC.
\]

Section 5 motivates by examples the need for abduction in counterfactual reasoning.

5 Examples

5.1 Had the forest fire occurred, if...?

Consider the counterfactual from the introduction

\[
\text{If only there had not been so many dry leaves on the forest floor, } (CF_{dry, fire}) \text{ then the forest fire wouldn’t have occurred.}
\]

Let us assume the following program \( P \) where \( IC = \emptyset \):

\[
\begin{align*}
\text{forest}_fire & \leftarrow \text{lightning} \land \overline{ab}, & \text{lightning} & \leftarrow T, \\
ab & \leftarrow \text{dry_leaves}, & \text{dry_leaves} & \leftarrow T.
\end{align*}
\]

According to Definition 3 we identify that

\[
S = \{\text{dry_leaves}\} \quad \text{and} \quad \text{Conc} = \{\text{forest}_fire\}.
\]

\( \text{lmcwc } P \) is \( \{\{\text{dry_leaves}, \text{lightning, forest_fire}\}, \{ab\}\} \), which satisfies the first requirement of Definition 3. The transformation of \( P \) wrt \( S \) is \( P^{\mk(S)} \):

\[
\begin{align*}
\text{forest}_fire & \leftarrow \text{lightning} \land \overline{ab}, & \text{lightning} & \leftarrow T, \\
ab & \leftarrow \text{dry_leaves}, & \text{dry_leaves} & \leftarrow T \land \mk(\text{dry_leaves}), \\
& & \text{dry_leaves} & \leftarrow \mk(\text{dry_leaves}),
\end{align*}
\]

for which the weak completion together with \( \mathcal{E}_{\mk(S)} = \{\mk(\text{dry_leaves}) \leftarrow \bot\} \) is

\[
\begin{align*}
\text{forest}_fire & \leftrightarrow \text{lightning} \land \overline{ab}, & \text{lightning} & \leftrightarrow T, \\
ab & \leftrightarrow \text{dry_leaves}, & \mk(\text{dry_leaves}) & \leftrightarrow \bot, \\
\text{dry_leaves} & \leftrightarrow (T \land \mk(\text{dry_leaves})) \lor \mk(\text{dry_leaves}).
\end{align*}
\]
\[ \text{lm}_{\text{wc}} (P^{\text{mk}(S)} \cup E^{\text{mk}(S)}) \text{ is } \{\text{lightning}, \text{ab}\}, \{\text{mk(dry_leaves)}, \text{dry_leaves, forest_fire}\} \]
and indeed entails Conc. Let us extend \( P \) by a clause that represents (C2):
\[
\text{ffire} \leftarrow \text{fire-raising}.
\]
S and Conc stay the same. Transformed \( P^{\text{mk}(S)} \) together with \( E^{\text{mk}(S)} \) is:
\[
\begin{align*}
\text{ffire} & \leftarrow \text{lightning} \land \text{dry_leaves}, \\
\text{ffire} & \leftarrow \text{fire-raising}, \\
\text{lightning} & \leftarrow \top, \\
\text{dry_leaves} & \leftarrow \top \land \text{mk(dry_leaves)}, \\
\text{mk(dry_leaves)} & \leftarrow \bot.
\end{align*}
\]
The corresponding weak completion is:
\[
\begin{align*}
\text{ffire} & \leftrightarrow (\text{lightning} \land \text{dry_leaves}) \lor \text{fire-raising}, \\
\text{lightning} & \leftrightarrow \top, \\
\text{dry_leaves} & \leftrightarrow \top \land \text{mk(dry_leaves)}, \\
\text{mk(dry_leaves)} & \leftrightarrow \bot,
\end{align*}
\]
and its least model of its weak completion is:
\[
\langle \{\text{lightning}\}, \{\text{dry_leaves}, \text{mk(dry_leaves)}\}\rangle.
\]
It does not state any truth about \( \text{ffire} \), because \( \text{fire-raising} \) is unknown. \( E^{\text{mk}(S)} \) does not counterfactually explain Conc, thus \( \text{CF}_{\text{dry,ffire}} \) is not a valid counterfactual.

Note that \( \text{CF}_{\text{dry,ffire}} \) is invalid only because the weak completion semantics adopts an open world assumption on undefined atoms. Under well-founded and stable model semantics \( \text{fire-raising} \) would have been assumed false, which entails \( \neg \text{ffire} \) to be in the well-founded model. Accordingly, under well-founded semantics we would conclude \( \text{CF}_{\text{dry,ffire}} \) to be valid instead. We assume that the evaluation according to weak completion semantics is more appropriate. Our argument is as follows: if it is known that there is a forest fire, which would either have occurred through a lightning (and only when the leaves are dry) (C1), or by a fire-raising (C2), the appropriate answer to, whether \( \text{CF}_{\text{dry,ffire}} \) is valid, should be it is unknown because a fire-raising could have been the cause for the forest fire. For the well-founded model to produce the same result, \( \text{fire-raising} \) would need to be explicitly declared unknown, say by means of \( \text{fire-raising} \leftarrow U \), with reserved atom \( U \) defined as \( U \leftarrow \neg U \). An even more interesting answer would be yes, but only if there had not been a fire-raising.

