Contextual Reasoning: Usually Birds can Abductively Fly

Emmanuelle-Anna Dietz Saldanha[1], Steffen Hölldobler[1,2], and Luís Moniz Pereira[3⋆]

[1]International Center for Computational Logic, TU Dresden, 01062 Dresden, Germany, {dietz,sh}@iccl.tu-dresden.de
[2] and North-Caucasus Federal University, Stavropol, Russian Federation
[3]NOVA Laboratory for Computer Science and Informatics, Departamento de Informática Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal, lmp@fct.unl.pt

Abstract. We present a new logic programming approach to contextual reasoning, based on the Weak Completion Semantics (WCS), the latter of which has been successfully applied in the past to adequately model various human reasoning tasks. One of the properties of WCS is the open world assumption with respect to undefined atoms. This is a characteristic that is different to other common Logic Programming semantics, a property that seems suitable when modeling human reasoning. Notwithstanding, we have noticed that the famous Tweety default reasoning example, originally introduced by Reiter, cannot be modeled straightforwardly under WCS. Hence, to address the issue and taking Pereira and Pinto's inspection points as inspiration, we develop a notion of contextual reasoning for which we introduce contextual logic programs. We reconsider the formal properties of WCS with respect to these and verify whether they still hold. Finally, we set forth contextual abduction and show that not only the original Tweety example can be nicely modeled within the new approach, but more sophisticated examples as well, where context plays an important role.

1 Introduction

We shall develop a characterization of contextual reasoning, which has its origins in [21, 22], and which in turn was inspired by [23]. This will be done in the context of logic programming (cf. [18]), weak completion [13], abduction [16], Stenning and van Lambalgen's representation of implications as well as their semantic operator Φ [25] and our use of three-valued Łukasiewicz logic [19], instead of their use of [10] which has been assembled together in [9, 12–15]. This approach–which we call WCS for *Weak Completion Semantics*–has been applied to adequately model various human reasoning tasks [5, 7, 8] and to reasoning about conditionals [4, 6].

Consider the famous Tweety example from [24]: *Usually birds can fly. Tweety and Jerry are birds.* They can be encoded by the following (datalog) program, $\mathcal{P}_1$, where 'ab' stands for 'abnormal':

⋆ The authors are mentioned in alphabetical order.

$$can\_fly(X) \leftarrow bird(X) \wedge \neg ab(X). \qquad ab(X) \leftarrow \bot. \qquad bird(tweety) \leftarrow \top. \quad bird(jerry) \leftarrow \top.$$

We derive $can\_fly(tweety)$ and $can\_fly(jerry)$ as nothing is abnormal with respect to Tweety and Jerry. We modify the example by replacing the first statement with: *Usually birds can fly, but kiwis and penguins cannot.* This is encoded by $\mathcal{P}_2$:

$$can\_fly(X) \leftarrow bird(X) \wedge \neg ab(X). \quad ab(X) \leftarrow kiwi(X). \quad bird(tweety) \leftarrow \top. \quad ab(X) \leftarrow penguin(X). \quad bird(jerry) \leftarrow \top.$$

The ground instances of the weak completion of this program are as follows:

$$\begin{array}{ll} can\_fly(tweety) \leftrightarrow bird(tweety) \wedge \neg ab(tweety). & bird(tweety) \leftrightarrow \top. \\ can\_fly(jerry) \leftrightarrow bird(jerry) \wedge \neg ab(jerry). & bird(jerry) \leftrightarrow \top. \\ ab(tweety) \leftrightarrow kiwi(tweety) \vee penguin(tweety). & \\ ab(jerry) \leftrightarrow kiwi(jerry) \vee penguin(jerry). & \end{array}$$

Different than under Clark's completion semantics [3], the Stable Model Semantics [11] or the Well-Founded Semantics [26], $kiwi(tweety)$, $penguin(tweety)$, $kiwi(jerry)$ and $penguin(jerry)$ are not assumed to be false by the closed world assumption but stay unknown by WCS's open world assumption. In other words, they are neither true nor false.[1] This leads to the following consequence under WCS: As we don't know whether Tweety and Jerry are penguins or kiwis, we cannot derive that they can fly. Even if we model this case with the help of abduction, e.g. we observe that Jerry flies, $\mathcal{O} = \{can\_fly(jerry)\}$, we still need to state in its explanation that Jerry must be neither a penguin nor a kiwi.

We want to avoid explicitly stating that all exception cases are false such as the Completion Semantics, the Stable Model Semantics or Well-Founded Semantics will do. We don't think that humans actively apply the closed world assumption in reality, i.e. that they explicitly add negation to the cases they don't know anything about. Instead, we assume that humans, if they are not for some reason aware of exceptions, simply ignore these cases. In other words, they do not consciously become aware of all exceptions when they reason.[2] Accordingly, when modeling these cases with logic programs, we should leave the truth values of these exception cases unknown and find a mechanism that just ignores them. At the moment, we cannot express this syntactically in WCS programs.

After Sections 2 and 3 have introduced preliminaires and abduction, we present contextual programs in Section 4, and verify whether the same properties of the Φ operator hold for contextual programs as for the programs we have considered so far in our modeling of applications. Section 5 presents contextual abductive reasoning and specifies the notion of contextual side-effects. We terminate with conclusions, including open questions.

2 Preliminaries

(Program) clauses are expressions of the forms $A \leftarrow L_1 \wedge \ldots \wedge L_n$ (called *rules*), $A \leftarrow \top$ (called *facts*), and $A \leftarrow \bot$ (called *assumptions*), where $n \geq 1$, A is an

[1] If those atoms were assumed to be false, then typical human reasoning tasks like the suppression task [2] could not be modeled adequately [7].

[2] Currently, we know of at least 40 species of birds that can't fly.

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	$\bot$

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

$\leftrightarrow_{\text{Ł}}$	$\top$	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.

atom, and each L_i, $1 \leq i \leq n$, is a literal. A is called *head* and $L_1 \wedge \ldots \wedge L_n$ as well as $\top$ and $\bot$, standing for *true* and *false*, respectively, are called *body* of the corresponding clauses. A *(logic) program* is a set of clauses. Throughout this paper, $\mathcal{P}$ denotes a program. We assume for each $\mathcal{P}$ that the alphabet consists precisely of the symbols occurring in $\mathcal{P}$ and that non-propositional programs contain at least one constant symbol.

$\mathtt{g}\mathcal{P}$ denotes the set of all ground instances of clauses occurring in $\mathcal{P}$. Let $\mathcal{P}$ be a ground program and A a ground atom. An *exception clause* in $\mathcal{P}$ is a clause whose head is an abnormality predicate, i.e. it features predicate ab or ab_i, where $1 \leq i \leq n$. A is *defined in* $\mathcal{P}$ iff $\mathcal{P}$ contains a rule or a fact with head A. A is *undefined in* $\mathcal{P}$ iff A is not defined in $\mathcal{P}$. The *definition of* A *in* $\mathcal{P}$ is as follows:

$$def(A, \mathcal{P}) = \{A \leftarrow Body \mid A \leftarrow Body \text{ is a rule or a fact occurring in } \mathcal{P}\}.$$

$\neg A$ is *assumed in* $\mathcal{P}$ iff $\mathcal{P}$ contains an assumption with head A and $def(A, \mathcal{P}) = \emptyset$.

