An Abductive Counterfactual Reasoning Approach in Logic Programming

Luís Moniz Pereira[*1]
Emmanuelle-Anna Dietz [†1,2]
Steffen Hölldobler[†2]

[1]*NOVA Laboratory for Computer Science and Informatics (NOVA LINCS)*
Departamento de Informática Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal

[2]*International Center for Computational Logic*
TU Dresden, D-01062 Dresden, Germany

submitted 1 January 2003; revised 1 January 2003; accepted 1 January 2003

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 *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.

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 (Pereira et al. 1991a; Ginsberg 1986; Gabbay et al. 2000; Bench-Capon 1989). One of the first elaborate analysis was carried out by Lewis, who employed a possible world semantics for counterfactuals (Lewis 1973). 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

[*] lmp@fct.unl.pt
[†] {dietz,sh}@iccl.tu-dresden.de

but what if it had not. Counterfactuals are sometimes called subjunctive conditionals. By *conditional* we mean a statement of the form *if condition then consequence*. A counterfactual is then a conditional of the form

$\mathcal{D}$ would have been the case, if $\mathcal{C}$ had been the case.

where the condition $\mathcal{C}$ (sometimes also referred to as antecedent or precondition) and the consequence $\mathcal{D}$ are finite and consistent sets of literals. In the sequel, we denote them as $cond(\mathcal{C}, \mathcal{D})$. Consider the example from Byrne in (Byrne 2007, 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.

Given the context, we us assume the causal relation of the conditional:

If there is a lightning and the leaves are dry, then there is a forest fire. C_1

A counterfactual we might think of could be as follows:

If only there had not been so many dry leaves on the forest floor, $(cond(\overline{dry}, \overline{ffire}))$
then the forest fire wouldn't have occurred.

Similarly to (Halpern and Hitchcock 2013), we extend this scenario with another possible reason for a forest fire, represented by the following conditional:

If there is a fire-raising, then there is a forest fire. C_2

The condition, fire-raising, expresses the act of intentionally burning something. We assume that, different to C_1, for a fire-raising to be successful, i.e., to intentionally set a forest fire, it is not necessary that the leaves be dry: 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. We assume, as for conditionals, that counterfactuals are composed of two parts: A *had condition*, which in $cond(\overline{dry}, \overline{ffire})$ corresponds to 'there had not been so many dry leaves on the forest floor' and a *would consequent*, which in $cond(\overline{dry}, \overline{ffire})$ corresponds to 'the forest fire wouldn't have occurred'. In the following we will refer to them simply as condition and consequent, respectively. A counterfactual is valid if the consequence would be true in a situation where the condition had been actually true. In order to test its validity, it is necessary to go 'a step back' and assume that the condition is actually true. For instance, $cond(\overline{dry}, \overline{ffire})$ 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 was indeed a lightning. Then, when evaluating $cond(\overline{dry}, \overline{ffire})$ we would preferably come to the conclusion that the forest fire would not have occurred, and thus, that $cond(\overline{dry}, \overline{ffire})$ is valid. Let us extend the scenario with the information that we know after reading C_2. We do not want to conclude that $cond(\overline{dry}, \overline{ffire})$ is valid anymore, as 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 (Pearl 2000), Pearl presents a theory for employing counterfactuals that includes conjectures and Bayesian networks, extensively spelled out and exemplified in (Woodward 2003). His main idea is to accept a counterfactual if its consequent is true after adding the condition hypothetically to the beliefs and making the minimal required adjustments to maintain consistency of the model. This is achieved by isolating the condition 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, were captured in LP by Baral et al. with their system P-log (Baral and Hunsaker 2007; Baral et al. 2009). P-log has been used in probabilistic moral reasoning (Anh et al. 2012) and the authors (Pereira and Saptawijaya 2016a; Pereira and Saptawijaya 2016b; Pereira and Saptawijaya 2016d) intend to employ counterfactuals in moral reasoning, as part of ongoing work using LP (Saptawijaya and Pereira 2014; Pereira and Saptawijaya 2016a).

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 non probabilistic 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 (Ramsey 1931); Lewis's maximal world-similarity one (Lewis 1973); and Rescher's systematic reconstruction of the belief system, using principles of saliency and prioritization (Rescher 2007).

Different from our LP least weak completion model approach, and its revision of logic rules negating the counterfactual premise, Ginsberg (Ginsberg 1986) 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 (Pereira and Aparício 1989) 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 (Pereira et al. 1991a). The authors of (Pereira and Saptawijaya 2016a; Pereira and Saptawijaya 2016c), inspired by our work, have defined and implemented a well-founded semantics approach to LP counterfactual reasoning with applications to morality.

2.1 Pearl's Do-Calculus

Pearl (Pearl 2000) proposes a structural theory of counterfactuals in Bayesian networks which determines the probability of a counterfactual. A counterfactual requires a hypothetical modification of the current situation. It is warranted if "the consequent follows after adding the condition hypothetically to the beliefs and the minimal required adjustments to maintain consistency of the model are made"(Pearl 2000). We briefly sketch the main idea of Pearl's well-known theory: 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. Additionally there is a set V of variables, for which each variable $V_i \in V$ is assigned a value through a function F. The probability function of every (endogenous) variable in V is uniquely determined by the instantiated background variables $u \in U$. 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 $P(Y_x = y \mid e)$ where e is any propositional evidence, can be computed by the three step process:

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

As the description of Step 1 already states, in this step the additional current evidences are abduced and the past circumstances (U) are updated accordingly; step 2 changes the past sufficiently enough for consistency, when imposing 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 afforded by the evidence e. In a nutshell, intervene to impose x and determine probability of y, things otherwise being equal, but compatible with the 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)* (Rescher 2005, 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 (Vennekens et al. 2009; Vennekens et al. 2010), 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: $\forall x(A_1 : \alpha_1) \vee \cdots \vee (A_1 : \alpha_1) \leftarrow \delta$ where δ is a first-order formula and A_i are atoms, the α_i are non-zero probabilities, and the tuple of variable x contains free variables in δ and the A_i. They can be read as follows "for each x, δ 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 α_i." (Vennekens et al. 2009) Their semantics is based on Shafer's probability trees (Shafer 1996), where each node corresponds to an interpretation for a given vocabulary. Each node in the tree is a state whose parents represent an event that causes a probabilistic transition to one of its children, which is represented by a probability distribution $\pi_{(T)}(l)$ for each leaf l. Following Pearl, an intervention is 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 $(C \mid R) \cup A$. 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 from our approach, they do not encounter the issue of conflicting laws that might ignore other ones, because first the imposed CP-law has a probability higher than 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 (Lloyd 1984; Hölldobler 2009).

3.1 Logic Programs

We restrict ourselves to propositional programs, i.e. the set of terms consists only of constants and variables. A *logic program* $\mathcal{P}$ is a finite set of clauses of the form

$$A \quad \leftarrow \quad L_1 \wedge \ldots \wedge L_n, \tag{1}$$

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. We define that, in case $n = 0$, 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 $\mathcal{P}$ 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 $\mathsf{pos}(F)$ ($\mathsf{neg}(F)$, resp.) denotes the conjunction of all positive (negative, resp.) literals occurring in F. An empty conjunction is always true, therefore, if F does not contain any literal, $\mathsf{pos}(F) = \mathsf{neg}(F) = \top$. Accordingly, it holds that $\mathsf{pos}(\top) = \mathsf{neg}(\top) = \top$. To let pos and neg also be applicable to bodies of negative facts, we define additionally $\mathsf{pos}(\bot) := \mathsf{neg}(\bot) := \top$.

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

$$p \;\leftarrow\; q \wedge \overline{r} \wedge s, \qquad q \;\leftarrow\; t, \qquad r \;\leftarrow\; \bot, \qquad s \;\leftarrow\; \top,$$

F	$\neg F$
$\top$	$\bot$
$\bot$	$\top$
U	U

$\wedge$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	U	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$

$\vee$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	$\top$	U	U
$\bot$	$\top$	U	$\bot$

$\leftarrow_{Ł}$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	$\top$	$\top$
$\bot$	$\bot$	U	$\top$

$\leftrightarrow_{Ł}$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	$\top$	U
$\bot$	$\bot$	U	$\top$

Table 1. $\top$*,* $\bot$*, and* U *denote true, false, and unknown, respectively.*

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:

$$\mathsf{pos}(q \wedge \overline{r} \wedge s) \;=\; q \wedge s, \qquad\qquad \mathsf{neg}(q \wedge \overline{r} \wedge s) \;=\; \overline{r}.$$

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

$$\mathsf{atoms}(\mathcal{P}) \;=\; \{p, q, r, s, t\}, \qquad \mathsf{def}(\mathcal{P}) \;=\; \{p, q, r, s\}, \qquad \mathsf{undef}(\mathcal{P}) \;=\; \{t\}.$$

Consider the following transformation for $\mathcal{P}$:

1. For each $A \in \mathsf{def}(\mathcal{P})$, replace all clauses of the form $A \leftarrow \mathit{Body}_1, \ldots, A \leftarrow \mathit{Body}_m$ occurring in $\mathcal{P}$ by $A \leftarrow \mathit{Body}_1 \vee \ldots \vee \mathit{Body}_m$.
2. If $A \in \mathsf{undef}(\mathcal{P})$, then add $A \leftarrow \bot$.
3. Replace all occurrences of $\leftarrow$ by $\leftrightarrow$.

The resulting set of equivalences is called the *completion* of $\mathcal{P}$ (Clark 1978). If Step 2 is omitted, then the resulting set is called the *weak completion* of $\mathcal{P}$ ($\mathsf{wc}\,\mathcal{P}$). In contrast to completed programs, the model intersection property holds for weakly completed programs (Hölldobler and Kencana Ramli 2009a). This guarantees the existence of a least model for every program.

Let $\mathcal{P}$ be a program and $p, q \in \mathsf{atoms}(\mathcal{P})$. p *depends negatively on* q wrt $\mathcal{P}$ iff $\mathcal{P}$ contains a clause of the form $p \leftarrow \mathit{Body}$ and $\overline{q}$ is in $\mathsf{neg}(\mathit{Body})$. 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 $p \leftarrow \mathit{Body}$ and q is in $\mathsf{pos}(\mathit{Body})$. p *depends on* q 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

Under *two-valued semantics*, a *two-valued interpretation* I of a program $\mathcal{P}$ is a mapping of $\mathsf{atoms}(\mathcal{P})$ 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 (Łukasiewicz 1920), for which the corresponding truth values are $\top$, $\bot$ and U, which mean *true*, *false* and *unknown*, respectively. A *three-valued interpretation* I is a mapping from $\mathsf{atoms}(\mathcal{P})$ to the 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 U. Let $I = \langle I^\top, I^\bot \rangle$ and $J = \langle J^\top, J^\bot \rangle$ be two interpretations. We define

$$I \subseteq J \quad \text{iff} \quad I^\top \subseteq J^\top \text{ and } I^\bot \subseteq J^\bot.$$

An interpretation I which maps a formula F to $\top$ under Ł-logic, is written as $I(F) = \top$. $\mathcal{M}$ is a *three-valued model* of $\mathcal{P}$ if it is a three-valued interpretation, which maps each clause occurring

Program $\mathcal{P}$		$\mathsf{lm\,wc}\,\mathcal{P}$	Well-founded Model of $\mathcal{P}$
$\mathcal{P}_1$	$= \{p \leftarrow q\}$	$\langle\emptyset,\emptyset\rangle$	$\langle\emptyset,\{p,q\}\rangle$
$\mathcal{P}_2$	$= \{p \leftarrow \neg q,\ q \leftarrow \neg p\}$	$\langle\emptyset,\emptyset\rangle$	$\langle\emptyset,\emptyset\rangle$
$\mathcal{P}_3$	$= \{p \leftarrow q,\ q \leftarrow p\}$	$\langle\emptyset,\emptyset\rangle$	$\langle\emptyset,\{p,q\}\rangle$
$\mathcal{P}_4$	$= \{p \leftarrow \neg p\}$	$\langle\emptyset,\emptyset\rangle$	$\langle\emptyset,\emptyset\rangle$