Let us assume that arsonists would never go out when there is a storm, especially not when there is lightning: there would never be fire-raising and lightning at the same time. We represent this information by the IC:
\[
\bot \leftarrow \text{fire-raising} \land \text{lightning}.
\]
According to Definition 4 from the knowledge imparted by this IC, we assume the background observation expressed by \( O = \{ \neg \text{fire-raising} \} \). It’s corresponding only explanation is \( \mathcal{E} \):

\[
\text{fire-raising} \leftarrow \bot.
\]

According to Definition 4 we need to transform by \((P \cup \mathcal{E}_{\text{mk}(S)} \cup \mathcal{E})_{\text{mk}(S)}\):

\[
\begin{align*}
\text{ffire} & \leftarrow \text{lightning} \land \text{dry\_leaves}, \\
\text{lightning} & \leftarrow \top, \\
\text{ffire} & \leftarrow \text{fire-raising}, \\
\text{fire-raising} & \leftarrow \bot, \\
\text{dry\_leaves} & \leftarrow \top \land \text{mk}(\text{dry\_leaves}), \\
\text{mk}(\text{dry\_leaves}) & \leftarrow \bot.
\end{align*}
\]

Its weak completion is:

\[
\begin{align*}
\text{ffire} & \leftrightarrow (\text{lightning} \land \text{dry\_leaves}) \lor \text{fire-raising}, \\
\text{lightning} & \leftrightarrow \top, \\
\text{fire-raising} & \leftrightarrow \bot, \\
\text{dry\_leaves} & \leftrightarrow \top \land \text{mk}(\text{dry\_leaves}), \\
\text{mk}(\text{dry\_leaves}) & \leftrightarrow \bot.
\end{align*}
\]

The least model of the weak completion is:

\[
\langle \{\text{lightning}\}, \{\text{dry\_leaves}, \text{mk}(\text{dry\_leaves}), \text{fire-raising}, \text{mk}(\text{fire-raising}), \text{ffire}\} \rangle,
\]

which indeed implies Conc. By Definition 4 \( \mathcal{E}_{\text{mk}(S)} \) counterfactually explains Conc given \( O, P \) and IC. Note that this example differs to the previous one, in the way that now, by the background observations, we additionally assumed fire-raising to be false. This new result allows us to extract a counterfactual that refines \( \text{CF}_{\text{dry,ffire}} \):

\[
\text{Given that there had not been a fire-raising,}
\]

\[
\text{If only there had not been so many dry leaves on the forest floor,}
\]

\[
\text{then the forest fire wouldn’t have occurred.}
\]

Note also that the premise of a counterfactual may implicitly provide with us observations that further add to explanations. We might not have known for a fact that the dry leaves were on the forest floor before we actually heard about the counterfactual. The latter instructs us to presuppose that they were there indeed. But that may in turn require us to abduce a background cause, such that there was a strong wind, namely if we were told that \( \text{dry\_leaves} \leftarrow \text{strong\_wind} \). To complicate matters, a strong wind may lead to an uncontrolled forest fire, opening the way to more complex counterfactual conclusions. In summary, counterfactual premises can lead to implicit secondary observations in need of explanation.
5.2 Someone else shot Kennedy, if...?

Consider the following two statements from [46]:

- If Oswald hadn’t killed Kennedy, someone else would have.  \( (CF_{os,se}) \)
- If Oswald didn’t kill Kennedy, someone else did.  \( (I_{os,se}) \)

The first statement is a counterfactual whereas the second one is an indicative conditional. The difference on how we understand them, is that, \( CF_{os,se} \) asks for revising not only that Oswald hadn’t shot Kennedy, but its consequences involved, namely that Kennedy has in fact been killed. In contrast to this interpretation of the counterfactual, the other one, \( I_{os,se} \), implies that Kennedy was actually killed. According to [46], the majority of the people rejects the first but accepts the second statement.