Let $\mathcal{P}$ be a ground program. Consider the following transformation: (1) Replace all clauses with the same head $A \leftarrow Body_1$, $A \leftarrow Body_2$, ... by $A \leftarrow Body_1 \vee Body_2 \vee \ldots$. (2) Replace all occurrences of $\leftarrow$ by $\leftrightarrow$. The resulting set is called *weak completion* of $\mathcal{P}$ or $wc\mathcal{P}$. Note that undefined atoms are not identified with $\bot$ as in the completion of $\mathcal{P}$ [3].

We consider the three-valued Łukasiewicz (or Ł-) logic [19] (see Table 1) and represent each interpretation I by $\langle I^\top, I^\bot \rangle$, where $I^\top = \{A \mid I(A) = \top\}$, $I^\bot = \{A \mid I(A) = \bot\}$, $I^\top \cap I^\bot = \emptyset$, and each ground atom $A \notin I^\top \cup I^\bot$ is mapped to U (*unknown*). Let $\langle I^\top, I^\bot \rangle$ and $\langle J^\top, J^\bot \rangle$ be two interpretations. We define $\langle I^\top, I^\bot \rangle \subseteq \langle J^\top, J^\bot \rangle$ iff $I^\top \subseteq J^\top$ and $I^\bot \subseteq J^\bot$. Logic programs as well as their weak completions admit a least model under Ł-logic. The least Ł-model of $wc\mathcal{P}$ can be obtained as the least fixed point of the following operator, which is due to Stenning and van Lambalgen [25]: $\Phi_\mathcal{P}(\langle I^\top, I^\bot \rangle) = \langle J^\top, J^\bot \rangle$, where

$$\begin{aligned} J^\top &= \{A \mid \text{there exists } A \leftarrow Body \in \mathtt{g}\mathcal{P} \text{ with } I(Body) = \top\}, \\ J^\bot &= \{A \mid \text{there exists } A \leftarrow Body \in \mathtt{g}\mathcal{P} \\ &\qquad \text{and for all } A \leftarrow Body \in \mathtt{g}\mathcal{P} \text{ we find } I(Body) = \bot\}. \end{aligned}$$

$(\mathcal{I}, \subseteq)$ is a complete partial order, where $\mathcal{I}$ is the set of all interpretations, the least one being $\langle \emptyset, \emptyset \rangle$. $\Phi_\mathcal{P}$ is monotonic on $(\mathcal{I}, \subseteq)$, i.e., $I_1 \subseteq I_2$ implies $\Phi_\mathcal{P}(I_1) \subseteq \Phi_\mathcal{P}(I_2)$. By the Knaster-Tarski theorem, $\Phi_\mathcal{P}$ has a least fixed point. Moreover, $\Phi_\mathcal{P}$ is continuous for finite datalog programs. For details see [13, 17].

Weak Completion Semantics (WCS) is the approach that considers weakly completed logic programs to reason with respect to the least Ł-models of these programs. We write $\mathcal{P} \models_{wcs} F$ iff formula F holds in the least Ł-model of $wc\mathcal{P}$.

In the remainder of this paper, $\mathcal{M}_\mathcal{P}$ denotes the Ł-model of $wc\mathcal{P}$, which has been computed by the $\Phi_\mathcal{P}$ operator.

A set of *integrity constraints* $\mathcal{IC}$ comprises clauses of the form $\mathrm{U} \leftarrow \mathit{Body}$, where *Body* is a conjunction of literals. Given $\mathcal{P}$ and $\mathcal{IC}$, $\mathcal{P}$ *satisfies* $\mathcal{IC}$ iff for all $\mathrm{U} \leftarrow \mathit{Body} \in \mathcal{IC}$, we find that $\mathcal{M}_\mathcal{P}(\mathit{Body}) \neq \top$. This definition allows us to specify that literals can be either unknown or false. This understanding is similar to the definition of the integrity constraints for the Well-founded Semantics in [20]. Note that in Section 4, when we introduce contextual programs we can restrict $\mathcal{IC}$s to have $\bot$ in heads instead, because contextual literals, as we will see, can compensate for that in $\mathcal{IC}$ bodies, without loss of generality.

3 Abduction

Let $\mathcal{P}$ be a ground program. The *set of abducibles of* $\mathcal{P}$ is

$$\begin{aligned}\mathcal{A}_\mathcal{P} = \{&A \leftarrow \top \mid A \text{ is undefined in } \mathcal{P} \text{ or } A \text{ is head of an exception clause in } \mathcal{P}\}\\ \cup\, \{&A \leftarrow \bot \mid A \text{ is undefined in } \mathcal{P} \text{ and } \neg A \text{ is not assumed in } \mathcal{P}\}.\end{aligned}$$

It consists of facts and assumptions for the undefined ground atoms occurring in $\mathcal{P}$ as well as of defeaters for assumptions and exception clauses occurring in $\mathcal{P}$. A fact of the form $A \leftarrow \top$ is called *defeater* for the assumption $A \leftarrow \bot$ or for an exception clause $ab(X) \leftarrow \mathit{Body}$, because the weak completion of the fact together with these clauses is semantically equivalent to $A \leftrightarrow \top$.[3]

An *abductive framework* consists of a logic program $\mathcal{P}$, a set of *abducibles* $\mathcal{A} \subseteq \mathcal{A}_\mathcal{P}$, a set of *integrity constraints* $\mathcal{IC}$ and the entailment relation $\models_{wcs}$. An abductive framework is denoted by $\langle \mathcal{P}, \mathcal{A}, \mathcal{IC}, \models_{wcs} \rangle$.

In the sequel, we consider datalog programs, i.e. abductive frameworks are defined with respect to the ground instances of the program. Consider again $\mathcal{P}_1$: Its least Ł-model, $\mathcal{M}_{\mathcal{P}_1} = \langle \mathcal{M}^\top_{\mathcal{P}_1}, \mathcal{M}^\bot_{\mathcal{P}_1} \rangle$, is

$$\begin{aligned}\mathcal{M}^\top_{\mathcal{P}_1} &= \{\mathit{bird}(\mathit{tweety}), \mathit{bird}(\mathit{jerry}), \mathit{can\_fly}(\mathit{tweety}), \mathit{can\_fly}(\mathit{jerry})\},\\ \mathcal{M}^\bot_{\mathcal{P}_1} &= \{ab(\mathit{tweety}), ab(\mathit{jerry})\}.\end{aligned}$$

The set of abducibles of $\mathcal{P}_1$ is $\mathcal{A}_{\mathcal{P}_1} = \{ab(\mathit{tweety}) \leftarrow \top,\ ab(\mathit{jerry}) \leftarrow \top\}$.

One should observe that each program and, in particular, each finite set of facts and assumptions has an Ł-model. For the latter case, it can be obtained by mapping all heads occurring in the set to true. Thus, in the next definition, explanations as well as the union of a program and an explanation are satisfiable.