Table 2. *The notions of weak completion and well-founded semantics shown by some program examples.*

in $\mathcal{P}$ to $\top$. $\mathcal{M}$ is the *least model* of $\mathcal{P}$ iff for any other model $\mathcal{M}$ of $\mathcal{P}$ it holds that $\mathcal{M} \subseteq \mathcal{M}'$. 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 (Apt and van Emden 1982). For instance, the least fixed point of the $\mathsf{T}_\mathcal{P}$ operator ($\mathsf{lfp}\,\mathsf{T}_\mathcal{P}$) corresponds to the least two-valued model of a definite program $\mathcal{P}$. Let us define the consequence relation, $\models_{\mathsf{T}_\mathcal{P}}$, where, given a program $\mathcal{P}$ and an atom A, $\mathcal{P} \models_{\mathsf{T}_\mathcal{P}} A$ iff $A \in \mathsf{lfp}\,\mathsf{T}_\mathcal{P}$. If $\mathcal{P}$ is definite, then it has always a least model. However, this does not necessarily hold, if $\mathcal{P}$ is not definite. However this does not hold for programs that are not definite.
Hölldobler and Kencana Ramli (Hölldobler and Kencana Ramli 2009a; Hölldobler and Kencana Ramli 2009b) 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 (Dietz et al. 2012; Pereira et al. 2014; Pereira et al. 2014). 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 (Dietz et al. 2013). The least model of the weak completion of a program $\mathcal{P}$ under Łukasiewicz Semantics ($\mathsf{lm\,wc}\,\mathcal{P}$) is identical to the least fixed point of the following semantic operator, SvL, which was introduced by Stenning and van Lambalgen (Stenning and van Lambalgen 2008) for propositional programs and has been generalized for first-order programs (Hölldobler and Kencana Ramli 2009a). Let I be an interpretation and $\mathcal{P}$ be a program. Then the application of SvL to I and $\mathcal{P}$, denoted by $\Phi_\mathcal{P}(I)$, is the interpretation $J = \langle J^\top, J^\perp\rangle$, where

$$\begin{aligned} J^\top &= \{A \mid A \leftarrow \mathit{Body} \in \mathsf{def}(A,\mathcal{P}) \text{ and } I(\mathit{Body}) = \top\} \\ J^\perp &= \{A \mid \mathsf{def}(A,\mathcal{P}) \neq \emptyset \text{ and} \\ &\qquad \text{for all } A \leftarrow \mathit{Body} \in \mathsf{def}(A,\mathcal{P}) \text{ we find that } I(\mathit{Body}) = \perp\} \end{aligned}$$

The SvL operator is monotonic, which has been shown as Proposition 3.21 in (Kencana Ramli 2009). From $I = \langle\emptyset,\emptyset\rangle$, $\mathsf{lm\,wc}\,\mathcal{P}$ is computed by iterating $\Phi_\mathcal{P}$. Given a program $\mathcal{P}$ and a formula F $\mathcal{P} \models_{wcs} F$ iff $\mathsf{lm\,wc}\,\mathcal{P}(F) = \top$.

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

$$\mathcal{P} \models A \text{ iff } \mathcal{P} \models_{wcs} A.$$

Proof
According to the definition for the $\mathsf{T}_\mathcal{P}$ operator, $\mathcal{P} \models_{\mathsf{T}_\mathcal{P}} A$ iff at some moment during the fixed point iteration of $\mathsf{T}_\mathcal{P}$ there exists a clause $A \leftarrow \mathit{Body} \in \mathsf{def}(A, \mathcal{P})$ with $I(\mathit{Body}) = \top$. If this is the case, and only then, according to the definition for the SvL operator, $A \in I^\top$ at some moment during the fixed point iteration of SvL. As SvL is monotonic, A is also true in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}$. □

Note that Φ_{SvL} differs in a subtle way from the well-known Fitting operator Φ_F, introduced in (Fitting 1985): The definition of Φ_F is like that of Φ_{SvL}, except that in the specification of $J^\perp$ the first line "there exists a clause $A \leftarrow \mathit{Body} \in \mathcal{P}$ 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 Φ_{SvL} is applied instead of Φ_F, this does not hold for any interpretation I.
The correspondence between weak completion semantics and well-founded semantics (Van Gelder et al. 1991) for tight programs, i.e. those without positive loops, is shown in (Dietz et al. 2013). Table 2 shows some examples which give an intuitive idea of their similarities and differences.

3.4 Integrity Constraints

Until now, integrity constraints have not been examined in the context of the weak completion semantics. Yet, they might be useful, and therefore we will explain how we can understand them under three-valued logics and how we will deal with them. Usually, under two-valued semantics a set of *integrity constraints* $\mathcal{IC}$, contains clauses of the following form:

$$\bot \quad \leftarrow \quad \mathit{Body},$$

where *Body* is a conjunction of literals. $\mathcal{P}$ *satisfies* $\mathcal{IC}$ iff $\mathcal{P} \cup \mathcal{IC}$ is satisfiable. Under two-valued semantics a set of clauses is satisfiable if there exists a two-valued model for it. This unambiguously implies that for each clause in $\mathcal{IC}$, *Body* is mapped to false under this model.
Under three-valued semantics, there are two possible ways on how to understand integrity constraints: Either we require that the *Body* of the clause occurring in the set of integrity constraints is false under the model under consideration or that the *Body* is unknown. At first glance it might be natural to assume that the *Body* of the $\mathcal{IC}$ should be false. However, considering that we are interested in modeling human reasoning that might not deliver the desired result. Assume that we want to formalize the following conditional in a logic program:

If it rains then they will not go to the beach.

The consequence of this conditional is the negation of *they will go to the beach*. Let us assume that $beach$ denotes *they will go to the beach*. As we do not allow negative literals in the head of clauses we need to introduce an axillary atom which represents the negation of the consequence, e.g. $beach'$. The logic program $\mathcal{P}$ representing the conditional contains the following two clauses:

$$\begin{array}{lcl} beach' & \leftarrow & rain, \\ beach & \leftarrow & \neg beach', \end{array}$$

where $rain$ stands for *it rains*. The second conditional states that $beach$ will be true if $beach'$ is false. If an interpretation $\langle I^\top, I^\perp \rangle$ contains both $beach$ and $beach'$ in $I^\top$ it should be invalidated as a model of $\mathcal{P}$ in general. This can be specified by the following integrity constraint:

$$\bot \quad \leftarrow \quad beach \wedge beach',$$

which, given Table 1, implies that either $beach'$ or $beach$ has to be false. Both cannot stay unknown, even though possibly nothing is stated about the truth of them. We are not interested in finding the truth-least but the knowledge-least model, that is, both $I^\top$ and $I^\bot$ should be minimized, or, in other words, the unknown values should be maximized. Therefore, we understand integrity constraints as

$$\mathsf{U} \quad \leftarrow \quad Body,$$

instead. Since in the following we only consider whether integrity constraints are true under Łukasiewicz semantics, the body of the integrity constraint can be either false or unknown according to Table 1. For the example above we modify the integrity constraint accordingly. The $\mathcal{IC}$ is defined as

$$\mathsf{U} \quad \leftarrow \quad beach \wedge beach'.$$

This understanding that the body can be either false or unknown, is similar to the definition of the integrity constraints for the well-founded semantics in (Pereira et al. 1991b). In the sequel, $\mathcal{IC}$ refers to this kind of integrity constraints if we consider them under three-valued semantics. Note that in case we consider integrity constraints under two-valued semantics, they will necessarily have to be understood as $\bot \leftarrow Body$.

As the SvL operator cannot handle $\mathcal{IC}$s, we need to apply a two step approach. Firstly, we compute the least fixed point of the given program and, secondly, we verify whether it satisfies the requirements of the $\mathcal{IC}$s. Given an interpretation I and a set of integrity constraints $\mathcal{IC}$, *I satisfies $\mathcal{IC}$* iff all clauses in $\mathcal{IC}$ are true under I. We extend the model intersection property for all models of the weak completion that satisfy $\mathcal{IC}$ and show that if there exists a model of a weakly completed program that satisfies a set of integrity constraints, then there exists a least model of this weakly completed program that satisfies this set. First, consider the following proposition:

Proposition 2
Let $\mathcal{M}$ be a non-empty set of all models of the weak completion of program $\mathcal{P}$:

1. The intersection of all these models, $\bigcap \mathcal{M}$, is a model of the weak completion of $\mathcal{P}$.
2. This intersection is the least model of the weak completion of $\mathcal{P}$.

Proof
1. has been shown in (Hölldobler and Kencana Ramli 2009a) and 2. follows immediately from the definitions of least models and of intersection. □

Proposition 3
If there exists a model of the weak completion of program $\mathcal{P}$ that satisfies a set of integrity constraints $\mathcal{IC}$, then there exists a least model of the weak completion of $\mathcal{P}$ that satisfies $\mathcal{IC}$.

Proof
Assume that there exists a least model of the weak completion of a program $\mathcal{P}$ that satisfies the set of integrity constraints $\mathcal{IC}$. Assume that $\mathcal{M}$ is the set of all models of the weak completion of $\mathcal{P}$, that satisfy $\mathcal{IC}$. According to Proposition 2.1, the intersection of these models is also a model which satisfies $\mathcal{IC}$. According to Proposition 2.2, it is also the least model of the weak completion of $\mathcal{P}$. □

3.5 Abduction

In this section, we will mainly focus on three-valued abduction and show the correspondence to two-valued abduction. *An abductive framework* (Kakas et al. 1993) is a quadruple $\langle \mathcal{P}, \mathcal{A}, \mathcal{IC}, \models \rangle$, consisting of a program $\mathcal{P}$ as knowledge base, a finite set of abducibles $\mathcal{A}_{\mathcal{P}}$, a finite set of integrity constraints $\mathcal{IC}$, and a consequence relation $\models$.

3.5.1 Two-valued Abduction

A *two-valued abductive framework* (Kakas et al. 1993) is the quadruple $\langle \mathcal{P}, \mathcal{A}_{2,\mathcal{P}}, \mathcal{IC}, \models_{\mathsf{T}_{\mathcal{P}}} \rangle$, where $\mathcal{P}$ is definite and $\mathcal{A}_{2,\mathcal{P}}$ is defined as

$$\{A \leftarrow \top \mid A \in \mathsf{undef}(\mathcal{P})\}.$$

An *Observation* $\mathcal{O}$ is a non-empty set of literals that we want to explain. Note that in a two-valued abductive framework the clauses in $\mathcal{IC}$ are of the form $\bot \leftarrow$ *Body*.

Definition 1
Let $\langle \mathcal{P}, \mathcal{A}_{2,\mathcal{P}}, \mathcal{IC}, \models_{\mathsf{T}_{\mathcal{P}}} \rangle$ be a two-valued abductive framework where $\mathcal{P}$ satisfies $\mathcal{IC}$, $\mathcal{E} \subseteq \mathcal{A}_{2,\mathcal{P}}$ and $\mathcal{O}$ is an observation.

$\mathcal{O}$ is *two-valued explained by $\mathcal{E}$ given $\mathcal{P}$ and $\mathcal{IC}$* iff
$\mathcal{P} \cup \mathcal{E} \models_{\mathsf{T}_{\mathcal{P}}} \mathcal{O}$ and $\mathcal{P} \cup \mathcal{E} \models_{\mathsf{T}_{\mathcal{P}}} \mathcal{IC}$.
$\mathcal{O}$ is *two-valued explainable given $\mathcal{P}$ and $\mathcal{IC}$* iff there exists an $\mathcal{E}$
such that $\mathcal{O}$ is two-valued explained by $\mathcal{E}$ given $\mathcal{P}$ and $\mathcal{IC}$.

Note that if $\mathcal{P} \models_{\mathsf{T}_{\mathcal{P}}} \mathcal{O}$ then $\mathcal{E}$ is empty. Normally, only set inclusion minimal (or otherwise preferred) explanations are considered. We assume henceforth that explanations are minimal, that means there is no other explanation $\mathcal{E}' \subset \mathcal{E}$ for $\mathcal{O}$. Someone might possibly think of another preference criterion instead.