According to Definition 3 let us evaluate \( CF_{os,se} \):

\[
S = \{oswald\_shot\} \quad \text{and} \quad \text{Conc} = \{someone\_else\_shot\}.
\]

where \( P \) is

\[
\begin{align*}
\text{kennedy\_died} & \leftarrow oswald\_shot, \\
\text{kennedy\_died} & \leftarrow \text{someone\_else\_shot}.
\end{align*}
\]

The least model of the weak completion of \( P \) is \( \langle \{oswald\_shot, k\_died\} , \emptyset \rangle \), which complies with the first requirement in Definition 3. \( P^{mk(S)} \) is

\[
\begin{align*}
\text{kennedy\_died} & \leftarrow oswald\_shot, \\
\text{kennedy\_died} & \leftarrow \text{someone\_else\_shot}, \\
oswald\_shot & \leftarrow T \land \text{mk(oswald\_shot)}, \\
oswald\_shot & \leftarrow \text{mk(oswald\_shot)}.
\end{align*}
\]

Its weak completion together with:

\[
E_{mk(S)} = \{\text{mk(oswald\_shot) } \leftarrow \bot\},
\]

is:

\[
\begin{align*}
kennedy\_died & \leftrightarrow oswald\_shot \lor \text{someone\_else\_shot}, \\
oswald\_shot & \leftrightarrow (T \land \text{mk(oswald\_shot)}) \lor \text{mk(oswald\_shot)}, \\
\text{mk(oswald\_shot)} & \leftrightarrow \bot.
\end{align*}
\]

Its least model of the weak completion is:

\[
\langle \emptyset, \{\text{mk(oswald\_shot)}, oswald\_shot\} \rangle,
\]

which does not entail Conc, thus \( CF_{os,se} \) is not valid. This complies with the conclusion the majority of people would answer.
According to Definition 4 let us evaluate $Ios$, where still $S = \{oswald\_shot\}$ and $Conc = \{someone\_else\_shot\}$. Additionally, $Ios$ implies $kennedy\_died$, which affords us background information that needs to be explained in the context. Accordingly, we define observation $O = \{kennedy\_died\}$. $P^{mk}(S)$ and $E^{mk}(S)$ are defined as just discussed for $CF_{os,se}$. The only explanation for $O$ wrt $P^{mk}(S) \cup E^{mk}(S)$, is

$$E = \{someone\_else\_shot \leftarrow \top\}$$

Consider the weak completion of $(P \cup E^{mk}(S) \cup E)^{mk}(S)$:

- $kennedy\_died \leftrightarrow oswald\_shot \lor someone\_else\_shot$,
- $oswald\_shot \leftrightarrow (\top \land mk(oswald\_shot)) \lor mk(oswald\_shot)$,
- $mk(oswald\_shot) \leftrightarrow \bot$,
- $someone\_else\_shot \leftrightarrow \top$,

for which the corresponding least model of the weak completion now entails $Conc$:

$$\langle \{kennedy\_died, someone\_else\_shot\}, \{oswald\_shot\} \rangle,$$

and thus $Ios$,se is valid, which corresponds to the opinion of the majority of people.

6 Deep Revision

In our approach to counterfactual reasoning, revision is limited to the facts or conclusions of clauses explicitly stated in the premise. However, our approach is not just applicable to superficial revision afforded by their prevention. One could think of a more sophisticated revision procedure that would include revising clause body atoms related to the premise but not explicitly stated in the counterfactual. As our usage of the mk’s is quite general and flexible, it can be easily extended to revisable facts (or clauses) by following up on them in clause bodies accordingly, to apply the second step of the intervention and prevention procedure described above. The same goes for the prevention prompted by ICs, explained in Section 3.4.

We opted for removing surface contradictions, by analogy with Pearl (and LP updates for that matter), but we could well imagine that some rules may be protected, so one would not introduce mk’s in their body; then one may have to revise the antecedent conditions in their body; that leads to (skeptical or credulous) non-deterministic revision in general, and of course preferences. In the context of declarative debugging, [41] introduces preference relations and distinguishes between stable (or protected) and changeable rules. When avoiding the inclusion of mk in rules we
mean to consider them un-revisable, non-counterfactualizable, but then possibly allowing for revision of the clauses for their subgoals, possibly using preferences, the preferences themselves being revisable. An approach that deals with the deep belief revision of assumptions in LPs under the well-founded semantics, if needed by counterfactuals, including integrity constraints and protected clauses, and subject to a possible world semantics, is presented in [1].

7 Conclusion and Future Work

We have presented a computational logic approach for counterfactuals in the context of human reasoning by analyzing investigated examples by psychologists such as Byrne. LP seems to be the ideal candidate as we can straightforwardly represent required knowledge in a meta-language. With integrity constraints we can allow for wider, deeper, revisions, and the integrity constraints can be productive of abductions. With inspection points, not shown here, we have the option of integrity constraints being just consumers and not producers of abducibles [29,30]. The examples under consideration do not require deep revision, and one doubts people would use counterfactuals that way; nevertheless, we can potentially bring in the power of logic programming, by e.g. deep non-deterministic revision, preferences, meta-preferences, and meta-reasoning. These can also express embedded counterfactuals, to be treated in stages.