An *observation* $\mathcal{O}$ is a set of ground literals; it is *explainable* in the framework $\langle \mathcal{P}, \mathcal{A}, \mathcal{IC}, \models_{wcs} \rangle$ iff there exists an $\mathcal{E} \subseteq \mathcal{A}$ called *explanation* such that $\mathcal{M}_{\mathcal{P} \cup \mathcal{E}} \models_{wcs} L$ for all $L \in \mathcal{O}$ and $\mathcal{P} \cup \mathcal{E}$ satisfies $\mathcal{IC}$. Sometimes explanations are required to be *minimal* in that they cannot be subsumed by another explanation.

Consider the framework $\langle \mathcal{P}_1, \mathcal{A}_{\mathcal{P}_1}, \emptyset, \models_{wcs} \rangle$ and let $\mathcal{O} = \{\neg \mathit{can\_fly}(\mathit{tweety})\}$. There are two explanations, viz. $\mathcal{E}_1 = \{ab(\mathit{tweety}) \leftarrow \top\}$ and $\mathcal{E}_2 = \{ab(\mathit{tweety}) \leftarrow \top,\ ab(\mathit{jerry}) \leftarrow \top\}$ with $\mathcal{E}_1$ being minimal. We obtain $\mathcal{M}_{\mathcal{P}_1 \cup \mathcal{E}_1}$ with

[3] Defeaters were not part of the first specification of the set of abducibles under WCS [7, 15], but without the defeaters the first example discussed in this section cannot be solved.

$$\mathcal{M}^{\top}_{\mathcal{P}_1 \cup \mathcal{E}_1} = \{bird(tweety), bird(jerry), can\_fly(jerry), ab(tweety)\},$$
$$\mathcal{M}^{\bot}_{\mathcal{P}_1 \cup \mathcal{E}_1} = \{ab(jerry), can\_fly(tweety)\}.$$

Consider again $\mathcal{P}_2$: We obtain $\mathcal{M}_{\mathcal{P}_2} = \langle \{bird(tweety), bird(jerry)\}, \emptyset \rangle$ and

$$\begin{aligned} \mathcal{A}_{\mathcal{P}_2} = \mathcal{A}_{\mathcal{P}_1} \cup \{\ & kiwi(tweety) \leftarrow \bot, \quad penguin(tweety) \leftarrow \bot, \\ & kiwi(tweety) \leftarrow \top, \quad penguin(tweety) \leftarrow \top, \\ & kiwi(jerry) \leftarrow \bot, \quad penguin(jerry) \leftarrow \bot, \\ & kiwi(jerry) \leftarrow \top, \quad penguin(jerry) \leftarrow \top \quad \} \end{aligned}$$

Considering $\langle \mathcal{P}_2, \mathcal{A}_{\mathcal{P}_2}, \emptyset, \models_{wcs} \rangle$ and $\mathcal{O} = \{can\_fly(jerry)\}$ we obtain the minimal explanation $\{kiwi(jerry) \leftarrow \bot,\ penguin(jerry) \leftarrow \bot\}$. In other words, in order to explain that *Jerry can fly* we have to assume that *Jerry* does not belong to any of the known exceptions. The question that arises already in [24] is whether and how we can avoid the explicit investigation into all known exceptions and conclude instead that *Jerry can fly by default*? Posed differently, does an ordinary human knowing the statements mentioned above and observing that *Jerry can fly* accept this observation as *default* or does he or she accept this observation only after explicitly reasoning just in case *Jerry is not a kiwi and not a penguin*?

4 Contextual Programs

In [23], Pereira and Pinto have introduced *inspection points* of the form $inspect(L)$, where L is a literal. Inspection points are treated as meta-predicates belonging to a special case of abducibles: $inspect(L)$ can only be abduced to explain some observation in case L is abduced to explain some given observation. More formally, $\mathcal{E}$ is an explanation if for each $inspect(L) \in \mathcal{E}$ we find that $L \in \mathcal{E}$ too. That is, $inspect(L)$ is only accepted in the context of L.

L	$\mathsf{ctxt}(L)$
$\top$	$\top$
$\bot$	$\bot$
U	$\bot$

Table 2. truth table for $\mathsf{ctxt}(L)$.

Inspired by the idea underlying inspection points, we introduce a new truth-functional operator ctxt (called *context*), whose meaning is specified in Table 2. With the help of ctxt, preferences on explanations, among other things, can be syntactically specified. These preferences are context-dependent.

The interpretation of ctxt can be understood as a mapping from three-valuedness to two-valuedness. It is one possible way to capture negation as failure (or negation by default) under WCS. The original idea of negation as failure [3] is to derive the negation of A in case we fail to derive A, where the meaning of derivation failure depends on the semantics. Negation as failure does not exist under WCS, quite the contrary is the case. Let $\mathcal{P}_3 = \{p \leftarrow q,\ p \leftarrow \bot\}$. Its weak completion is $wc\mathcal{P}_3 = \{p \leftrightarrow q \vee \bot\}$, which is semantically equivalent to $\{p \leftrightarrow q\}$. The least model of $wc\mathcal{P}_3$ is $\langle \emptyset, \emptyset \rangle$, so p and q are both unknown, whereas they would be false if negation as failure had been adopted under WCS. The assumption $p \leftarrow \bot$ has been overridden by the first clause of $\mathcal{P}_3$ and does not have any effect at all. On the other hand, $\mathsf{ctxt}(L) = \bot$ if L is unknown.

We extend the definition of programs by allowing expressions of the form $\mathsf{ctxt}(L)$ in the body of clauses. Formally, *contexual rules* are expressions of the

form $A \leftarrow L_1 \wedge \ldots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \ldots \wedge \mathsf{ctxt}(L_{m+p})$, where $m, p \in \mathbb{N}$ such that $m + p \geq 1$. A *contextual (datalog) program* $\mathcal{P}$ is a finite set of contextual rules, facts and assumptions.

Consider the following two programs and their weak completions:

$$\begin{aligned} \mathcal{P}_4 &= \{p \leftarrow \neg q\}, & \mathcal{P}_5 &= \{p \leftarrow \mathsf{ctxt}(\neg q)\}, \\ wc\mathcal{P}_4 &= \{p \leftrightarrow \neg q\}, & wc\mathcal{P}_5 &= \{p \leftrightarrow \mathsf{ctxt}(\neg q)\}. \end{aligned}$$

Starting with $I_0 = \langle \emptyset, \emptyset \rangle$ we compute the corresponding least fixed points of $\Phi_{\mathcal{P}_4}$ and $\Phi_{\mathcal{P}_5}$ as $\Phi_{\mathcal{P}_4}(\langle \emptyset, \emptyset \rangle) = \langle \emptyset, \emptyset \rangle$ and $\Phi_{\mathcal{P}_5}(\langle \emptyset, \emptyset \rangle) = \langle \emptyset, \{p\} \rangle = \Phi_{\mathcal{P}_5}(\langle \emptyset, \{p\} \rangle)$. $\mathsf{ctxt}(\neg q)$ in $\mathcal{P}_5$ mimics negation as failure: Nothing is known about q, therefore we derive that p is false in $\mathcal{P}_5$.