3.5.2 Three-valued Abduction

Similarly, for the three-valued semantics considered here, for which we have defined a *three-valued abductive framework* as a quadruple $\langle \mathcal{P}, \mathcal{A}, \mathcal{IC}, \models_{wcs} \rangle$, consisting of a program $\mathcal{P}$ as knowledge base, a set of abducibles $\mathcal{A}$, a set of integrity constraints $\mathcal{IC}$, and the logical consequence relation $\models_{wcs}$. Again, an *Observation* $\mathcal{O}$ is a non-empty set of literals. As we deal with the weak completion semantics, abducibles may now not only be positive facts but can also take the form of negative facts, and otherwise they remain unknown. Therefore, the set of abducibles $\mathcal{A}_{\mathcal{P}}$ for three-valued abduction is extended with the corresponding negative facts

$$\{A \leftarrow \top \mid A \in \mathsf{undef}(\mathcal{P})\} \quad \cup \quad \{A \leftarrow \bot \mid A \in \mathsf{undef}(\mathcal{P})\}$$

Proposition 4
Given a definite program $\mathcal{P}$, the following holds:

$$\text{If } \{A \leftarrow \top\} \subseteq \mathcal{A}_{2,\mathcal{P}}, \text{ then } \{A \leftarrow \top, A \leftarrow \bot\} \subseteq \mathcal{A}_{\mathcal{P}}$$

Proof
This follows immediately from the definitions for $\mathcal{A}_{2,\mathcal{P}}$ and $\mathcal{A}_{\mathcal{P}}$. □

Definition 2
Let $\langle \mathcal{P}, \mathcal{A}_{\mathcal{P}}, \mathcal{IC}, \models_{wcs} \rangle$ be a three-valued abductive framework where $\mathcal{P}$ satisfies $\mathcal{IC}$, $\mathcal{E} \subseteq \mathcal{A}_{\mathcal{P}}$ and $\mathcal{O}$ is an observation.

$\mathcal{O}$ is *three-valued explained by $\mathcal{E}$ given $\mathcal{P}$ and $\mathcal{IC}$* iff
$\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{O}$ and $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{IC}$.
$\mathcal{O}$ is *three-valued explainable given $\mathcal{P}$ and $\mathcal{IC}$* iff there exists an $\mathcal{E}$
such that $\mathcal{O}$ is three-valued explained by $\mathcal{E}$ given $\mathcal{P}$ and $\mathcal{IC}$.

In abduction, we distinguish between *credulous* and *skeptical reasoning*. Credulous reasoning means that there exists at least one model which entails the observation to be explained. Skeptical reasoning demands that every model of the program entails the observation.

F *follows skeptically from $\mathcal{P}$, $\mathcal{IC}$ and $\mathcal{O}$* iff $\mathcal{O}$ can be three-valued explainable given $\mathcal{P}$ and $\mathcal{IC}$,
and for all $\mathcal{E}$ for $\mathcal{O}$ it holds that $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$.
F *follows credulously from $\mathcal{P}$, $\mathcal{IC}$ and $\mathcal{O}$* iff there exists a $\mathcal{E}$ for $\mathcal{O}$
and it holds that $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$.

Three-valued abduction is illustrated in the following example: Assume that $\mathcal{IC} = \emptyset$ and consider the following three clauses in $\mathcal{P}$:

$$\begin{array}{lcl} p & \leftarrow & \neg q \wedge r \wedge t, \\ p & \leftarrow & \neg s \wedge r, \\ t & \leftarrow & \top, \end{array}$$

together with observation $\mathcal{O} = \{p\}$. The set of abducibles $\mathcal{A}_{\mathcal{P}}$, in the three-valued abductive framework $\langle \mathcal{P}, \mathcal{A}_{\mathcal{P}}, \mathcal{IC}, \models_{wcs} \rangle$, is

$$\begin{array}{lclclclclcl} q & \leftarrow & \top, & \quad & r & \leftarrow & \top, & \quad & s & \leftarrow & \top, \\ q & \leftarrow & \bot, & & r & \leftarrow & \bot, & & s & \leftarrow & \bot, \end{array}$$

for which there are the following two minimal explanations for $\mathcal{O}$:

$$\mathcal{E}_{rq} = \{r \leftarrow \top, \quad q \leftarrow \bot\} \qquad \text{and} \qquad \mathcal{E}_{sr} = \{s \leftarrow \bot, \quad r \leftarrow \top\}.$$

As r follows from all minimal explanations, it follows skeptically from $\mathcal{P}$ and $\mathcal{O}$, whereas $\neg q$ and $\neg s$ only follow credulously.
Note that in the case the abducibles are not abduced as positive or negative facts, they stay unknown in the least model of the weak completion. If we do not want to make each undefined atom an abducible, i.e. we want to allow for unknown, non-abducible susceptible knowledge, we can simply add the clause $A \leftarrow A$ for the atom under consideration.

3.5.3 Correspondence

Proposition 5
Given a two-valued abductive framework $\langle \mathcal{P}, \mathcal{A}_{2,\mathcal{P}}, \mathcal{IC}, \models_{\top_{\mathcal{P}}} \rangle$, a three-valued abductive framework $\langle \mathcal{P}, \mathcal{A}_{\mathcal{P}}, \mathcal{IC}, \models_{wcs} \rangle$, where $\mathcal{P}$ is definite, $\mathcal{E} \subseteq \mathcal{A}_{2,\mathcal{P}}$ and observation $\mathcal{O}$ is a non-empty set of literals. The following holds:

1. If $\mathcal{E}$ is a two-valued explanation for $\mathcal{O}$ given $\mathcal{P}$ and $\mathcal{IC}$
then $\mathcal{E}$ is an explanation for $\mathcal{O}$ given $\mathcal{P}$ and $\mathcal{IC}$.
2. If $\mathcal{O}$ is two-valued explained given $\mathcal{P}$ and $\mathcal{IC}$
then $\mathcal{O}$ is three-valued explained given $\mathcal{P}$ and $\mathcal{IC}$.

Proof
(2) follows from (1), so we show that (1) holds. Let us assume that $\mathcal{E}$ is a two-valued explanation for $\mathcal{O}$ given $\mathcal{P}$ and $\mathcal{IC}$, then $\mathcal{P} \cup \mathcal{E} \models_{\mathsf{T}_\mathcal{P}} \mathcal{O}$ and $\mathcal{P} \cup \mathcal{E} \models_{\mathsf{T}_\mathcal{P}} \mathcal{IC}$. To show: $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{O}$ and $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{IC}$.

1. $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{O}$ follow immediately from $\mathcal{P} \cup \mathcal{E} \models_{\mathsf{T}_\mathcal{P}} \mathcal{O}$ and Proposition 1.
2. $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{IC}$ means that $\mathsf{lm}_2(\mathcal{P} \cup \mathcal{E} \cup \mathcal{IC})$ is satisfiable. This implies that the body of all clauses in $\mathcal{IC}$ are mapped to false in $\mathsf{lm}_2(\mathcal{P} \cup \mathcal{E})$. If they were true in $\mathsf{lm}\,\mathsf{wc}\,(\mathcal{P} \cup \mathcal{E})$, then, according to Proposition 1, they would also have to be true in $\mathsf{lm}_2(\mathcal{P} \cup \mathcal{E})$. Therefore, the body of all clauses in $\mathcal{IC}$ are false in $\mathsf{lm}\,\mathsf{wc}\,(\mathcal{P} \cup \mathcal{E})$. Accordingly, $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{IC}$.

□

The other direction does not hold. Consider $\mathcal{P}$:

$$p \leftarrow q$$

and observation $\mathcal{O} = \{\neg p\}$. Its three-valued explanation is $\mathcal{E} = \{q \leftarrow \bot\}$, where $\mathcal{E} \in \mathcal{A}_\mathcal{P}$. However, $\mathcal{E} \notin \mathcal{A}_{2,\mathcal{P}}$ and therefore it cannot be a two-valued explanation for $\mathcal{O}$.
Since in the following we will mainly consider three-valued abduction, we implicitly assume all the abductive frameworks and explanations to be three-valued, if not explicitly stated otherwise.

4 Abductive Counterfactual Reasoning

Based on Pearl's theory, Sloman extensively clarifies and discusses the distinction between causal and counterfactual reasoning in (Sloman 2005). In both cases, a specific relation of cause and effect is described. Recall that a counterfactual statement is of the form

$$\mathcal{D} \textit{ would have been the case, if } \mathcal{C} \textit{ had been the case.} \qquad (cond(\mathcal{C}, \mathcal{D}))$$

where $\mathcal{C}$ and $\mathcal{D}$ are finite and consistent sets of literals. 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 $\mathcal{P}$, containing the following clause:

$$beach \quad \leftarrow \quad \overline{rain},$$

whose least model of the weak completion is $\langle \emptyset, \emptyset \rangle$, because $\overline{rain}$ is unknown. We conjecture the following counterfactual:

She would have gone to the beach, if it hadn't rained. (i.e. $cond(b, \overline{r})$)

Let us impose the hypothetical truth value of the condition of $cond(b, \overline{r})$ by simply adding $\overline{r}$ as a negative fact, i.e. $rain \leftarrow \bot$, to $\mathcal{P}$. $\mathcal{P}$ contains now the following two clauses:

$$beach \quad \leftarrow \quad \overline{rain}, \qquad\qquad rain \quad \leftarrow \quad \bot.$$

The corresponding least model of the weak completion is $\langle \{beach\}, \{rain\} \rangle$, and indeed $beach$ is true. Accordingly, $cond(b, \overline{r})$ seems to be valid wrt $\mathcal{P}$. But assume that initially $rain$ is actually true. This is represented by a program containing the following two clauses:

$$beach \quad \leftarrow \quad \overline{rain}, \qquad\qquad rain \quad \leftarrow \quad \top.$$

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

$$beach \leftarrow \overline{rain}, \qquad rain \leftarrow \top, \qquad rain \leftarrow \bot.$$

Consider its weak completion:

$$beach \leftrightarrow \overline{rain}, \qquad rain \leftrightarrow \top \vee \bot.$$

The corresponding least model of the weak completion is $\langle\{rain\}, \{beach\}\rangle$ and does not imply $beach$: It seems that imposing $\overline{r}$ implies more than simply adding the negative fact $rain \leftarrow \bot$. In the following, we present an approach that allows for real interventions and permits for adequate counterfactual evaluation.

4.1 The mk *Transformation*

Differently from abduction, in counterfactual reasoning the condition might already have a clause generating a truth value in $\mathcal{P}$, which means that we cannot ensure that it is in the set of abducibles, $\mathcal{A}_\mathcal{P}$, because this set only contains facts about undefined atoms in $\mathcal{P}$.
For this reason, we introduce a reserved abducibles constructor $\mathsf{mk}(A)$, called "make", for all atoms $A \in \mathsf{atoms}(\mathcal{P})$, a constructor which has no clauses in $\mathcal{P}$ and hence is undefined. The set of mk-abducibles $\mathcal{A}^{\mathsf{mk}}$ is:

$$\{\mathsf{mk}(A) \leftarrow \top \mid \mathcal{P} \not\models_{wcs} A\} \quad \cup \quad \{\mathsf{mk}(A) \leftarrow \bot \mid \mathcal{P} \not\models_{wcs} \overline{A}\}.$$

$\mathcal{A}^{\mathsf{mk}}$ contains all positive (negative) facts about all $A \in \mathsf{atoms}(\mathcal{P})$ if A is not already true (false) in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}$, i.e. A can only be either unknown or false (true) in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}$.
If the condition states that *if A had been true* or *if A had been false*, we abduce

$$\mathsf{mk}(A) \leftarrow \top \qquad \text{or} \qquad \mathsf{mk}(A) \leftarrow \bot,$$

respectively. If we abduce $\mathsf{mk}(A) \leftarrow \top$ or $\mathsf{mk}(A) \leftarrow \bot$, we say that we mk-abduce $A \leftarrow \top$ or $A \leftarrow \bot$, respectively. Note that by the definition of $\mathcal{A}^{\mathsf{mk}}$, 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 already false (or true) in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}$. Our wish is to "counter abduce", which complies with the natural language understanding of counterfactuals (Hewings 2013).