However, we haven’t dealt with general counterfactual reasoning and do not take into consideration some of its relevant aspects. For instance, the counterfactual premise can lead to contradictions or IC violations, and there may be more than one way of correcting them, some more plausible than others, with different side-effects than others. For this purpose we can extend our framework to include inspection points [47], to help rule out solutions. This relates Rescher’s [19] weakest link principle, dealing with determining the weakest link to eliminate an inconsistency. A framework taking the domain into consideration could be an extension using preferences to prioritize the links.

The testing of causality involves counterfactual reasoning, but, in turn, counterfactual reasoning needs to know about causes. This circular dependency is to be resolved on pragmatic grounds by the knowledge representation, e.g., changing the length of a pendulum causes a change of its period, but not vice-versa.
8 Acknowledgements

Thanks to Trevor Bench-Capon, Veronica Dahl, Chiaki Sakama & Ari Saptawijaya for valuable feedback. DAAD FITweltweit program supported 2nd author UNL stay.

References

Abductive Counterfactual Reasoning


A Proofs

Proposition 3. Given a two-valued abductive framework \( \langle P, A_2, P, IC, \models_{lmwc}^L \rangle \), and a three-valued abductive framework \( \langle P, A_P, IC, \models_{lmwc}^L \rangle \), where \( P \) is definite, \( E \subseteq A_2 \), and \( O \) is an observation, the following holds:

1. If \( E \) is a two-explanation for \( O \) given \( P \) and \( IC \), then \( E \) is an explanation for \( O \) given \( P \) and \( IC \).

2. If \( O \) is two-explained given \( P \) and \( IC \) then \( O \) is explained given \( P \) and \( IC \).

Proof. We only need to prove that (1) holds, because (2) follows from (1). Let us assume that \( E \) is a two-explanation for \( O \) given \( P \) and \( IC \), that means \( P \nmid O \), \( P \cup E \models O \) and \( \text{lm}_2(P \cup E) \) satisfies \( IC \).
To show: \( P \nmid O \), \( (P \cup E) \models O \) and \( \text{lm}_{wc}(P \cup E) \) satisfies \( IC \).

1. \( P \nmid O \) and \( P \cup E \models O \) follow immediately given \( P \nmid O \), \( P \cup E \models O \) and Proposition 1.

2. \( \text{lm}_2(P \cup E) \) satisfies \( IC \) means that \( \text{lm}_2(P \cup E \cup IC) \) is consistent. This implies that the body of all clauses in \( IC \) are false in \( \text{lm}_2(P \cup E) \). If they were true in \( \text{lm}_{wc}(P \cup E) \), then, according to Proposition 1 they would also have to be true in \( \text{lm}_2(P \cup E) \). Therefore, the body of all clauses in \( IC \) are either false or unknown in \( \text{lm}_{wc}(P \cup E) \). In either case, \( IC \) is satisfied.

Proposition 4. Given a program \( P \), the sets \( A_P \), and \( A_{mk}^P \), the following holds:

If \( A \leftarrow \top \) or \( A \leftarrow \bot \in A_P \) then \( \text{mk}(A) \leftarrow \top \in A_{mk}^P \) and \( \text{mk}(A) \leftarrow \bot \in A_{mk}^P \).

Proof. ‘\( A \leftarrow \top \) or \( A \leftarrow \bot \in A_P \)’ is a sufficient condition in the proposition, because \( A_P \) is the set of all positive and negative facts of all undefined atoms in \( P \). Therefore, whenever \( A \leftarrow \top \in A_P \), necessarily also \( A \leftarrow \bot \in A_P \), and vice versa. Given that \( A \in \text{undef}(P) \), \( A \) stays unknown in \( \text{lm}_{wc}P \). Accordingly, both, \( P \nmid_{lmwc}^L A \) and \( P \nmid_{lmwc}^L \overline{A} \) hold. Therefore, \( \text{mk}(A) \leftarrow \top \) and \( \text{mk}(A) \leftarrow \bot \in A_{mk}^P \).

Proposition 6. Given a program \( P \) and a set of literals \( S \), the following holds:

Every \( L \in S \) is unknown in \( \text{lm}_{wc}P_{mk(S)} \).