By means of ctxt, the common syntactical form for integrity constraints can be re-established: $\mathcal{IC}$ comprises clauses of the form $\bot \leftarrow \mathit{Body}$, where Body is a conjunction of $L_1 \wedge \ldots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \ldots \wedge \mathsf{ctxt}(L_{m+p})$, $m, p \in \mathbb{N}$ and $m + p \geq 1$. Given $\mathcal{P}$ and $\mathcal{IC}$, $\mathcal{P}$ *satisfies* $\mathcal{IC}$ iff for all $\bot \leftarrow \mathit{Body} \in \mathcal{IC}$, we find that $\mathcal{M}_{\mathcal{P}}(\mathit{Body}) = \bot$. Because ctxt is allowed in the body of these clauses, the same understanding as in Section 2 can be maintained: If a literal L should be either unknown or false, then we can simply write $\bot \leftarrow \mathsf{ctxt}(L)$.

In the remainder of this paper we assume that the underlying semantics of these contextual programs is the Ł-logic defined in Section 2 extended by the truth table for ctxt defined in Table 2. Although the Φ-operator admits a least fixed point for programs $\mathcal{P}_4$ and $\mathcal{P}_5$, we need to check whether this holds in general under the modified logic. As another example consider

$$\mathcal{P}_6 = \{p \leftarrow q \wedge \mathsf{ctxt}(r),\ r \leftarrow \neg s,\ s \leftarrow \mathsf{ctxt}(t)\}.$$

Applying Φ to $\mathcal{P}_6$ starting with the empty interpretation $I_0 = \langle \emptyset, \emptyset \rangle$ results in:

$$\begin{aligned} \Phi_{\mathcal{P}_6}(I_0) = \langle \emptyset, \{p, s\} \rangle = I_1, \qquad & \Phi_{\mathcal{P}_6}(I_1) = \langle \{r\}, \{p, s\} \rangle = I_2, \\ & \Phi_{\mathcal{P}_6}(I_2) = \langle \{r\}, \{s\} \rangle = \Phi_{\mathcal{P}_6}(\langle \{r\}, \{s\} \rangle). \end{aligned}$$

One should observe that $I_1 \subseteq I_2$, but $\Phi_{\mathcal{P}_6}(I_1) \subseteq \Phi_{\mathcal{P}_6}(I_2)$ does not hold. Hence, Φ is not monotonic anymore and, therefore, we cannot employ the Knaster-Tarski theorem to conclude that Φ has a least fixed point. In fact, it not always has a least fixed point, as the program $\mathcal{P}_7 = \{p \leftarrow \mathsf{ctxt}(\neg p)\}$ shows. Starting with $I_0 = \langle \emptyset, \emptyset \rangle$ we obtain:

$$\Phi_{\mathcal{P}_7}(I_0) = \langle \emptyset, \{p\} \rangle = I_1, \quad \Phi_{\mathcal{P}_7}(I_1) = \langle \{p\}, \emptyset \rangle = I_2, \quad \Phi_{\mathcal{P}_7}(I_2) = \langle \{\emptyset, \{p\} \rangle = I_1.$$

Is it possible to show that we can guarantee a least fixed point for a particular class of contextual programs? In this paper we follow an idea first developed by Fitting in [10] for programs under Kripke-Kleene logic. The idea is adapted to programs under Ł-logic in [12, 17]. The Banach Contraction Theorem [1] states that every contraction mapping has a unique fixed point. Hence, the goal is to show that $\Phi_{\mathcal{P}}$ is a contraction on an appropriately defined metric space. Unfortunately, semantic operators are in general not contractions. This holds for Fitting's approach as well as for its adaptation discussed in [12, 17], and program $\mathcal{P}_7$ shows that it also does not apply for the logic considered in this

Section. But, as we will show, it applies to acyclic contextual programs. Before being able to do so, we need some definitions.

A *metric* on a space $\mathcal{M}$ is a mapping $d : \mathcal{M} \times \mathcal{M} \mapsto \mathbb{R}$, such that for all $x, y, z \in \mathcal{M}$ we have: $d(x, y) = 0$ iff $x = y$, $d(x, y) = d(y, x)$, and $d(x, y) \leq d(x, z) + d(z, y)$. A metric space $(\mathcal{M}, d)$ is *complete* if every Cauchy sequence converges. A sequence $s_1, s_2, s_3, \ldots$ is *Cauchy* if, for every $\epsilon > 0$ there is an integer N such that for all $n, m \geq N, d(s_n, s_m) \leq \epsilon$. The sequence *converges* if there is an s such that, for every $\epsilon > 0$, there is an integer N such that for all $n \geq N, d(s_n, s) \leq \epsilon$. Let $(\mathcal{M}, d)$ be a metric space: A mapping $f : \mathcal{M} \mapsto \mathcal{M}$ is a *contraction* if for all $x, y \in \mathcal{M}$ there exists a $k \in \mathbb{R}$ with $0 < k < 1$ such that $d(f(x), f(y)) \leq k \cdot d(x, y)$.

Theorem 1. [1] *A contraction mapping f on a complete metric space has a unique fixed point. The sequence x, $f(x)$, $f(f(x))$, ... converges to this fixed point for any x.*

A *level mapping* for a contextual program $\mathcal{P}$ is a function ℓ which assigns to each ground atom a natural number. It is extended to ground literals and expressions of the form $\mathsf{ctxt}(L)$ as follows, where L is a ground literal and A a ground atom: $\ell(\neg A) = \ell(A)$ and $\ell(\mathsf{ctxt}(L)) = \ell(L)$. A *contextual program $\mathcal{P}$ is acyclic with respect to a level mapping ℓ* iff for every rule $A \leftarrow L_1 \wedge \ldots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \ldots \wedge \mathsf{ctxt}(L_{m+p}) \in \mathcal{P}$ and for all $1 \leq i \leq m$, we find that $\ell(A) > \ell(L_i)$ and for all $m + 1 \leq j \leq m + p$, we find that $\ell(A) > \ell(\mathsf{ctxt}(L_j))$. A *contextual program $\mathcal{P}$ is acyclic* iff it is acyclic with respect to some level mapping ℓ.