Proposition 6
Given a program $\mathcal{P}$, $\mathcal{A}$ and $\mathcal{A}^{\mathsf{mk}}$, the following holds:

$$\text{If } \{A \leftarrow \top, A \leftarrow \bot\} \subseteq \mathcal{A}_\mathcal{P} \text{ then } \{\mathsf{mk}(A) \leftarrow \top, \mathsf{mk}(A) \leftarrow \bot\} \subseteq \mathcal{A}^{\mathsf{mk}}.$$

Proof
This follows immediately from the definitions for $\mathcal{A}_\mathcal{P}$ and $\mathcal{A}^{\mathsf{mk}}$. □

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

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

Given a set of literals S, consider the following transformation for $\mathcal{P}$:

1. For each $A, \neg A \in S$, if $A \in \mathsf{def}(\mathcal{P})$, replace all clauses $A \leftarrow \mathit{Body}$ occurring in $\mathcal{P}$ by $A \leftarrow \mathit{Body} \wedge \mathsf{mk}(A)$.

2. For each $A, \neg A \in S$, add $A \leftarrow \mathsf{mk}(A)$.

The resulting program is denoted $\mathcal{P}^{\mathsf{mk}(S)}$. Because we consider logic programs under the weak completion, the added clause will have the form of $A \leftrightarrow \mathsf{mk}(A) \vee \ldots$ in $\mathsf{wc}\,\mathcal{P}^{\mathsf{mk}(S)}$. The idea is that subsequently $\mathsf{mk}(A)$ can defeat all the rules for A if made false, and can impose A if made true (cf. Proposition 10). Let us exemplify the transformation with the simple program for q, given above. Given that $S = \{q\}$, $\mathcal{P}^{\mathsf{mk}(S)}$ is:

$$q \leftarrow \top \wedge \mathsf{mk}(q), \qquad\qquad q \leftarrow \mathsf{mk}(q).$$

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

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

Proposition 7
Given a program $\mathcal{P}$ and a set of literals S, the following holds:

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

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

This non elimination will be important if we want to reason about the program or its transformation, say for prducing justifications, for debugging, for meta-interpretation, for applying preferences, or for deep revision (cf. Section 5). These though are topics beyond the scope of the present paper.

Proposition 8
Given a program $\mathcal{P}$ and a set of literals S, the following holds:

Every $L \in S$ is unknown in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}^{\mathsf{mk}(S)}$.

The proof is in Appendix A.

Proposition 9
Given a program $\mathcal{P}$ and a set of literals S, the following holds:

$$\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}^{\mathsf{mk}(S)} \subseteq \mathsf{lm}\,\mathsf{wc}\,\mathcal{P}.$$

Sketch of proof
After the transformation no new facts are added to the program, and, consequently no additional facts will be in its least model. The possible subset relation is because according to Proposition 8, every $L \in S$ is unknown in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}^{\mathsf{mk}(S)}$. Consequently, every retrieved fact in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}$ that depends on L possibly becomes unknown in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}^{\mathsf{mk}(S)}$. □

The following example shows the intuition behind Proposition 9. $\mathcal{P}$ is:

$$\begin{array}{lllllll} r & \leftarrow & p \wedge q, & \qquad & p & \leftarrow & \top, \\ s & \leftarrow & p \wedge \neg q, & & q & \leftarrow & \top. \end{array}$$

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

$$\begin{array}{lllllllllll} r & \leftarrow & p \wedge q, & \qquad & p & \leftarrow & \top \wedge \mathsf{mk}(p), & \qquad & p & \leftarrow & \mathsf{mk}(p), \\ s & \leftarrow & p \wedge \neg q, & & q & \leftarrow & \top. & & & & \end{array}$$

Its weak completion is

$$\begin{array}{rclcrcl} r & \leftrightarrow & q \wedge p, & \quad & p & \leftrightarrow & \mathsf{mk}(p) \vee (\top \wedge \mathsf{mk}(p)), \\ s & \leftrightarrow & \neg q \wedge p, & & q & \leftrightarrow & \top. \end{array}$$

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

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

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

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

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

$$\forall L \in S : \; \mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)} \models_{wcs} L.$$

Proposition 11
Given a program $\mathcal{P}$, a consistent set of literals S, $\mathcal{E}_{\mathsf{mk}(S)} \subseteq \mathcal{A}^{\mathsf{mk}}$
and $\mathcal{E}_S \subset \mathcal{A}_{\mathcal{P}}$, the following holds:

$$\mathsf{lm}\,\mathsf{wc}\,(\mathcal{P} \cup \mathcal{E}_S) \subset \mathsf{lm}\,\mathsf{wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)}).$$

Proposition 12
Given a program $\mathcal{P}$, a consistent set of literals S, $\mathcal{E}_{\mathsf{mk}(S)} \subseteq \mathcal{A}^{\mathsf{mk}}$ and $\mathcal{E}_S \subset \mathcal{A}_{\mathcal{P}}$, the following holds:

$$\mathsf{lm}\,\mathsf{wc}\,(\mathcal{P} \cup \mathcal{E}_S) = \mathsf{lm}\,\mathsf{wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)}) \setminus \{\mathsf{mk}(A) \mid \mathsf{mk}(A) \in \mathcal{A}^{\mathsf{mk}}\}.$$

The proofs for Proposition 10, 11 and 12 are in Appendix A.

4.2 Evaluation

A counterfactual framework is a tuple, $\langle \mathcal{P}, \mathcal{A}_{\mathcal{P}}, \mathcal{A}^{\mathsf{mk}}, \mathcal{IC}, \models_{wcs} \rangle$, consisting of a program $\mathcal{P}$, a set of abducibles $\mathcal{A}_{\mathcal{P}}$, a set of mk-abducibles $\mathcal{A}^{\mathsf{mk}}$, a set of integrity constraints $\mathcal{IC}$, and the logical consequence relation $\models_{wcs}$. Differently from abduction defined in Section 3.5, the observation $\mathcal{O}$ is now allowed to be empty, for counterfactuals may be evaluated without extra observations. In this case, its explanation $\mathcal{E}$ is empty as well. On the other hand, the counterfactual statement in consideration, $cond(\mathcal{C}, \mathcal{D})$, can neither have empty $\mathcal{C}$ nor empty $\mathcal{D}$.

Definition 3
Let $\langle \mathcal{P}, \mathcal{A}_{\mathcal{P}}, \mathcal{A}^{\mathsf{mk}}, \mathcal{IC}, \models_{wcs} \rangle$ be a counterfactual framework where $\mathcal{E}_{\mathsf{mk}(\mathcal{C})} \subset \mathcal{A}^{\mathsf{mk}}$. Assume that $\mathcal{O}$ is explained by $\mathcal{E} \subset \mathcal{A}_{\mathcal{P}}$ given $\mathcal{P}$ and $\mathcal{IC}$ where $\mathcal{E}$ is consistent with $\mathcal{C}$.[1]

$cond(\mathcal{C}, \mathcal{D})$ is valid given $\mathcal{P} \cup \mathcal{E}$, $\mathcal{O}$ and $\mathcal{IC}$ iff

$$\mathcal{P} \cup \mathcal{E} \not\models_{wcs} \mathcal{D},\; (\mathcal{P} \cup \mathcal{E})^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})} \models_{wcs} \mathcal{D}, \text{and } \mathsf{lm}\,\mathsf{wc}\,((\mathcal{P} \cup \mathcal{E})^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})}) \text{ satisfies } \mathcal{IC}.$$

[1] $\mathcal{E}$ is consistent with $\mathcal{C}$ if we do not have $A \leftarrow \top \in \mathcal{E}$ and $\neg A \in \mathcal{C}$ or $A \leftarrow \bot \in \mathcal{E}$ and $A \in \mathcal{C}$.

By including $\mathcal{O}$ it allows us to set the counterfactual in a context in which some external observations might be relevant. Different than in (Dietz et al. 2015b), we do not simply allow an automated abductive procedure to explain the condition of the counterfactual. Instead we opt for the possibility to also fix exogenous information manually from the start, by employing $\mathcal{O}$. This allows us to determine the external situation in which a counterfactual is evaluated and to compare different counterfactuals in the same situation (cf. examples in Section 4.3). The approach presented in (Dietz et al. 2015b) cannot guarantee these fixed exogenous observations, as possibly conditions of different counterfactuals might abduce different explanations. Accordingly, the situation in which they are evaluated, changes and they are incomparable. The fixed observations also allow us to keep some information unknown and not accidentally abduce additional facts with respect to the counterfactual. Section 4.3 below motivates by examples the need for this abductive step in counterfactual reasoning.

Theorem 13
Given an abductive framework $\langle \mathcal{P}, \mathcal{A}_\mathcal{P}, \mathcal{IC}, \models_{wcs} \rangle$, where $\mathcal{O}$ is an observation and a counterfactual framework $\langle \mathcal{P}, \mathcal{A}_\mathcal{P}, \mathcal{A}^{\mathsf{mk}}, \mathcal{IC}, \models_{wcs} \rangle$, where $\mathcal{E} \subset \mathcal{A}_\mathcal{P}$, $\mathcal{E}_{\mathsf{mk}(\mathcal{C})} \subset \mathcal{A}^{\mathsf{mk}}$ and observation $\mathcal{O} = \mathcal{D}$. The following holds:

If $\mathcal{E}$ is an explanation for $\mathcal{O}$ given $\mathcal{P}$ and $\mathcal{IC}$,
then $cond(\mathcal{C}, \mathcal{D})$ is a valid counterfactual given $\mathcal{P}$ and $\mathcal{IC}$.

Proof
Let us assume that $\mathcal{E}$ is an explanation for $\mathcal{O}$ given $\mathcal{P}$ and $\mathcal{IC}$, that is, $\mathcal{P} \not\models_{wcs} \mathcal{O}$, $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{O}$ and $\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E})$ satisfies $\mathcal{IC}$.

To show: $\mathcal{P} \not\models_{wcs} \mathcal{D}$, $\mathcal{P}^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})} \models_{wcs} \mathcal{D}$, and
$\mathsf{lm\,wc}\,(\mathcal{P}^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})})$ satisfies $\mathcal{IC}$.

1. $\mathcal{P} \not\models_{wcs} \mathcal{D}$ follows from $\mathcal{P} \not\models_{wcs} \mathcal{O}$ and because $\mathcal{O} = \mathcal{D}$.
2. $\mathcal{P}^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})} \models_{wcs} \mathcal{O}$ follows from $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{O}$ and Proposition 11.
3. $\mathsf{lm\,wc}\,(\mathcal{P}^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})})$ satisfies $\mathcal{IC}$, follows from $\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E})$ satisfies $\mathcal{IC}$, by Proposition 12 and because $\mathcal{IC}$ does not include mk-predicates.