Proof. We distinguish between two cases for all \( L \in S \), where \( L = A \) or \( L = \overline{A} \).
1. \( A \in \text{undef}(\mathcal{P}) \), that means, \( A \) does not have a definition in \( \mathcal{P} \). Then, the definition of \( A \) in \( \mathcal{P}^{\text{mk}(S)} \), is \( \{ A \leftarrow \text{mk}(A) \} \). As \( \text{mk}(A) \) is a meta-predicate that has only been introduced by the program transformation, \( \text{mk}(A) \in \text{undef}(\mathcal{P}^{\text{mk}(S)}) \). Accordingly, \( \text{mk}(A) \) and therefore \( A \), are unknown in \( \text{lm}_{\text{wc}} \mathcal{P}^{\text{mk}(S)} \).

2. \( A \not\in \text{undef}(\mathcal{P}) \), that means, there is at least one clause with head \( A \) in \( \mathcal{P} \).
   
   (a) According to the program transformation, the body of every clause of the form
   
   \[
   A \leftarrow \text{Body}_1, A \leftarrow \text{Body}_2, \ldots \in \mathcal{P},
   \]
   
   is conjoined with \( \text{mk}(A) \), that is, \( \mathcal{P}^{\text{mk}(S)} \) consists of the clauses
   
   \[
   A \leftarrow \text{Body}_1 \land \text{mk}(A), A \leftarrow \text{Body}_2 \land \text{mk}(A), \ldots \in \mathcal{P},
   \]
   
   for every clause with head \( A \). Because \( \text{mk}(A) \in \text{undef}(\mathcal{P}^{\text{mk}(S)}) \), the body of these clauses are either unknown or false in \( \text{lm}_{\text{wc}} \mathcal{P}^{\text{mk}(S)} \).

   (b) The weak completion of \( \mathcal{P}^{\text{mk}(S)} \) contains
   
   \[
   A \leftrightarrow (\text{Body}_1 \land \text{mk}(A)) \lor (\text{Body}_2 \land \text{mk}(A)) \lor \cdots \lor \text{mk}(A).
   \]
   
   As \( \text{Body}_i \land \text{mk}(A) \), \( 1 \leq i \leq n \), are unknown or false according to (a) and \( \text{mk}(A) \) is unknown in \( \text{lm}_{\text{wc}} \mathcal{P}^{\text{mk}(S)} \) we conclude that \( A \) is is necessarily unknown in \( \text{lm}_{\text{wc}} \mathcal{P}^{\text{mk}(S)} \). Because \( \text{mk}(A) \in \text{undef}(\mathcal{P}^{\text{mk}(S)}) \), the body of these clauses are either unknown or false in \( \mathcal{P}^{\text{mk}(S)} \).

\[ \square \]

**Proposition 8.** Given a program \( \mathcal{P} \), a consistent set of literals \( S \) and \( \mathcal{E}_{\text{mk}(S)} \subseteq A_{\mathcal{P}^{\text{mk}}} \), the following holds:

\[
\forall L \in S : \mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)} \models_{\text{lm}_{\text{wc}}} L.
\]

**Proof.** We distinguish between the following two possible cases for all \( L \in S \):

1. Assume that \( L = \overline{A} \), thus \( \{ \text{mk}(A) \leftarrow \bot \} \in \mathcal{E}_{\text{mk}(S)} \). Accordingly,
   
   \[
   \text{mk}(A) \leftarrow \bot \in \text{wc} (\mathcal{P} \cup \mathcal{E}_{\text{mk}(S)}).
   \]
   
   By the program transformation \( \mathcal{P}^{\text{mk}(S)} \), \( \text{mk}(A) \) is added to each clause whose definition is \( A \), that is, for each such clause
   
   \[
   A \leftarrow \text{Body} \land \text{mk}(A) \in \text{wc} (\mathcal{P} \cup \mathcal{E}_{\text{mk}(S)}).
   \]
As each body whose head is \( A \) contains \( \bot \), it will always be false. Consequently, \( A \) is false in \( \text{lm}_L \text{wc} (\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)}) \), i.e.

\[
\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)} \models_{\text{lw}} S.
\]

2. Assume that \( L = A \), thus \( \{ \text{mk}(A) \leftarrow \top \} \in \mathcal{E}_{\text{mk}(S)} \). Accordingly,

\[
\text{mk}(A) \leftarrow \top \in \text{wc} (\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)}).
\]

By the program transformation \( \mathcal{P}^{\text{mk}(S)} \), \( \text{mk}(A) \) is added to each clause whose definition is \( A \), that is, for each such clause

\[
A \leftarrow \text{mk}(A) \lor \in \text{wc} (\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)}).
\]

As \( \text{mk}(A) \) is true, the disjunction of the body of \( A \) will always be true. Consequently, \( A \) is true in \( \text{lm}_L \text{wc} (\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)}) \), i.e.

\[
\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)} \models_{\text{lw}} A.
\]

Note that under weak completion semantics positive facts are prioritized over negative facts. Therefore in the second case, it would already be sufficient if we just added \( A \leftarrow \top \) to \( \mathcal{P} \), because \( A \leftarrow \top \lor \ldots \in \text{wc} (\mathcal{P} \cup \{ A \leftarrow \top \}) \). Consequently, the body can always be reduced to \( \top \), which is semantically equal to true.