Let ℓ be a level mapping and I and J be interpretations. The function $d_\ell : \mathcal{I} \times \mathcal{I} \mapsto \mathbb{R}$ is defined as

$$d_\ell(I, J) = \begin{cases} (\frac{1}{2})^n & I \neq J \text{ and } I(A) = J(A) \neq \mathrm{U} \text{ for all } A \text{ with } \ell(A) < n \text{ and,} \\ & \text{for some } A \text{ with } \ell(A) = n, I(A) \neq J(A) \text{ or } I(A) = J(A) = \mathrm{U}, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition 2. [17] *d_ℓ is a metric and $(\mathcal{I}, d_\ell)$ is a complete metric space.*

Theorem 3. *Let $\mathcal{P}$ be an acyclic contextual program with respect to the level mapping ℓ. Then $\Phi_\mathcal{P}$ is a contraction on $(\mathcal{I}, d_\ell)$.*

Proof. Given that $\mathcal{P}$ is a contextual program, we have to show that

$$d_\ell(\Phi_\mathcal{P}(I), \Phi_\mathcal{P}(J)) \leq \frac{1}{2} \cdot d_\ell(I, J). \tag{1}$$

If $I = J$, then $\Phi_\mathcal{P}(I) = \Phi_\mathcal{P}(J)$, $d_\ell(\Phi_\mathcal{P}(I), \Phi_\mathcal{P}(J)) = d_\ell(I, J) = 0$, and (1) holds. If $I \neq J$, then since ℓ is total, we obtain $d_\ell(I, J) = \frac{1}{2}^n$ for some $n \in \mathbb{N}$. We will show that $d_\ell(\Phi_\mathcal{P}(I), \Phi_\mathcal{P}(J)) \leq (\frac{1}{2})^{n+1}$, i.e. for all ground atoms $A \in \mathsf{g}\mathcal{P}$, with $\ell(A) \leq n$ we have that $\Phi_\mathcal{P}(I)(A) = \Phi_\mathcal{P}(J)(A)$.
Let us take some A with $\ell(A) \leq n$ and let $def(A, \mathcal{P})$ be the set of all clauses in $\mathsf{g}\mathcal{P}$ where A is the head of. As $\mathcal{P}$ is acyclic, for all rules

$$A \leftarrow L_1 \wedge \ldots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \ldots \wedge \mathsf{ctxt}(L_{m+p}) \in def(A, \mathcal{P})$$

for all $1 \leq i \leq m$ we obtain $\ell(L_i) < \ell(A) \leq n$, and for all $m+1 \leq j \leq m+p$, we obtain $\ell(\mathsf{ctxt}(L_j)) < \ell(A) \leq n$. We know that $d_\ell(I,J) \leq (\frac{1}{2})^n$, so for all $1 \leq i \leq m$, $I(L_i) = J(L_i)$, and for all $m+1 \leq j \leq m+p$, $I(\mathsf{ctxt}(L_j)) = J(\mathsf{ctxt}(L_j))$. Therefore, I and J interpret identically all bodies of clauses with A in the head. Consequently, $\Phi_\mathcal{P}(I)(A) = \Phi_\mathcal{P}(J)(A)$. ■

Corollary 4. *Let $\mathcal{P}$ be an acyclic contextual program. $\Phi_\mathcal{P}$ has a unique fixed point. This fixed point can be reached in finite time by iterating $\Phi_\mathcal{P}$ starting from any interpretation.*

Proof. 1. By Proposition 2, $(\mathcal{I}, d_\ell)$ is a complete metric space.
2. By Theorem 3, $\Phi_\mathcal{P}$ is a contraction for acyclic contextual $\mathcal{P}$ on $\mathcal{I}$ using d_ℓ.
3. By 1. and 2. and the Banach Contraction Theorem, Theorem 1, for any acyclic contextual $\mathcal{P}$, $\Phi_\mathcal{P}$ has a unique fixed point. Furthermore, this fixed point can be reached starting from any interpretation.
4. By 3. and because $\mathcal{P}$ is finite, $\Phi_\mathcal{P}$ can be reached in a finite number of times starting from any interpretation. ■

The following result can be shown analogously to the proof for $\mathcal{P}$ in [17].

Proposition 5. *Let $\mathcal{P}$ be an acyclic contextual program. Then, the least fixed point of $\Phi_\mathcal{P}$ is a model of $wc\mathcal{P}$.*

Proof. Assume that the least fixed point of $\Phi_\mathcal{P} = \langle I^\top, I^\bot \rangle$ and $A \leftrightarrow F \in wc\mathcal{P}$. We distinguish between 3 cases:

1. If $I(A) = \top$, then according to the definition of $\Phi_\mathcal{P}$, there exists a clause $A \leftarrow L_1 \wedge \cdots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \cdots \wedge \mathsf{ctxt}(L_{m+p})$, such that for all $1 \leq i \leq m+p$, $I(L_i) = \top$. As $L_1 \wedge \cdots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \cdots \wedge \mathsf{ctxt}(L_{m+p})$ is one of the disjuncts in F, $I(F) = \top$, and thus $I(A \leftrightarrow F) = \top$.
2. If $I(A) = \mathrm{U}$, then according to the definition of $\Phi_\mathcal{P}$, there is no clause $A \leftarrow L_1 \wedge \cdots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \cdots \wedge \mathsf{ctxt}(L_{m+p})$, such that for all $1 \leq i \leq m+p$, $I(L_i) = \top$ and there is at least one clause $A \leftarrow L_1 \wedge \cdots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \cdots \wedge \mathsf{ctxt}(L_{m+p})$, such that for all $1 \leq i \leq m+p$, $I(L_i) \neq \bot$ and there exists $1 \leq j \leq m$, $I(L_j) = \mathrm{U}$. As none of the disjuncts in F is true, and at least one is unknown, $I(F) = \mathrm{U}$ and thus $I(A \leftrightarrow F) = \top$.
3. If $I(A) = \bot$, then according to the definition of $\Phi_\mathcal{P}$, there exists a clause $A \leftarrow L_1 \wedge \cdots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \cdots \wedge \mathsf{ctxt}(L_{m+p})$ and for all clauses $A \leftarrow L_1 \wedge \cdots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \cdots \wedge \mathsf{ctxt}(L_{m+p})$, there exists $1 \leq i \leq m+p$ such that $I(L_i) = \bot$ or there exists $m+1 \leq j \leq m+p$, such that $I(L_j) \neq \top$. As all disjuncts in F are false, $I(F) = \bot$ and thus $I(A \leftrightarrow F) = \top$. ■

As has been shown in [14], for non-contextual programs, the least fixed point of $\Phi_\mathcal{P}$ is identical to the least model of the weak completion of $\mathcal{P}$. As the following example shows this does not hold for contextual programs, as the weak completion of contextual programs might have more than one minimal model. Consider $\mathcal{P} = \{s \leftarrow \neg r,\ r \leftarrow \neg p \wedge q,\ q \leftarrow \mathsf{ctxt}(\neg p)\}$. Its weak completion is $wc\mathcal{P} = \{s \leftrightarrow \neg r,\ r \leftrightarrow \neg p \wedge q,\ q \leftrightarrow \mathsf{ctxt}(\neg p)\}$. The least fixed point of $\Phi_\mathcal{P}$

is $\langle\{s\},\{q,r\}\rangle$, which is a minimal model of $wc\mathcal{P}$. However, yet another minimal model of $wc\mathcal{P}$ is $\langle\{q,r\},\{p,s\}\rangle$. But this model is not supported in the sense that if we iterate $\Phi_{\mathcal{P}}$ starting with this model, then we will compute $\langle\{s\},\{q,r\}\rangle$. As the fixpoint of $\Phi_{\mathcal{P}}$ is unique and the only supported minimal model of $wc\mathcal{P}$, we define $\mathcal{P} \models_{wcs} F$ if and only if F holds in the least fixed point of $\Phi_{\mathcal{P}}$.