□

4.3 Examples

4.3.1 The Forest Fire

Consider the counterfactual from the introduction

If only there had not been so many dry leaves on the forest floor,
then the forest fire wouldn't have occurred. $(cond(\overline{dry}, \overline{\mathit{ffire}}))$

Assume that $\mathcal{IC} = \emptyset$ and $\mathcal{P}$ is:

$$\begin{array}{lcl@{\qquad\qquad}lcl} \mathit{forest\text{-}fire} & \leftarrow & \mathit{lightning} \wedge \overline{ab}, & \mathit{lightning} & \leftarrow & \top, \\ ab & \leftarrow & \overline{\mathit{dry\text{-}leaves}}, & \mathit{dry\text{-}leaves} & \leftarrow & \top. \end{array}$$

We identify that

$$\mathcal{C} \;=\; \{\overline{\mathit{dry\text{-}leaves}}\} \qquad \text{and} \qquad \mathcal{D} \;=\; \{\overline{\mathit{forest\text{-}fire}}\}.$$

$\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}$ is $\langle\{\textit{dry-leaves}, \textit{lightning}, \textit{forest-fire}\}, \{ab\}\rangle$, which satisfies the first requirement of Definition 3. The transformation of $\mathcal{P}$ wrt $\mathcal{C}$ is $\mathcal{P}^{\mathsf{mk}(\mathcal{C})}$:

$$\begin{array}{lcllcl} \textit{forest-fire} & \leftarrow & \textit{lightning} \wedge \overline{ab}, & \textit{lightning} & \leftarrow & \top, \\ ab & \leftarrow & \overline{\textit{dry-leaves}}, & \textit{dry-leaves} & \leftarrow & \top \wedge \mathsf{mk}(\textit{dry-leaves}), \\ & & & \textit{dry-leaves} & \leftarrow & \mathsf{mk}(\textit{dry-leaves}), \end{array}$$

for which the weak completion together with $\mathcal{E}_{\mathsf{mk}(\mathcal{C})} = \{\mathsf{mk}(\textit{dry-leaves}) \leftarrow \bot\}$ is

$$\begin{array}{lcllcl} \textit{forest-fire} & \leftrightarrow & \textit{lightning} \wedge \overline{ab}, & \textit{lightning} & \leftrightarrow & \top, \\ ab & \leftrightarrow & \overline{\textit{dry-leaves}}, & \mathsf{mk}(\textit{dry-leaves}) & \leftrightarrow & \bot, \\ \textit{dry-leaves} & \leftrightarrow & (\top \wedge \mathsf{mk}(\textit{dry-leaves})) \vee \mathsf{mk}(\textit{dry-leaves}). & & & \end{array}$$

$\mathsf{lm}\,\mathsf{wc}\,(\mathcal{P}^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})})$ is $\langle\{\textit{lightning}, ab\}, \{\mathsf{mk}(\textit{dry-leaves}), \textit{dry-leaves}, \textit{forest-fire}\}\rangle$ and indeed entails $\mathcal{D}$. Let us extend $\mathcal{P}$ by a clause that represents (C_2):

$$\textit{forest-fire} \leftarrow \textit{fire-raising}.$$

$\mathcal{C}$ and $\mathcal{D}$ stay the same. Transformed $\mathcal{P}^{\mathsf{mk}(\mathcal{C})}$ together with $\mathcal{E}_{\mathsf{mk}(\mathcal{C})}$ is:

$$\begin{array}{lcl} \textit{forest-fire} & \leftarrow & \textit{lightning} \wedge ab_1, \\ \textit{forest-fire} & \leftarrow & \textit{fire-raising} \wedge ab_2, \\ ab_1 & \leftarrow & \textit{dry-leaves}, \\ ab_2 & \leftarrow & \bot, \\ \textit{lightning} & \leftarrow & \top, \\ \textit{dry-leaves} & \leftarrow & \top \wedge \mathsf{mk}(\textit{dry-leaves}), \\ \mathsf{mk}(\textit{dry-leaves}) & \leftarrow & \bot. \end{array}$$

The corresponding weak completion is:

$$\begin{array}{lcl} \textit{forest-fire} & \leftrightarrow & (\textit{lightning} \wedge ab_1) \vee (\textit{fire-raising} \wedge ab_2), \\ ab_1 & \leftrightarrow & \textit{dry-leaves}, \\ ab_2 & \leftrightarrow & \bot, \\ \textit{lightning} & \leftrightarrow & \top, \\ \textit{dry-leaves} & \leftrightarrow & \top \wedge \mathsf{mk}(\textit{dry-leaves}), \\ \mathsf{mk}(\textit{dry-leaves}) & \leftrightarrow & \bot, \end{array}$$

and its least model of its weak completion is:

$$\langle\{\textit{lightning}, ab_1\}, \{\textit{dry-leaves}, \mathsf{mk}(\textit{dry-leaves}), ab_2\}\rangle.$$

It does not state any truth about *forest-fire*, because *fire-raising* is unknown. $\mathcal{E}_{\mathsf{mk}(\mathcal{C})}$ does not counterfactually explain $\mathcal{D}$, thus $cond(\overline{dry}, \overline{ffire})$ is not a valid counterfactual.

Note that $cond(\overline{dry}, \overline{ffire})$ is not valid only because the weak completion semantics adopts an open world assumption on undefined atoms. Under well-founded and stable model semantics *fire-raising* would have been assumed false, which entails $\neg$*forest-fire* to be in the well-founded model. Accordingly, under well-founded semantics we would conclude $cond(\overline{dry}, \overline{ffire})$ 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) (C_1), or by a fire-raising (C_2), the appropriate answer to, whether $cond(\overline{dry}, \overline{ffire})$ 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, *fire-raising* would need to be explicitly declared unknown, say by means of *fire-raising* $\leftarrow \mathsf{U}$, with reserved atom U defined as $\mathsf{U} \leftarrow \neg\mathsf{U}$. 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 $\mathcal{IC}$:

$$\mathsf{U} \leftarrow \textit{fire-raising} \wedge \textit{lightning}.$$

From the knowledge imparted by this $\mathcal{IC}$ and that *lightning* is true, we assume the background observation expressed by $\mathcal{O} = \{\neg \textit{fire-raising}\}$, since abducibles are two-valued and thus *fire-raising* can only be false, to satisfy $\mathcal{IC}$. Its corresponding only explanation is $\mathcal{E}$:

$$\textit{fire-raising} \quad \leftarrow \quad \bot.$$

According to Definition 3 we need to transform by $\mathcal{P}^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}$:

$$\begin{array}{lll}
\textit{forest-fire} & \leftarrow & \textit{lightning} \wedge \textit{dry-leaves}, \\
\textit{lightning} & \leftarrow & \top, \\
\textit{forest-fire} & \leftarrow & \textit{fire-raising}, \\
\textit{dry-leaves} & \leftarrow & \top \wedge \mathsf{mk}(\textit{dry-leaves}), \\
\mathsf{mk}(\textit{dry-leaves}) & \leftarrow & \bot, \\
\textit{fire-raising} & \leftarrow & \bot.
\end{array}$$

Its weak completion is:

$$\begin{array}{lll}
\textit{forest-fire} & \leftrightarrow & (\textit{lightning} \wedge \textit{dry-leaves}) \vee \textit{fire-raising}, \\
\textit{lightning} & \leftrightarrow & \top, \\
\textit{dry-leaves} & \leftrightarrow & \top \wedge \mathsf{mk}(\textit{dry-leaves}), \\
\mathsf{mk}(\textit{dry-leaves}) & \leftrightarrow & \bot, \\
\textit{fire-raising} & \leftrightarrow & \bot.
\end{array}$$

The least model of the weak completion is:

$$\langle \{\textit{lightning}\}, \{\textit{dry-leaves}, \mathsf{mk}(\textit{dry-leaves}), \textit{fire-raising}, \mathsf{mk}(\textit{fire-raising}), \textit{forest-fire}\} \rangle,$$

which indeed implies $\mathcal{D}$. By Definition 3, $cond(\overline{dry}, \overline{\textit{ffire}})$ is valid given $\mathcal{O}$, $\mathcal{P}$ and $\mathcal{IC}$. 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 $cond(\overline{dry}, \overline{\textit{ffire}})$ with respect to extended $\mathcal{P}$:

Given that there had not been a fire-raising,

If only there had not been so many dry leaves on the forest floor,

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 *dry-leaves* ← *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.

4.3.2 Kennedy

Let us consider yet another example from (Pearl 2011), given the following two statements:

$$\textit{If Oswald hadn't killed Kennedy, someone else would have.} \qquad (\mathit{cond}_{\overline{os},se}\ )$$

$$\textit{If Oswald didn't kill Kennedy, someone else did.} \qquad (\mathit{cond}'_{\overline{os},se}\ )$$

The first statement is a counterfactual whereas the second one is an indicative conditional. The difference on how we understand them, is that, $\mathit{cond}_{\overline{os},se}$ asks for revising not only that Oswald hadn't shot Kennedy, but its attending consequences too, namely that Kennedy was killed. In contrast to this interpretation of the counterfactual, the other one, $\mathit{cond}'_{\overline{os},se}$, implies that Kennedy was actually killed. According to (Pearl 2011), the majority of the people rejects the first but accepts the second statement. Let us evaluate $\mathit{cond}_{\overline{os},se}$:

$$\mathcal{C} \;=\; \{\overline{\mathit{oswald\_shot}}\} \quad \text{and} \quad \mathcal{D} \;=\; \{\mathit{someone\_else\_shot}\},$$

where $\mathcal{P}$ is:

$$\begin{array}{lcl}
\mathit{kennedy\_died} & \leftarrow & \mathit{oswald\_shot}, \\
\mathit{kennedy\_died} & \leftarrow & \mathit{someone\_else\_shot}, \\
\mathit{oswald\_shot} & \leftarrow & \top.
\end{array}$$

The $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}$ is $\langle\{\mathit{oswald\_shot}, \mathit{kennedy\_died}\}, \emptyset\rangle$, which complies with the first requirement in Definition 3. $\mathcal{P}^{\mathsf{mk}(\mathcal{C})}$ is:

$$\begin{array}{lcl}
\mathit{kennedy\_died} & \leftarrow & \mathit{oswald\_shot}, \\
\mathit{kennedy\_died} & \leftarrow & \mathit{someone\_else\_shot}, \\
\mathit{oswald\_shot} & \leftarrow & \top \wedge \mathsf{mk}(\mathit{oswald\_shot}), \\
\mathit{oswald\_shot} & \leftarrow & \mathsf{mk}(\mathit{oswald\_shot}).
\end{array}$$

Its weak completion together with:

$$\mathcal{E}_{\mathsf{mk}(\mathcal{C})} \;=\; \{\mathsf{mk}(\mathit{oswald\_shot}) \leftarrow \bot\},$$

is:

$$\begin{array}{lcl}
\mathit{kennedy\_died} & \leftrightarrow & \mathit{oswald\_shot} \vee \mathit{someone\_else\_shot}, \\
\mathit{oswald\_shot} & \leftrightarrow & (\top \wedge \mathsf{mk}(\mathit{oswald\_shot})) \vee \mathsf{mk}(\mathit{oswald\_shot}), \\
\mathsf{mk}(\mathit{oswald\_shot}) & \leftrightarrow & \bot.
\end{array}$$

Its least model of the weak completion is:

$$\langle\emptyset, \{\mathsf{mk}(\mathit{oswald\_shot}), \mathit{oswald\_shot}\}\rangle,$$

which does not entail $\mathcal{D}$, thus $\mathit{cond}_{\overline{os},se}$ is not valid. This complies with the conclusion the majority of people would anwer.

Let us consider $\mathit{cond}'_{\overline{os},se}$, where $\mathcal{C} = \{\overline{\mathit{oswald\_shot}}\}$ and $\mathcal{D} = \{\mathit{someone\_else\_shot}\}$. Additionally, $\mathit{cond}'_{\overline{os},se}$ implies *kennedy_died*, which affords us background information that needs to be explained in the context. Accordingly, we define observation $\mathcal{O} = \{\mathit{kennedy\_died}\}$. $\mathcal{P}^{\mathsf{mk}(\mathcal{C})}$ and $\mathcal{E}_{\mathsf{mk}(\mathcal{C})}$ are defined as just discussed for $\mathit{cond}_{\overline{os},se}$. The only explanation for $\mathcal{O}$ wrt $\mathcal{P}^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})}$, is

$$\mathcal{E} = \{\mathit{someone\_else\_shot} \leftarrow \top\}.$$

Consider the weak completion of $\mathcal{P}^{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}_{\mathsf{mk}(\mathcal{C})} \cup \mathcal{E}$:

$$\begin{array}{lcl} \mathit{kennedy\_died} & \leftrightarrow & \mathit{oswald\_shot} \vee \mathit{someone\_else\_shot}, \\ \mathit{oswald\_shot} & \leftrightarrow & (\top \wedge \mathsf{mk}(\mathit{oswald\_shot})) \vee \mathsf{mk}(\mathit{oswald\_shot}), \\ \mathsf{mk}(\mathit{oswald\_shot}) & \leftrightarrow & \bot, \\ \mathit{someone\_else\_shot} & \leftrightarrow & \top, \end{array}$$

for which the corresponding least model of the weak completion now entails $\mathcal{D}$:

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

and thus $\mathit{cond}'_{\overline{os},se}$ is valid, which corresponds to the opinion of the majority of people.