**Proposition 9.** Given a program \( \mathcal{P} \), a consistent set of literals \( S \), \( \mathcal{E}_{\text{mk}(S)} \subseteq \mathcal{A}_{\mathcal{P}}^{\text{mk}} \) and \( \mathcal{E}_S \subseteq \mathcal{A}_P \), the following holds:

\[
\text{lm}_L \text{wc} (\mathcal{P} \cup \mathcal{E}_S) \subset \text{lm}_L \text{wc} (\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)}).
\]

**Proof.** Assume \( \text{lm}_L \text{wc} (\mathcal{P} \cup \mathcal{E}_S) = \langle I^T, I^\bot \rangle \) and \( \text{lm}_L \text{wc} (\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)}) = \langle J^T, J^\bot \rangle \). Given that \( \mathcal{E}_S \subseteq \mathcal{A}_P \), we know that for every \( L \in S \), where \( L = A \) or \( L = \overline{A} \), there exists a corresponding positive fact \( A \leftarrow \top \in \mathcal{A}_P \) and a corresponding negative fact \( A \leftarrow \bot \in \mathcal{A}_P \). We distinguish two cases for \( A \):

1. If \( A \in I^T \), then there is a clause \( A \leftarrow \text{Body} \), such that \( \text{Body} \) is true. We need to distinguish between three cases, to show that \( A \in J^T \):

   a. If \( A = L \), then the clause that determines \( A \)'s truth value in \( \mathcal{P} \cup \mathcal{E}_{\text{mk}(S)} \) is \( \mathcal{E}_{\text{mk}(S)} \). Analogously, the clause that determines \( A \)'s truth value in \( \mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)} \) is \( \mathcal{E}_{\text{mk}(S)} \).
(b) If \( A \neq L \), but \( A \) depends on \( L \) in \( \mathcal{P} \), then \( A \) also depends on \( L \) in \( \mathcal{P}^{\text{mk}(S)} \) according to Proposition 5.

(c) If \( A \neq L \) and \( A \) does not depend on \( L \), then the clauses for which \( A \) is the definition of are not affected in \( \mathcal{P}^{\text{mk}(S)} \).

2. If \( A \in I^\perp \), then for all clauses \( A \leftarrow \text{Body} \), \( \text{Body} \) is false in \( \text{lm}_L \text{wc}(\mathcal{P} \cup \mathcal{E}_S) \). We need to distinguish between three cases, to show that \( A \in J^\perp \):

(a) If \( A = L \), then the clause that determines \( A \)'s truth value in \( \mathcal{P} \cup \mathcal{E}_S \) is \( \mathcal{E}_S \). Analogously, the clause that determines \( A \)'s truth value in \( \mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)} \) is \( \mathcal{E}_{\text{mk}(S)} \).

(b) If \( A \neq L \), but \( A \) depends on \( L \) in \( \mathcal{P} \), then \( A \) also depends on \( L \) in \( \mathcal{P}^{\text{mk}(S)} \) according to Proposition 5.

(c) If \( A \neq L \) and \( A \) does not depend on \( L \), then the clauses for which \( A \) is the definition of are not affected in \( \mathcal{P}^{\text{mk}(S)} \).

It is a strict subset relation, because \( \mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)} \) states at least the truth of an additional \( \text{mk} \)-predicate that is not defined in \( \mathcal{P} \cup \mathcal{E}_S \), that is, for every \( A \in S \), \( \mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)} \models \text{lm}_L \text{wc} \text{mk}(A) \) and for every \( \mathcal{A} \in S \), \( \mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)} \not\models \text{lm}_L \text{wc} \text{mk}(A) \); but in both cases, \( \mathcal{P} \cup \mathcal{E}_S \not\models \text{lm}_L \text{wc} \text{mk}(A) \) and \( \mathcal{P} \cup \mathcal{E}_S \not\models \text{lm}_L \text{wc} \text{mk}(A) \).