5 Contextual Abduction

How can we prefer explanations that explain the normal cases to explanations that explain the exception cases? How can we express that some explanations have to be considered only if there is some evidence for considering the exception cases? We want to avoid having to consider all explanations if there is no evidence for considering exception cases. On the other hand, we don't want to state that all exception cases are false, as we must do for $\mathcal{O} = \{\mathit{can\_fly}(\mathit{jerry})\}$ given $\mathcal{P}_1$.

Consider the following definition for strong dependency in contextual programs: Given a rule $A \leftarrow L_1 \wedge \ldots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \ldots \wedge \mathsf{ctxt}(L_{m+p})$ for all $1 \leq i \leq m$, A 'strongly depends on' L_i. The 'strongly depends on' relation is transitive. If A strongly depends on L_i, then $\neg A$ strongly depends on L_i. Furthermore, if $L_i = B$, then A strongly depends on $\neg B$ and if $L_i = \neg B$, then A strongly depends on B. Consider program $\mathcal{P} = \{p \leftarrow r,\ p \leftarrow \mathsf{ctxt}(q)\}$: p strongly depends on r, p strongly depends on $\neg r$, $\neg p$ strongly depends on r and $\neg p$ strongly depends on $\neg r$. However, p does not strongly depend on q, nor on $\mathsf{ctxt}(q)$.

A *contextual abductive framework* is a quadruple $\langle\mathcal{P},\mathcal{A},\mathcal{IC},\models_{wcs}\rangle$, consisting of an acyclic contextual program $\mathcal{P}$, a set of abducibles $\mathcal{A} \subseteq \mathcal{A}_{\mathcal{P}}$, a set of integrity constraints $\mathcal{IC}$, and the entailment relation $\models_{wcs}$, where $\mathcal{A}_{\mathcal{P}}$ is defined as

$$\begin{aligned}\mathcal{A}_{\mathcal{P}} = \{&A \leftarrow \top \mid A \text{ is undefined in } \mathcal{P} \text{ or } A \text{ is head of an exception clause in } \mathcal{P}\}\\ \cup\ \{&A \leftarrow \bot \mid A \text{ is undefined in } \mathcal{P} \text{ and } \neg A \text{ is not assumed in } \mathcal{P}\}.\end{aligned}$$

Let an *observation* $\mathcal{O}$ be a non-empty set of ground literals. Note that $\mathcal{O}$ does not contain formulas of the form $\mathsf{ctxt}(L)$.

Let $\langle\mathcal{P},\mathcal{A},\mathcal{IC},\models_{wcs}\rangle$ be a contextual abductive framework where $\mathcal{P}$ satisfies $\mathcal{IC}$, $\mathcal{E} \subseteq \mathcal{A}$ and $\mathcal{O}$ is an observation. $\mathcal{O}$ is *contextually explained by $\mathcal{E}$ given $\mathcal{P}$ and $\mathcal{IC}$* iff $\mathcal{O}$ is explained by $\mathcal{E}$ given $\mathcal{P}$ and $\mathcal{IC}$ and for all $A \leftarrow \top \in \mathcal{E}$ and for all $A \leftarrow \bot \in \mathcal{E}$ there exists $L \in \mathcal{O}$, such that L strongly depends on A. $\mathcal{O}$ is *contextually explainable given $\mathcal{P}$ and $\mathcal{IC}$* iff there exists some $\mathcal{E}$ such that $\mathcal{O}$ is contextually explained by $\mathcal{E}$ given $\mathcal{P}$ and $\mathcal{IC}$.

Similar to explanations in abduction, we assume that contextual explanations are minimal, that is, there is no other contextual explanation $\mathcal{E}' \subset \mathcal{E}$ for $\mathcal{O}$ given $\mathcal{P}$ and $\mathcal{IC}$. Note that, compared to explanations, contextual explanations have an additional requirement: There has to be a literal in the observation, which strongly depends on some atom for which there exists a fact or assumption in $\mathcal{E}$. We distinguish between skeptical and credulous reasoning in the usual way:

F *contextually follows skeptically from $\mathcal{P}$, $\mathcal{IC}$ and $\mathcal{O}$* iff $\mathcal{O}$ can be contextually explained 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 contextually follows credulously from $\mathcal{P}$, $\mathcal{IC}$ and $\mathcal{O}$ iff there exists some $\mathcal{E}$ that contextually explains $\mathcal{O}$ and it holds that $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$.

Let us adapt $\mathcal{P}_2$ from the introduction such that all exceptions, viz. X being a penguin or a kiwi, are evaluated with respect to their context, ctxt, instead:

$$\begin{aligned}\mathcal{P}_8 = \{\ & \mathit{can\_fly}(X) \leftarrow \mathit{bird}(X) \wedge \neg \mathit{ab}_1(X), && \mathit{bird}(\mathit{tweety}) \leftarrow \top,\\ & \mathit{ab}_1(X) \leftarrow \mathsf{ctxt}(\mathit{kiwi}(X)), && \mathit{bird}(\mathit{jerry}) \leftarrow \top,\\ & \mathit{ab}_1(X) \leftarrow \mathsf{ctxt}(\mathit{penguin}(X)) && \}.\end{aligned}$$

We obtain $\mathcal{M}_{\mathcal{P}_8}$ with

$$\begin{aligned}\mathcal{M}^{\top}_{\mathcal{P}_8} &= \{\mathit{bird}(\mathit{tweety}), \mathit{bird}(\mathit{jerry}), \mathit{can\_fly}(\mathit{tweety}), \mathit{can\_fly}(\mathit{jerry})\},\\ \mathcal{M}^{\bot}_{\mathcal{P}_8} &= \{\mathit{ab}_1(\mathit{tweety}), \mathit{ab}_1(\mathit{jerry})\}.\end{aligned}$$

This model already entails the observation $\mathcal{O} = \{\mathit{can\_fly}(\mathit{jerry})\}$ and, thus, $\mathcal{O}$ does not need any explanation beyond the minimal empty one.