5 Deep and Superficial Revision

In our approach to counterfactual reasoning, revision is limited to the facts or conclusions of clauses explicitly stated in the premise. 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 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, (Dell'Acqua and Pereira 2005) 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 (Pereira et al. 1991a).

In (Dietz and Hölldobler 2015; Dietz et al. 2015a; Dietz et al. 2015b), minimal revision followed by abduction (MFRA) is proposed, where the condition of a conditional is tried to be explained as much as possible and only if necessary, revision is applied. For instance, consider the following clauses:

$$\begin{array}{lcl} \mathit{ffire} & \leftarrow & \mathit{lightning} \wedge \neg ab, \\ \mathit{lightning} & \leftarrow & \top, \\ ab & \leftarrow & \neg \mathit{dry\text{-}leaves}, \\ \mathit{dry\text{-}leaves} & \leftarrow & \neg rain, \\ beach & \leftarrow & \neg rain, \end{array}$$

Let us evaluate $cond(\overline{\mathit{dry\text{-}leaves}}, beach)$ and $cond(\overline{\mathit{lightning}}, beach)$: According to MRFA, the condition of $cond(\overline{\mathit{dry\text{-}leaves}}, beach)$ is explained by abducing $\mathcal{E} = \{rain \leftarrow \top\}$, which now changes the situation in which $cond(\overline{\mathit{dry\text{-}leaves}}, beach)$ is evaluated: $rain$ is not unknown anymore but true. On the other hand, by evaluating $cond(\overline{\mathit{lightning}}, beach)$ according to MRFA, only revision is applied and $rain$ stays unknown. The automated abductive mechanism in MRFA changes the situation for $cond(\overline{\mathit{dry\text{-}leaves}}, beach)$, which can be problematic if we want to evaluate $cond(\overline{\mathit{dry\text{-}leaves}}, beach)$ and $cond(\overline{\mathit{lightning}}, beach)$ with respect to the same situation and compare their validity: $cond(\overline{\mathit{dry\text{-}leaves}}, beach)$ is false and $cond(\overline{\mathit{lightning}}, beach)$ is unknown, according to MRFA.

On the other hand, the approach which we propose here gives us the possibility to pragmatically decide whether $rain$ should be true or stay unknown. In case we decide that $rain$ is true, then for both cases $cond(\overline{dry\text{-}leaves}, beach)$ and $cond(\overline{lightning}, beach)$ is false. However, in case want to keep this information unknown and not dependent on the original causes then $cond(\overline{dry\text{-}leaves}, beach)$ and $cond(\overline{lightning}, beach)$ are unknown.

We can fix the information in which we want to evaluate the counterfactual through the observation that is initially determined. MFRA does not include such a mechanism as the abductive procedure is uniquely determined by the condition under consideration.

6 Conclusion and Future Work

We have presented a computational logic approach for non probabilistic counterfactuals in the context of human reasoning by analyzing investigated examples by psychologists such as Byrne. Our approach is epistemologically akin to Rescher's, inspired by Pearl's theory (Pearl 2000), technically extended to LP semantics by weak completion and abduction, and scaffolded, if needed, in declarative debugging and belief revision methods (Dell'Acqua and Pereira 2005; Saptawijaya and Pereira 2013; Alferes et al. 1995; Alferes et al. 1996). 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 (Pereira et al. 2014; Pereira et al. 2014).

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 $\mathcal{IC}$ 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 (Pereira and Pinto 2011), to help rule out solutions. This relates Rescher's (Rescher 2005) 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.

In (Dietz et al. 2015a), the case is studied where both premises and conclusion evaluate to unknown. Then, abduction for fixing the premises and abduction for fixing the conclusion are needed. These fixating abductions must be relevant to one another. The definitions of relevancy studied there could be adopted here as well.

7 Acknowledgements

Thanks to Trevor Bench-Capon, Veronica Dahl, Chiaki Sakama & Ari Saptawijaya for valuable feedback. LMP acknowledges the support from FCT/MEC NOVA LINCS PEst UID/CEC/04516/2013. DAAD FITweltweit program supported 2nd author UNL stay.

References

ALFERES, J. J., DAMÁSIO, C. V., AND PEREIRA, L. M. 1995. A logic programming system for non-monotonic reasoning. *J. Autom. Reasoning 14,* 1, 93–147.

ALFERES, J. J., PEREIRA, L. M., AND PRZYMUSINSKI, T. C. 1996. Belief revision in non-monotonic reasoning and logic programming. *Fundam. Inform. 28,* 1-2, 1–22.

ANH, H. T., SAPTAWIJAYA, A., AND PEREIRA, L. M. 2012. Moral reasoning under uncertainty. In *LPAR*, N. Bjørner and A. Voronkov, Eds. Lecture Notes in Computer Science, vol. 7180. Springer, 212–227.

APT, K. R. AND VAN EMDEN, M. H. 1982. Contributions to the theory of logic programming. *Journal of the ACM 29,* 3, 841–862.

BARAL, C., GELFOND, M., AND RUSHTON, J. N. 2009. Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming 9,* 1, 57–144.

BARAL, C. AND HUNSAKER, M. 2007. Using the probabilistic logic programming language p-log for causal and counterfactual reasoning and non-naive conditioning. In *IJCAI*, M. M. Veloso, Ed. 243–249.

BENCH-CAPON, T. J. M. 1989. Representing counterfactual conditionals. In *Proc. of the Seventh Conference of the Society for the Study of Artificial Intelligence and Simulation of Behaviour*, A. G. Cohn, Ed. Pitman and Kaufmann, Brighton, England, 51–60.

BYRNE, R. M. J. 2007. *The Rational Imagination: How People Create Alternatives to Reality.* MIT Press, Cambridge, MA.

CLARK, K. L. 1978. Negation as failure. In *Logic and Data Bases*, H. Gallaire and J. Minker, Eds. Vol. 1. Plenum Press, New York, NY, 293–322.

DELL'ACQUA, P. AND PEREIRA, L. M. 2005. Preference revision via declarative debugging. In *EPIA*, C. Bento, A. Cardoso, and G. Dias, Eds. Lecture Notes in Computer Science, vol. 3808. Springer, 18–28.

DIETZ, E.-A. AND HÖLLDOBLER, S. 2015. A new computational logic approach to reason with conditionals. In *13th International Conference on Logic Programming and Non-monotonic Reasoning*, F. Calimeri, G.I., Truszczynski, and M., Eds. Lecture Notes in Artificial Intelligence, vol. 9345. Springer.

DIETZ, E.-A., HÖLLDOBLER, S., AND PEREIRA, L. M. 2015a. On conditionals. In *GCAI 2015. Global Conference on Artificial Intelligence*, G. Gottlob, G. Sutcliffe, and A. Voronkov, Eds. EPiC Series in Computer Science, vol. 36. EasyChair, 79–92.

DIETZ, E.-A., HÖLLDOBLER, S., AND PEREIRA, L. M. 2015b. On indicative conditionals. In *Proceedings of the First International Workshop on Semantic Technologies*, S. Hlldobler and Y. Liang, Eds. CEUR Workshop Proceedings, vol. 1339. CEUR-WS.org, 19–30.

DIETZ, E.-A., HÖLLDOBLER, S., AND RAGNI, M. 2012. A computational logic approach to the suppression task. In *Proceedings of the 34th Annual Conference of the Cognitive Science Society, CogSci 2013*, N. Miyake, D. Peebles, and R. P. Cooper, Eds. Austin, TX: Cognitive Science Society, 1500–1505.

DIETZ, E.-A., HÖLLDOBLER, S., AND RAGNI, M. 2013. A computational logic approach to the abstract and the social case of the selection task. In *Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning*. Cyprus.

DIETZ, E.-A., HÖLLDOBLER, S., AND WERNHARD, C. 2013. Modeling the suppression task under weak completion and well-founded semantics. *Journal of Applied Non-Classical Logics*.

FITTING, M. 1985. A Kripke-Kleene semantics for logic programs. *Journal of Logic Programming 2,* 4, 295–312.

GABBAY, D., GIORDANO, L., MARTELLI, A., OLIVETTI, N., AND SAPINO, M. 2000. Conditional reasoning in logic programming. *Journal of Logic Programming 44,* 1-3, 37 – 74.

GÄRDENFORS, P. Conditionals and changes of belief. *Acta Philosophica Fennica*.

GINSBERG, M. L. 1986. Counterfactuals. *Artif. Intell. 30,* 1, 35–79.

HALPERN, J. Y. AND HITCHCOCK, C. 2013. Graded causation and defaults. *CoRR abs/1309.1226*.

HEWINGS, M. 2013. *Advanced Grammar in Use with Answers: A Self-Study Reference and Practice Book for Advanced Learners of English*, 3 ed. Cambridge University Press. Lesson M14.

HÖLLDOBLER, S. 2009. *Logik und Logikprogrammierung, Band 1: Grundlagen*. Synchron Publishers Heidelberg.

HÖLLDOBLER, S. AND KENCANA RAMLI, C. D. 2009a. Logic programs under three-valued Łukasiewicz semantics. In *Logic Programming, 25th International Conference, ICLP 2009*, P. M. Hill and D. S. Warren, Eds. Lecture Notes in Computer Science, vol. 5649. Springer, Heidelberg, 464–478.

HÖLLDOBLER, S. AND KENCANA RAMLI, C. D. 2009b. Logics and networks for human reasoning. In *International Conference on Artificial Neural Networks, ICANN 2009, Part II*, C. Alippi, M. M. Polycarpou, C. G. Panayiotou, and G. Ellinas, Eds. Lecture Notes in Computer Science, vol. 5769. Springer, Heidelberg, 85–94.

KAKAS, A. C., KOWALSKI, R. A., AND TONI, F. 1993. Abductive logic programming. *Journal of Logic and Computation 2,* 6, 719–770.

KENCANA RAMLI, C. D. 2009. Master's thesis, Institute for Artificial Intelligence, Department of Computer Science, Technische Universität Dresden, Dresden.

LEWIS, D. 1973. *Counterfactuals*. Blackwell Publishers, Oxford.

LLOYD, J. W. 1984. *Foundations of Logic Programming*. Springer-Verlag New York, Inc., New York, NY, USA.

ŁUKASIEWICZ, J. 1920. O logice trójwartościowej. *Ruch Filozoficzny 5*, 169–171. English translation: On three-valued logic. In: Łukasiewicz J. and Borkowski L. (ed.). (1990). *Selected Works*, Amsterdam: North Holland, pp. 87–88.

PEARL, J. 2000. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, USA.

PEARL, J. 2011. The algorithmization of counterfactuals. *Ann. Math. Artif. Intell. 61,* 1, 29–39.

PEREIRA, L. M. AND APARÍCIO, J. N. 1989. Relevant counterfactuals. In *EPIA*, J. P. Martins and E. M. Morgado, Eds. Lecture Notes in Computer Science, vol. 390. Springer, 107–118.

PEREIRA, L. M., APARÍCIO, J. N., AND ALFERES, J. J. 1991a. Counterfactual reasoning based on revising assumptions. In *Intl. Logic Progrogramming Symposium (ILPS'91)*, V. A. Saraswat and K. Ueda, Eds. MIT Press, 566–577.

PEREIRA, L. M., APARÍCIO, J. N., AND ALFERES, J. J. 1991b. Hypothetical reasoning with well founded semantics. In *Scandinavian Conference on Artificial Intelligence: Proc. of the SCAI'91*, B. Mayoh, Ed. IOS Press, Amsterdam, 289–300.

PEREIRA, L. M., DIETZ, E., AND HÖLLDOBLER, S. 2014. Contextual abductive reasoning with side-effects. *Journal of Theory and Practice of Logic Programming (TPLP) 14,* 4-5, 633–648.

PEREIRA, L. M., DIETZ, E.-A., AND HÖLLDOBLER, S. 2014. A computational logic approach to the belief bias effect. In *14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014).* short paper.

PEREIRA, L. M. AND PINTO, A. M. 2011. Inspecting side-effects of abduction in logic programs. In *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning: Essays dedicated to Michael Gelfond*, M. Balduccini and T. C. Son, Eds. LNAI, vol. 6565. Springer, 148–163.