**Proposition 10.** Given a program \( \mathcal{P} \), a consistent set of literals \( S \), \( \mathcal{E}_{\text{mk}(S)} \subseteq \mathcal{A}^{\text{mk}}_\mathcal{P} \) and \( \mathcal{E}_S \subseteq \mathcal{A}_\mathcal{P} \), the following holds:

\[
\text{lm}_L \text{wc}(\mathcal{P} \cup \mathcal{E}_S) = \text{lm}_L \text{wc}(\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)}) \setminus \{\text{mk}(A) \mid \text{mk}(A) \in \mathcal{A}^{\text{mk}}_\mathcal{P}\}.
\]

**Proof.** Assume \( \text{lm}_L \text{wc}(\mathcal{P} \cup \mathcal{E}_S) = \langle I^\top, I^\perp \rangle \) and \( \text{lm}_L \text{wc}(\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)}) = \langle J^\top, J^\perp \rangle \). According to Proposition 10, the following needs to hold:

\[
I^\top = J^\top \setminus \{\text{mk}(A) \mid \text{mk}(A) \in \mathcal{A}^{\text{mk}}_\mathcal{P}\} \quad \text{and} \quad I^\perp = J^\perp \setminus \{\text{mk}(A) \mid \text{mk}(A) \in \mathcal{A}^{\text{mk}}_\mathcal{P}\}.
\]

1. From Proposition 9, we know that \( \text{lm}_L \text{wc}(\mathcal{P} \cup \mathcal{E}_S) \subseteq \text{lm}_L \text{wc}(\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_S) \), which more specifically means that \( I^\top \subset J^\top \) and \( I^\perp \subset J^\perp \).

2. As mentioned in the proof of Proposition 9, the only reason for the strict subset relation is because \( \text{lm}_L \text{wc}(\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_S) \) necessarily states the truth about \( \text{mk}(A) \) for every \( L \in S \), where \( L = A \) or \( L = \mathcal{A} \), which is not stated in \( \text{lm}_L \text{wc}(\mathcal{P} \cup \mathcal{E}_S) \).

\[\square\]
B Lewis’s Axiomatic System of Counterfactuals

Lewis [5] introduced an axiomatic system of counterfactuals, reformulated in [48], describing properties of counterfactual reasoning by humans. Also, Lewis’s counterfactuals satisfy these properties, where $> \text{ is a counterfactual:}$

- **Fallacy of strengthening the antecedent** $A > B$ does not imply $A \land C > B$
- **Fallacy of transitivity** $A > B$ and $B > C$ do not imply $A > C$
- **Fallacy of contraposition** $A > B$ does not imply $\neg B > \neg A$
- **Combination of sentences** $A > B$ and $A > C$ imply $A > B \land C$

The first and the third property follow from that we use LP with non-monotonic defeasibility in our framework. For clarity we will illustrate them by examples.

**Proposition 12. (Fallacy of strengthening the antecedent)**

Let $\langle P, A_P^{mk}, CF_{S, Conc}, IC, \models_{lmwc} \rangle$ and $\langle P, A_P^{mk}, CF_{S', Conc}, IC, \models_{lmwc} \rangle$ be two counterfactual frameworks, where $S \subset S'$. The following holds:

If $CF_{S, Conc}$ is a valid counterfactual given $P$ and $IC$, then $CF_{S', Conc}$ is not necessarily a valid counterfactual given $P$ and $IC$.

Consider $CF_{\text{dry}, \text{fire}}$ again which we have shown valid wrt the program in consideration. A strengthening of the antecedent is possibly, $CF_{\text{dry}, \text{raising}, \text{fire}}$: 

*If only there had not been so many dry leaves on the forest floor, and there had been a fire-raising, then the forest fire wouldn’t have occurred.*

It is easy to see that $CF_{\text{dry}, \text{raising}, \text{fire}}$ is not valid wrt the same program.

**Proposition 13. (Fallacy of contraposition)**

Let $\langle P, A_P^{mk}, CF_{S, Conc}, IC, \models_{lmwc} \rangle$ and $\langle P, A_P^{mk}, CF_{S', Conc}, IC, \models_{lmwc} \rangle$ be two counterfactual frameworks. The following holds:

If $CF_{S, Conc}$ is a valid counterfactual given $P$ and $IC$, then $CF_{Conc, S}$ is not necessarily a valid counterfactual given $P$ and $IC$.

The contrapositive counterfactual of valid $CF_{\text{dry}, \text{fire}}$ is not valid:

*If there had been a forest fire, then there would have been dry leaves.*

**Proposition 14. (Fallacy of Transitivity)**

Let $\langle P, A_P^{mk}, CF_{S, Conc}, IC, \models_{lmwc} \rangle$, $\langle P, A_P^{mk}, CF_{S', Conc}, IC, \models_{lmwc} \rangle$ and $\langle P, A_P^{mk}, CF_{S, Conc}, IC, \models_{lmwc} \rangle$ be 3 counterfactual frameworks. The following holds:
Abductive Counterfactual Reasoning

If $CFS,Conc$ and $CFS',Conc'$ are valid counterfactuals given $\mathcal{P}$ and $\mathcal{IC}$, then $CFS,Conc'$ is not necessarily a valid counterfactual given $\mathcal{P}$ and $\mathcal{IC}$.