5.1 More about Tweety

Consider, the following extension of $\mathcal{P}_8$:

$$\begin{aligned}\mathcal{P}_9 = \mathcal{P}_8 \cup \{\ & \mathit{kiwi}(X) \leftarrow \mathit{featherslikeHair}(X) \wedge \neg \mathit{ab}_2(X),\\ & \mathit{penguin}(X) \leftarrow \mathit{blackAndWhite}(X) \wedge \neg \mathit{ab}_3(X),\\ & \mathit{ab}_2(X) \leftarrow \mathsf{ctxt}(\mathit{inEurope}(X)),\\ & \mathit{ab}_3(X) \leftarrow \mathsf{ctxt}(\mathit{inEurope}(X)) \qquad \}.\end{aligned}$$

We obtain $\mathcal{M}_{\mathcal{P}_9}$ with

$$\begin{aligned}\mathcal{M}^{\top}_{\mathcal{P}_9} &= \{\mathit{bird}(\mathit{tweety}), \mathit{bird}(\mathit{jerry}), \mathit{can\_fly}(\mathit{jerry}), \mathit{can\_fly}(\mathit{tweety})\,\},\\ \mathcal{M}^{\bot}_{\mathcal{P}_9} &= \{\mathit{ab}_i(\mathit{tweety}) \mid 1 \leq i \leq 3\} \cup \{\mathit{ab}_i(\mathit{jerry}) \mid 1 \leq i \leq 3\,\}.\end{aligned}$$

We observe that Tweety does not fly and that Tweety has feathers like hair: $\mathcal{O} = \{\neg \mathit{can\_fly}(\mathit{tweety}),\ \mathit{featherslikeHair}(\mathit{tweety})\}$. The set of abducibles is:

$$\begin{aligned}\mathcal{A}_{\mathcal{P}_9} = \mathcal{A}_{\mathcal{P}_2} \cup \{\ & \mathit{featherslikeHair}(\mathit{jerry}) \leftarrow \top, && \mathit{featherslikeHair}(\mathit{jerry}) \leftarrow \bot,\\ & \mathit{featherslikeHair}(\mathit{tweety}) \leftarrow \top, && \mathit{featherslikeHair}(\mathit{tweety}) \leftarrow \bot,\\ & \mathit{blackAndWhite}(\mathit{jerry}) \leftarrow \top, && \mathit{blackAndWhite}(\mathit{jerry}) \leftarrow \bot,\\ & \mathit{blackAndWhite}(\mathit{tweety}) \leftarrow \top, && \mathit{blackAndWhite}(\mathit{tweety}) \leftarrow \bot,\\ & \mathit{inEurope}(\mathit{jerry}) \leftarrow \top, && \mathit{inEurope}(\mathit{jerry}) \leftarrow \bot,\\ & \mathit{inEurope}(\mathit{tweety}) \leftarrow \top, && \mathit{inEurope}(\mathit{tweety}) \leftarrow \bot \qquad \}.\end{aligned}$$

$\mathcal{E} = \{\mathit{featherslikeHair}(\mathit{tweety}) \leftarrow \top\}$ is the only (minimal) contextual explanation for $\mathcal{O}$. We obtain $\mathcal{M}_{\mathcal{P}_9 \cup \mathcal{E}}$ with

$$\begin{aligned}\mathcal{M}^{\top}_{\mathcal{P}_9 \cup \mathcal{E}} &= \{\ \mathit{bird}(\mathit{tweety}), \mathit{featherslikeHair}(\mathit{tweety}), \mathit{kiwi}(\mathit{tweety}), \mathit{ab}_1(\mathit{tweety}),\\ &\qquad \mathit{bird}(\mathit{jerry}), \mathit{can\_fly}(\mathit{jerry})\,\},\\ \mathcal{M}^{\bot}_{\mathcal{P}_9 \cup \mathcal{E}} &= \{\ \mathit{can\_fly}(\mathit{tweety}), \mathit{ab}_2(\mathit{tweety}), \mathit{ab}_3(\mathit{tweety})\} \cup \{\mathit{ab}_i(\mathit{jerry}) \mid 1 \leq i \leq 3\,\}.\end{aligned}$$

From this model we derive that Tweety is a kiwi, even though $\mathit{inEurope}(\mathit{tweety})$ is unknown. This is exactly what we assume humans do while reasoning: They do not assume anything about Tweety living in Europe, and by using ctxt we can model their ignorance concerning it.

5.2 More about Jerry

Consider again $\mathcal{P}_9$ from the previous example together with the observation that Jerry flies and Jerry lives in Europe: $\mathcal{O} = \{can\_fly(jerry), inEurope(jerry)\}$. $\mathcal{A}_{\mathcal{P}_9}$ is defined in the previous example. The only contextual explanation for $\mathcal{O}$ is $\mathcal{E} = \{inEurope(jerry) \leftarrow \top\}$ and we obtain $\mathcal{M}_{\mathcal{P}_9 \cup \mathcal{E}}$ with

$$\begin{aligned}\mathcal{M}^{\top}_{\mathcal{P}_9 \cup \mathcal{E}} &= \{\, inEurope(jerry), bird(jerry), can\_fly(jerry), ab_2(jerry), ab_3(jerry),\\ &\qquad bird(tweety), can\_fly(tweety)\,\},\\ \mathcal{M}^{\bot}_{\mathcal{P}_9 \cup \mathcal{E}} &= \{\, kiwi(jerry), penguin(jerry), ab_1(jerry)\} \cup \{ab_i(tweety) \mid 1 \leq i \leq 3\,\}.\end{aligned}$$

From this model we can derive, from the contextual clauses, that Jerry is not a penguin and Jerry is not a kiwi!

5.3 Contextual Side-effects

Consider the following definitions originally presented in [21,22], which captures the idea of contextual side-effects: Given a program $\mathcal{P}$ and a set of integrity constraints $\mathcal{IC}$, let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two observations and $\mathcal{E}_1$ be a contextual explanation for $\mathcal{O}_1$. $\mathcal{O}_2$ is a *necessary contextual side-effect of* $\mathcal{O}_1$ *given* $\mathcal{P}$ *and* $\mathcal{IC}$ iff $\mathcal{O}_2$ cannot be contextually explained but $\mathcal{O}_1 \cup \mathcal{O}_2$ is contextually explained by $\mathcal{E}_1$. $\mathcal{O}_2$ is a *possible contextual side-effect of* $\mathcal{O}_1$ *given* $\mathcal{P}$ *and* $\mathcal{IC}$ iff $\mathcal{O}_2$ cannot be contextually explained by $\mathcal{E}_1$ but $\mathcal{O}_1 \cup \mathcal{O}_2$ is contextually explained by $\mathcal{E}_1$. The idea behind contextual side-effects is that every explanation $\mathcal{E}_1$ for $\mathcal{O}_1$ gives us an explanation for $\mathcal{O}_2$. Note that a necessary contextual side-effect is also a possible contextual side-effect.

Consider again the Tweety example of Section 5.1, where the observation is $\mathcal{O} = \{\neg can\_fly(tweety), featherslikeHair(tweety)\}$ and its only contextual explanation is $\mathcal{E} = \{featherslikeHair(tweety) \leftarrow \top\}$. $\mathcal{O}_2 = \{\neg can\_fly(tweety)\}$ cannot be contextually explained by $\mathcal{E}$. However, $\mathcal{O}$ can be contextually explained by $\mathcal{E}$ and $\mathcal{O}_2$ is a necessary contextual side-effect of $\mathcal{O} \backslash \mathcal{O}_2 = \{featherslikeHair(tweety)\}$.

On the other hand, consider again the Jerry example of Section 5.2, where the observation is $\mathcal{O} = \{can\_fly(jerry), inEurope(jerry)\}$, for which the only explanation is $\mathcal{E} = \{inEurope(jerry) \leftarrow \top\}$. As $\mathcal{O}_2 = \{can\_fly(jerry)\}$ already follows from the empty explanation, $\mathcal{E}_2 = \emptyset$, $\mathcal{O}_2$ cannot be considered a contextual side-effect of $\mathcal{O} \setminus \mathcal{O}_2 = \{inEurope(jerry)\}$.