PEREIRA, L. M. AND SAPTAWIJAYA, A. 2016a. Abduction and beyond in logic programming with application to morality. *IfColog Journal of Logics and their Applications, Special issue on "Frontiers of Abduction"*.

PEREIRA, L. M. AND SAPTAWIJAYA, A. 2016d. *Programming Machine Ethics*. Vol. 26. Springer SAPERE series, Berlin.

PEREIRA, L. M. AND SAPTAWIJAYA, A. forthcoming 2016b. Counterfactuals in critical thinking with application to morality. In *Model-Based Reasoning in Science and Technology. Models and Inferences: Logical, Epistemological, and Cognitive Issues*, C. C. Magnani, L., Ed. SAPERE series.

PEREIRA, L. M. AND SAPTAWIJAYA, A. forthcoming 2016c. Counterfactuals in logic programming with applications to agent morality. In *Applied formal/mathematical philosophy*, G. P. R. Urbaniak, Ed. Argumentation & Reasoning series. Springer Logic.

RAMSEY, F. 1931. *The foundations of mathematics and other logical essays*. Harcourt, Brace and Company.

RESCHER, N. 2005. *What If?: Thought Experimentation In Philosophy.* Transaction Publishers.

RESCHER, N. 2007. *Conditionals.* MIT Press, Cambridge, MA.

ROUTEN, T. AND BENCH-CAPON, T. J. M. 1991. Hierarchical formalizations. *International Journal of Man-Machine Studies 35,* 1, 69–93.

SAPTAWIJAYA, A. AND PEREIRA, L. M. 2013. Tabled abduction in logic programs. *Theory and Practice of Logic Programming 13,* 4-5-Online-Supplement.

SAPTAWIJAYA, A. AND PEREIRA, L. M. 2014. Towards modeling morality computationally with logic programming. In *PADL*, M. Flatt and H.-F. Guo, Eds. Lecture Notes in Computer Science, vol. 8324. Springer, 104–119.

SHAFER, G. 1996. *The art of causal conjecture.* MIT Press.

SLOMAN, S. 2005. *Causal Models : How People Think about the World and Its Alternatives*. Oxford University Press, USA.

STENNING, K. AND VAN LAMBALGEN, M. 2008. *Human Reasoning and Cognitive Science*. A Bradford Book. MIT Press, Cambridge, MA.

VAN GELDER, A., ROSS, K. A., AND SCHLIPF, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM 38,* 3, 619–649.

VENNEKENS, J., BRUYNOOGHE, M., AND DENECKER, M. 2010. Embracing events in causal modelling: Interventions and counterfactuals in CP-logic. In *Logics in Artificial Intelligence - 12th European Conference, JELIA 2010, Helsinki, Finland, September 13-15, 2010. Proceedings*, T. Janhunen and I. Niemelä, Eds. Lecture Notes in Computer Science, vol. 6341. Springer, 313–325.

VENNEKENS, J., DENECKER, M., AND BRUYNOOGHE, M. 2009. CP-logic: A language of causal probabilistic events and its relation to logic programming. *CoRR abs/0904.1672*.

WOODWARD, J. 2003. *Making Things Happen: A Theory of Causal Explanation.* Oxford University Press.

Appendix A Proofs

Proposition 5 1
Given a two-valued abductive framework $\langle \mathcal{P}, \mathcal{A}_{2,\mathcal{P}}, \mathcal{IC}, \models_{wcs} \rangle$, and a three-valued abductive framework $\langle \mathcal{P}, \mathcal{A}_{\mathcal{P}}, \mathcal{IC}, \models_{wcs} \rangle$, where $\mathcal{P}$ is definite, $\mathcal{E} \subseteq \mathcal{A}_{2,\mathcal{P}}$ and $\mathcal{O}$ is an observation, the following holds:

1. If $\mathcal{E}$ is a two-valued explanation for $\mathcal{O}$ given $\mathcal{P}$ and $\mathcal{IC}$, then $\mathcal{E}$ is an explanation for $\mathcal{O}$ given $\mathcal{P}$ and $\mathcal{IC}$.
2. If $\mathcal{O}$ is two-valued explained given $\mathcal{P}$ and $\mathcal{IC}$ then $\mathcal{O}$ is explained given $\mathcal{P}$ and $\mathcal{IC}$.

Proof
We only need to prove that (1) holds, because (2) follows from (1). Let us assume that $\mathcal{E}$ is a two-valued explanation for $\mathcal{O}$ given $\mathcal{P}$ and $\mathcal{IC}$, that means $\mathcal{P} \not\models \mathcal{O}$, $\mathcal{P} \cup \mathcal{E} \models \mathcal{O}$ and $\mathsf{lm}_2(\mathcal{P} \cup \mathcal{E})$ satisfies $\mathcal{IC}$.
To show: $\mathcal{P} \not\models_{wcs} \mathcal{O}$, $(\mathcal{P} \cup \mathcal{E}) \models_{wcs} \mathcal{O}$ and $\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E})$ satisfies $\mathcal{IC}$.

1. $\mathcal{P} \not\models_{wcs} \mathcal{O}$ and $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{O}$ follow immediately given $\mathcal{P} \not\models \mathcal{O}$, $\mathcal{P} \cup \mathcal{E} \models \mathcal{O}$ and Proposition 1.
2. $\mathsf{lm}_2(\mathcal{P} \cup \mathcal{E})$ satisfies $\mathcal{IC}$ means that $\mathsf{lm}_2(\mathcal{P} \cup \mathcal{E} \cup \mathcal{IC})$ is consistent. This implies that the body of all clauses in $\mathcal{IC}$ are false in $\mathsf{lm}_2(\mathcal{P} \cup \mathcal{E})$. If they were true in $\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E})$, then, according to Proposition 1, they would also have to be true in $\mathsf{lm}_2(\mathcal{P} \cup \mathcal{E})$. Therefore, the body of all clauses in $\mathcal{IC}$ are either false or unknown in $\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E})$. In either case, $\mathcal{IC}$ is satisfied.

□

Proposition 6 1
Given a program $\mathcal{P}$, the sets $\mathcal{A}_{\mathcal{P}}$, and $\mathcal{A}^{\mathsf{mk}}$, the following holds:

$$\text{If } A \leftarrow \top \text{ or } A \leftarrow \bot \in \mathcal{A}_{\mathcal{P}} \text{ then } \mathsf{mk}(A) \leftarrow \top \in \mathcal{A}^{\mathsf{mk}} \text{ and } \mathsf{mk}(A) \leftarrow \bot \in \mathcal{A}^{\mathsf{mk}}.$$

Proof
'$A \leftarrow \top$ *or* $A \leftarrow \bot \in \mathcal{A}_{\mathcal{P}}$' is a sufficient condition in the proposition, because $\mathcal{A}_{\mathcal{P}}$ is the set of all positive and negative facts of all undefined atoms in $\mathcal{P}$. Therefore, whenever $A \leftarrow \top \in \mathcal{A}_{\mathcal{P}}$, necessarily also $A \leftarrow \bot \in \mathcal{A}_{\mathcal{P}}$, and vice versa. Given that $A \in \mathsf{undef}(\mathcal{P})$, A stays unknown in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}$. Accordingly, both, $\mathcal{P} \not\models_{wcs} A$ and $\mathcal{P} \not\models_{wcs} \overline{A}$ hold. Therefore, $\mathsf{mk}(A) \leftarrow \top$ and $\mathsf{mk}(A) \leftarrow \bot \in \mathcal{A}^{\mathsf{mk}}$. □

Proposition 8 1
Given a program $\mathcal{P}$ and a set of literals S, the following holds:

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

Proof
We distinguish between two cases for all $L \in S$, where $L = A$ or $L = \overline{A}$.

1. $A \in \mathsf{undef}(\mathcal{P})$, that means, A does not have a definition in $\mathcal{P}$. Then, the definition of A in $\mathcal{P}^{\mathsf{mk}(S)}$, is $\{A \leftarrow \mathsf{mk}(A)\}$. As $\mathsf{mk}(A)$ is a meta-predicate that has only been introduced by the program transformation, $\mathsf{mk}(A) \in \mathsf{undef}(\mathcal{P}^{\mathsf{mk}(S)})$. Accordingly, $\mathsf{mk}(A)$ and therefore A, are unknown in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}^{\mathsf{mk}(S)}$.
2. $A \notin \mathsf{undef}(\mathcal{P})$, 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

   $$A \leftarrow \mathit{Body}_1, A \leftarrow \mathit{Body}_2, \dots \in \mathcal{P},$$

   is conjoined with $\mathsf{mk}(A)$, that is, $\mathcal{P}^{\mathsf{mk}(S)}$ consists of the clauses

   $$A \leftarrow \mathit{Body}_1 \wedge \mathsf{mk}(A), A \leftarrow \mathit{Body}_2 \wedge \mathsf{mk}(A), \dots \in \mathcal{P},$$

   for every clause with head A. Because $\mathsf{mk}(A) \in \mathsf{undef}(\mathcal{P}^{\mathsf{mk}(S)})$, the body of these clauses are either unknown or false in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}^{\mathsf{mk}(S)}$.
   (b) The weak completion of $\mathcal{P}^{\mathsf{mk}(S)}$ contains

   $$A \leftrightarrow (\mathit{Body}_1 \wedge \mathsf{mk}(A)) \vee (\mathit{Body}_2 \wedge \mathsf{mk}(A)) \vee \dots \vee \mathsf{mk}(A).$$

   As $\mathit{Body}_i \wedge \mathsf{mk}(A)$, $1 \leq i \leq n$, are unknown or false according to (a) and $\mathsf{mk}(A)$ is unknown in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}^{\mathsf{mk}(S)}$ we conclude that A is is necessarily unknown in $\mathsf{lm}\,\mathsf{wc}\,\mathcal{P}^{\mathsf{mk}(S)}$. Because $\mathsf{mk}(A) \in \mathsf{undef}(\mathcal{P}^{\mathsf{mk}(S)})$, the body of these clauses are either unknown or false in $\mathcal{P}^{\mathsf{mk}(S)}$.

□

Proposition 10 1
Given a program $\mathcal{P}$, a consistent set of literals S and $\mathcal{E}_{\mathsf{mk}(S)} \subseteq \mathcal{A}^{\mathsf{mk}}$, the following holds:

$$\forall L \in S : \ \mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)} \models_{wcs} L.$$

Proof
We distinguish between the following two possible cases for all $L \in S$:

1. Assume that $L = \overline{A}$, thus $\{\mathsf{mk}(A) \leftarrow \bot\} \in \mathcal{E}_{\mathsf{mk}(S)}$. Accordingly,
$$\mathsf{mk}(A) \leftrightarrow \bot \in \mathsf{wc}\,(\mathcal{P} \cup \mathcal{E}_{\mathsf{mk}(S)}).$$
By the program transformation $\mathcal{P}^{\mathsf{mk}(S)}$, $\mathsf{mk}(A)$ is added to each clause whose definition is A, that is, for each such clause
$$A \leftrightarrow \mathit{Body} \wedge \mathsf{mk}(A) \in \mathsf{wc}\,(\mathcal{P} \cup \mathcal{E}_{\mathsf{mk}(S)}).$$
As each body whose head is A contains $\bot$, it will always be false. Consequently, A is false in $\mathsf{lm\,wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)})$, i.e.
$$\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)} \models_{wcs} S.$$
2. Assume that $L = A$, thus $\{\mathsf{mk}(A) \leftarrow \top\} \in \mathcal{E}_{\mathsf{mk}(S)}$. Accordingly,
$$\mathsf{mk}(A) \leftrightarrow \top \in \mathsf{wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)}).$$
By the program transformation $\mathcal{P}^{\mathsf{mk}(S)}$, $\mathsf{mk}(A)$ is added to each clause whose definition is A, that is, for each such clause
$$A \leftrightarrow \mathsf{mk}(A) \vee \in \mathsf{wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)}.$$
As $\mathsf{mk}(A)$ is true, the disjunction of the body of A will always be true. Consequently, A is true in $\mathsf{lm\,wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)})$, i.e.
$$\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)} \models_{wcs} A.$$

□

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 $A \leftrightarrow \top \vee ... \in \mathsf{wc}\,(\mathcal{P} \cup \{A \leftarrow \top\})$. Consequently, the body can always be reduced to $\top$.