Let us show this by the following example from [49]:

If James Bond had been born in Russia he would have been a Communist. \hspace{1em} (CF$_1$)
If James Bond had been a Communist he would have been a traitor. \hspace{1em} (CF$_2$)
If James Bond had been born in Russia he would have been a traitor. \hspace{1em} (CF$_3$)

Counterfactual CF$_1$ is based on the background knowledge that James Bond was not born in a (former) communist country, such as Russia. Consider $\mathcal{P}$ with $\mathcal{IC} = \emptyset$:

\[
\begin{align*}
\text{communist(jbond)} & \leftrightarrow \text{born(jbond, ru)}, \\
\text{traitor(jbond)} & \leftrightarrow \text{communist(jbond) \land \neg born(jbond, ru)}, \\
\text{born(jbond, ru)} & \leftrightarrow (\perp \land \text{mk(born(jbond, ru))}) \lor \text{mk(born(jbond, ru))}, \\
\text{mk(born(jbond, ru))} & \leftrightharpoons \top.
\end{align*}
\]

The last clause is a negative fact, stating that James Bond was actually not born in Russia. In order to evaluate CF$_1$, we need to transform $\mathcal{P}$ wrt $S = \{\text{born(jbond, ru)}\}$. Accordingly, $\mathcal{E}_{mk(S)}$ is $\{\text{mk(born(jbond, ru))} \leftarrow \top\}$, where $\text{wc}(\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)})$ is:

\[
\begin{align*}
\text{communist(jbond)} & \leftrightarrow \text{born(jbond, ru)}, \\
\text{traitor(jbond)} & \leftrightarrow \text{communist(jbond) \land \neg born(jbond, ru)}, \\
\text{born(jbond, ru)} & \leftrightarrow (\perp \land \text{mk(born(jbond, ru))}) \lor \text{mk(born(jbond, ru))}, \\
\text{mk(born(jbond, ru))} & \leftrightharpoons \top.
\end{align*}
\]

Its corresponding least model of the weak completion is:

\[
\langle \{\text{mk(born(jbond, ru)), born(jbond, ru), communist(jbond)}\}, \{\text{traitor(jbond)}\}\rangle.
\]

It implies Conc and thus CF$_1$ is valid wrt $\mathcal{P}$ and $\mathcal{IC}$. For CF$_2$, $S = \text{communist(jbond)}$ and $\mathcal{E}_{\text{mk}(S)} = \{\text{mk(communist(jbond))} \leftarrow \top\}$, where $\text{Im}_{\text{wc}}(\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)})$ is:

\[
\langle \{\text{mk(communist(jbond)), communist(jbond), traitor(jbond)}\}, \{\text{born(jbond, ru)}\}\rangle,
\]

and thus also CF$_2$ is valid wrt to $\mathcal{P}$ and $\mathcal{IC}$ Let us now consider CF$_3$. The precondition is $S = \text{born(jbond, ru)}$ therefore $\mathcal{E}_{\text{mk}(S)}$ is $\{\text{mk(born(jbond, ru))} \leftarrow \top\}$. $\text{Im}_{\text{wc}}(\mathcal{P}^{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)})$ is:

\[
\langle \{\text{mk(born(jbond, ru)), born(jbond, ru), communist(jbond)}\}, \{\text{traitor(jbond)}\}\rangle,
\]

which does not entail Conc.

**Proposition 15.** (Combination of sentences)

Let $\langle \mathcal{P}, \mathcal{A}_{\text{P}}^{\text{mk}}, CFS,Conc, \mathcal{IC}, \models_{\text{Imwc}} \rangle$, and $\langle \mathcal{P}, \mathcal{A}_{\text{P}}^{\text{mk}}, CFS,Conc', \mathcal{IC}, \models_{\text{Imwc}} \rangle$ be two counterfactual frameworks. The following holds:
If $CF_{S,\text{Conc}}$ and $CF_{S,\text{Conc}'}$ are valid counterfactuals given $P$ and $IC$, then $CF_{S,(\text{Conc},\text{Conc}')}^{}$ is a valid counterfactual given $P$ and $IC$.

Proof. If $CF_{S,(\text{Conc},\text{Conc}')}^{}$ is not valid, then there is at least one literal $L \in \text{Conc} \cup \text{Conc}'$ for which $P_{\text{mk}(S)} \cup \mathcal{E}_{\text{mk}(S)} \not\models_{L_{\text{mmc}}} L$. Either $L \in \text{Conc}$ or $L \in \text{Conc}'$. Consequently, $CF_{S,\text{Conc}}$ or $CF_{S,\text{Conc}'}$ is not valid either. \qed