6 Conclusion

Prompted by the famous Tweety example, we first show that the Weak Completion Semantics does not yield the desired results. We would like to avoid having to abductively consider all exception cases and to automatically prefer normal explanations to those explanations specifying such exception cases. To do so, we set forth contextual programs, for the purpose of which we introduce ctxt, a new truth-functional operator, which turns out to fit quite well with the interpretation of negation as failure under three-valued semantics. Unfortunately, the usual Φ operator is not monotonic with respect to these contextual programs

anymore. Even worse, the Φ operator might not even have a least fixed point for some contextual programs. However, we can show that the Φ operator does always have a least fixed point if we restrict contextual programs to the class of acyclic ones. After that, we define contextual abduction and show that the Tweety example from the introduction leads to the desired results. Furthermore, we can now specify the relations between observations and explanations under contextual abduction, allowing us to define notions pertaining to contextual side-effects.

Some open questions are left to be investigated in the future. For instance, can the requirements for the classes of acyclic contextual programs be relaxed to those that are only acyclic with respect to the truth functional operator `ctxt`, so that the Φ operator is still guaranteed to yield a fixed point? Furthermore, as the Weak Completion Semantics seems to adequately model human reasoning, a natural question to ask is whether contextual reasoning can help us model pertinent psychological experiments. For this purpose, we are particularly interested in psychological findings that deal with context sensitive information.

Acknowledgements LMP acknowledges support from FCT/MEC NOVA LINCS Pest UID/CEC/04516/2013. Many thanks to Tobias Philipp and Christoph Wernhard.

References

1. S. Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fund. Math.*, 3:133–181, 1922.
2. R. Byrne. Suppressing valid inferences with conditionals. *Cognition*, 31:61–83, 1989.
3. K. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Databases*, pages 293–322. Plenum, New York, 1978.
4. E.-A. Dietz and S. Hölldobler. A new computational logic approach to reason with conditionals. In F. Calimeri, G. Ianni, and M. Truszczynski, eds., *Logic Programming and Nonmonotonic Reasoning, 13th International Conference, LPNMR*, vol. 9345 of *Lecture Notes in Artificial Intelligence*, pages 265–278. Springer, 2015.
5. E.-A. Dietz, S. Hölldobler, and R. Höps. A computational logic approach to human spatial reasoning. In *IEEE Symposium Series on Computational Intelligence*, pages 1637–1634, 2015.
6. E.-A. Dietz, S. Hölldobler, and L. M. Pereira. On conditionals. In G. Gottlob, G. Sutcliffe, and A. Voronkov, editors, *Global Conference on Artificial Intelligence*, volume 36 of *Epic Series in Computing*, pages 79–92. EasyChair, 2015.
7. E.-A. Dietz, S. Hölldobler, and M. Ragni. A computational logic approach to the suppression task. In N. Miyake, D. Peebles, and R. P. Cooper, editors, *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, pages 1500–1505. Cognitive Science Society, 2012.
8. E.-A. Dietz, S. Hölldobler, and M. Ragni. A computational logic approach to the abstract and the social case of the selection task. In *Proc. Eleventh International Symposium on Logical Formalizations of Commonsense Reasoning*, 2013. `commonsensereasoning.org/2013/proceedings.html`.

9. E.-A. Dietz, S. Hölldobler, and C. Wernhard. Modelling the suppression task under weak completion and well-founded semantics. *Journal of Applied Non-Classical Logics*, 24:61–85, 2014.
10. M. Fitting. Metric methods – three examples and a theorem. *Journal of Logic Programming*, 21(3):113–127, 1994.
11. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R. Kowalski and K. Bowen, editors, *Proceedings of the International Joint Conference and Symposium on Logic Programming*, pages 1070–1080. MIT Press, 1988.
12. S. Hölldobler and C. D. P. Kencana Ramli. Contraction properties of a semantic operator for human reasoning. In L. Li and K. K. Yen, editors, *Proceedings of the Fifth International Conference on Information*, pages 228–231. International Information Institute, 2009.
13. S. Hölldobler and C. D. P. Kencana Ramli. Logic programs under three-valued Łukasiewicz's semantics. In P. M. Hill and D. S. Warren, editors, *Logic Programming*, volume 5649 of *Lecture Notes in Computer Science*, pages 464–478. Springer-Verlag Berlin Heidelberg, 2009.
14. S. Hölldobler and C. D. P. Kencana Ramli. Logics and networks for human reasoning. In C. Alippi, M. M. Polycarpou, C. G. Panayiotou, and G. Ellinasetal, editors, *Artificial Neural Networks – ICANN*, volume 5769 of *Lecture Notes in Computer Science*, pages 85–94. Springer-Verlag Berlin Heidelberg, 2009.
15. S. Hölldobler, T. Philipp, and C. Wernhard. An abductive model for human reasoning. In *Proc. Tenth International Symposium on Logical Formalizations of Commonsense Reasoning*, 2011. `commonsensereasoning.org/2011/proceedings.html`.
16. A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive Logic Programming. *Journal of Logic and Computation*, 2(6):719–770, 1993.
17. C. D. P. Kencana Ramli. Logic programs and three-valued consequence operators. Master's thesis, International Center for Computational Logic, TU Dresden, 2009.
18. J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, 1984.
19. J. Łukasiewicz. O logice trójwartościowej. *Ruch Filozoficzny*, 5:169–171, 1920. English translation: On Three-Valued Logic. In: *Jan Łukasiewicz Selected Works*. (L. Borkowski, ed.), North Holland, 87-88, 1990.
20. L. M. Pereira, J. N. Aparício , and J. Alferes. Hypothetical reasoning with well founded semantics. In B. Mayoh, editor, *Proceedings of the 3th Scandinavian Conference on AI*, pages 289–300. IOS Press, 1991.
21. L. M. Pereira, E.-A. Dietz, and S. Hölldobler. An abductive reasoning approach to the belief-bias effect. In C. Baral, G. D. Giacomo, and T. Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 14th International Conference*, pages 653–656, Cambridge, MA, 2014. AAAI Press.
22. L. M. Pereira, E.-A. Dietz, and S. Hölldobler. Contextual abductive reasoning with side-effects. In I. Niemelä, ed., *Theory and Practice of Logic Programming (TPLP)*, vol. 14, pages 633–648, Cambridge, UK, 2014. Cambridge University Press.
23. L. M. Pereira and A. Pinto. Inspecting side-effects of abduction in logic programming. In M. Balduccini and T. Son, eds., *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning: Essays in Honour of Michael Gelfond*, vol. 6565 of *Lecture Notes in Artificial Intelligence*, pages 148–163. Springer, 2011.
24. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81 – 132, 1980.
25. K. Stenning and M. van Lambalgen. *Human Reasoning and Cognitive Science*. MIT Press, 2008.
26. A. van Gelder, K. A. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38:620–650, 1991.