Proposition 11 1
Given a program $\mathcal{P}$, a consistent set of literals S, $\mathcal{E}_{\mathsf{mk}(S)} \subseteq \mathcal{A}^{\mathsf{mk}}$
and $\mathcal{E}_S \subset \mathcal{A}_{\mathcal{P}}$, the following holds:
$$\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E}_S) \subset \mathsf{lm\,wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)}).$$

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

1. If $A \in I^\top$, then there is a clause $A \leftarrow \mathit{Body}$, such that *Body* is true. We need to distinguish between three cases, to show that $A \in J^\top$:
   (a) If $A = L$, then the clause that determines A's truth value in $\mathcal{P} \cup \mathcal{E}_{\mathsf{mk}(S)}$ is $\mathcal{E}_{\mathsf{mk}(S)}$. Analogously, the clause that determines A's truth value in $\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)}$ is $\mathcal{E}_{\mathsf{mk}(S)}$.

(b) If $A \neq L$, but A depends on L in $\mathcal{P}$, then A also depends on L in $\mathcal{P}^{\mathsf{mk}(S)}$ according to Proposition 7.

(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 $\mathcal{P}^{\mathsf{mk}(S)}$.

2. If $A \in I^{\perp}$, then for all clauses $A \leftarrow$ *Body*, *Body* is false in $\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E}_S)$. 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 $\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)}$ is $\mathcal{E}_{\mathsf{mk}(S)}$.

   (b) If $A \neq L$, but A depends on L in $\mathcal{P}$, then A also depends on L in $\mathcal{P}^{\mathsf{mk}(S)}$ according to Proposition 7.

   (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 $\mathcal{P}^{\mathsf{mk}(S)}$.

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

Proposition 12 1
Given a program $\mathcal{P}$, a consistent set of literals S, $\mathcal{E}_{\mathsf{mk}(S)} \subseteq \mathcal{A}^{\mathsf{mk}}$ and $\mathcal{E}_S \subset \mathcal{A}_{\mathcal{P}}$, the following holds:

$$\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E}_S) = \mathsf{lm\,wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)}) \setminus \{\mathsf{mk}(A) \mid \mathsf{mk}(A) \in \mathcal{A}^{\mathsf{mk}}\}.$$

Proof
Assume $\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E}_S) = \langle I^{\top}, I^{\perp} \rangle$ and $\mathsf{lm\,wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)}) = \langle J^{\top}, J^{\perp} \rangle$. According to Proposition 12, the following needs to hold:

$$I^{\top} = J^{\top} \setminus \{\mathsf{mk}(A) \mid \mathsf{mk}(A) \in \mathcal{A}^{\mathsf{mk}}\} \quad \text{and} \quad I^{\perp} = J^{\perp} \setminus \{\mathsf{mk}(A) \mid \mathsf{mk}(A) \in \mathcal{A}^{\mathsf{mk}}\}.$$

1. From Proposition 11, we know that $\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E}_S) \subset \mathsf{lm\,wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_S)$, which more specifically means that $I^{\top} \subset J^{\top}$ and $I^{\perp} \subset J^{\perp}$.
2. As mentioned in the proof of Proposition 11, the only reason for the strict subset relation is because $\mathsf{lm\,wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_S)$ necessarily states the truth about $\mathsf{mk}(A)$ for every $L \in S$, where $L = A$ or $L = \overline{A}$, which is not stated in $\mathsf{lm\,wc}\,(\mathcal{P} \cup \mathcal{E}_S)$.

□

Appendix B Lewis's Axiomatic System of Counterfactuals

Lewis (Lewis 1973) introduced an axiomatic system of counterfactuals, reformulated in (Gärdenfors), describing properties of counterfactual reasoning by humans. Also, Lewis's counterfactuals satisfy these properties, where $>$ is a counterfactual:

Fallacy of strengthening the condition	$A > B$	does not imply	$A \wedge C > B$
Fallacy of transitivity	$A > B$ & $B > C$	do not imply	$A > C$
Fallacy of contraposition	$A > B$	does not imply	$\neg B > \neg A$
Combination of sentences	$A > B$ & $A > C$	imply	$A > B \wedge C$

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 14

(Fallacy of strengthening the condition)
Let $\langle \mathcal{P}, \mathcal{A}^{\mathsf{mk}}, cond(\mathcal{C}, \mathcal{D}), \mathcal{IC}, \models_{wcs}\rangle$ and $\langle \mathcal{P}, \mathcal{A}^{\mathsf{mk}}, cond(\mathcal{C}', \mathcal{D}), \mathcal{IC}, \models_{wcs}\rangle$ be two counterfactual frameworks, where $S \subset S'$. The following holds:

If $cond(\mathcal{C}, \mathcal{D})$ is a valid counterfactual given $\mathcal{P}$ and $\mathcal{IC}$,
then $cond(\mathcal{C}', \mathcal{D})$ is not necessarily a valid counterfactual given $\mathcal{P}$ and $\mathcal{IC}$.

Proof

Consider $cond(\overline{\mathit{dry}}, \overline{\mathit{ffire}})$ again which we have shown valid wrt the program in consideration. A strengthening of the condition is possibly, $cond((\overline{\mathit{dry}}, \mathit{raising}), \mathit{ffire})$:

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 $cond(\overline{\mathit{dry}}, \mathit{raising}), \mathit{ffire})$ is not valid wrt the same program. □

Proposition 15

(Fallacy of contraposition)
Let $\langle \mathcal{P}, \mathcal{A}^{\mathsf{mk}}, cond(\mathcal{C}, \mathcal{D}), \mathcal{IC}, \models_{wcs}\rangle$ and $\langle \mathcal{P}, \mathcal{A}^{\mathsf{mk}}, cond(\mathcal{C}, \mathcal{D}), \mathcal{IC}, \models_{wcs}\rangle$ be two counterfactual frameworks. The following holds:

If $cond(\mathcal{C}, \mathcal{D})$ is a valid counterfactual given $\mathcal{P}$ and $\mathcal{IC}$,
then $cond(\mathcal{D}, \mathcal{C})$ is not necessarily a valid counterfactual given $\mathcal{P}$ and $\mathcal{IC}$.

Proof

The contrapositive counterfactual of valid $cond(\overline{\mathit{dry}}, \overline{\mathit{ffire}})$is not valid:

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

□

Proposition 16

(Fallacy of Transitivity)
Let $\langle \mathcal{P}, \mathcal{A}^{\mathsf{mk}}, cond(\mathcal{C}, \mathcal{D}), \mathcal{IC}, \models_{wcs}\rangle$, $\langle \mathcal{P}, \mathcal{A}^{\mathsf{mk}}, cond(\mathcal{C}', \mathcal{D}'), \mathcal{IC}, \models_{wcs}\rangle$ and
$\langle \mathcal{P}, \mathcal{A}^{\mathsf{mk}}, cond(\mathcal{C}, \mathcal{D}'), \mathcal{IC}, \models_{wcs}\rangle$ be 3 counterfactual frameworks. The following holds:

If $cond(\mathcal{C}, \mathcal{D})$ and $cond(\mathcal{C}', \mathcal{D}')$ are valid counterfactuals given $\mathcal{P}$ and $\mathcal{IC}$,
then $cond(\mathcal{C}, \mathcal{D}')$ is not necessarily a valid counterfactual given $\mathcal{P}$ and $\mathcal{IC}$.

Proof

Let us show this by the following example from (Routen and Bench-Capon 1991):

If James Bond had been born in Russia he would have been a Communist. $(cond_1)$
If James Bond had been a Communist he would have been a traitor. $(cond_2)$
If James Bond had been born in Russia he would have been a traitor. $(cond_3)$

Counterfactual $cond_1$ is based on the background knowledge that *James Bond was not born in a (former) communist country, such as Russia*. Consider $\mathcal{P}$ with $\mathcal{IC} = \emptyset$:

$$\begin{array}{lcl} communist(jbond) & \leftarrow & born(jbond, ru), \\ traitor(jbond) & \leftarrow & communist(jbond) \wedge \neg born(jbond, ru), \\ born(jbond, ru) & \leftarrow & \bot. \end{array}$$

The last clause is a negative fact, stating that James Bond was actually not born in Russia. In order to evaluate $cond_1$, we need to transform $\mathcal{P}$ wrt $S = \{born(jbond, ru)\}$. Accordingly, $\mathcal{E}_{\mathsf{mk}(S)}$ is $\{\mathsf{mk}(born(jbond, ru)) \leftarrow \top\}$, where $\mathsf{wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)})$ is:

$$\begin{array}{lcl} communist(jbond) & \leftrightarrow & born(jbond, ru), \\ traitor(jbond) & \leftrightarrow & communist(jbond) \wedge \neg born(jbond, ru), \\ born(jbond, ru) & \leftrightarrow & (\bot \wedge \mathsf{mk}(born(jbond, ru))) \vee \mathsf{mk}(born(jbond, ru)), \\ \mathsf{mk}(born(jbond, ru)) & \leftarrow & \top. \end{array}$$

Its corresponding least model of the weak completion is:

$$\langle\{\mathsf{mk}(born(jbond, ru)), born(jbond, ru), \boldsymbol{communist(jbond)}\}, \{traitor(jbond)\}\rangle.$$

It implies $\mathcal{D}$ and thus $cond_1$ is valid wrt $\mathcal{P}$ and $\mathcal{IC}$. For $cond_2$, $S = communist(jbond)$ and $\mathcal{E}_{\mathsf{mk}(S)} = \{\mathsf{mk}(communist(jbond)) \leftarrow \top\}$, where $\mathsf{lm}\,\mathsf{wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)})$ is:

$$\langle\{\mathsf{mk}(communist(jbond)), communist(jbond), \boldsymbol{traitor(jbond)}\}, \{born(jbond, ru)\}\rangle,$$

and thus also $cond_2$ is valid wrt to $\mathcal{P}$ and $\mathcal{IC}$ Let us now consider $cond_3$. The condition is $S = born(jbond, ru)$ therefore $\mathcal{E}_{\mathsf{mk}(S)}$ is $\{\mathsf{mk}(born(jbond, ru)) \leftarrow \top\}$. $\mathsf{lm}\,\mathsf{wc}\,(\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)})$ is:

$$\langle\{\mathsf{mk}(born(jbond, ru)), born(jbond, ru), communist(jbond)\}, \{traitor(jbond)\}, \rangle$$

which does not entail $\mathcal{D}$. □

Proposition 17
(Combination of sentences)
Let $\langle\mathcal{P}, \mathcal{A}^{\mathsf{mk}}, cond(\mathcal{C}, \mathcal{D}), \mathcal{IC}, \models_{wcs}\rangle$, and $\langle\mathcal{P}, \mathcal{A}^{\mathsf{mk}}, cond(\mathcal{C}, \mathcal{D}'), \mathcal{IC}, \models_{wcs}\rangle$ be two counterfactual frameworks. The following holds:

If $cond(\mathcal{C}, \mathcal{D})$ and $cond(\mathcal{C}, \mathcal{D}')$ are valid counterfactuals given $\mathcal{P}$ and $\mathcal{IC}$,
then $cond(\mathcal{C}, (\mathcal{D}, \mathcal{D}'))$ is a valid counterfactual given $\mathcal{P}$ and $\mathcal{IC}$.

Proof
If $cond(\mathcal{C}, (\mathcal{D}, \mathcal{D}'))$ is not valid, then there is at least one literal $L \in \mathcal{D} \cup \mathcal{D}'$ for which $\mathcal{P}^{\mathsf{mk}(S)} \cup \mathcal{E}_{\mathsf{mk}(S)} \not\models_{wcs} L$. Either $L \in \mathcal{D}$ or $L \in \mathcal{D}'$. Consequently, $cond(\mathcal{C}, \mathcal{D})$ or $cond(\mathcal{C}, \mathcal{D}')$ is not valid either. □