Contextual Abduction and its Complexity Issues

Emmanuelle-Anna Dietz Saldanha[1], Steffen Hölldobler[1,2], and Tobias Philipp[1*]

[1]International Center for Computational Logic, TU Dresden, Germany
[2]North-Caucasus Federal University, Stavropol, Russian Federation
{dietz,sh}@iccl.tu-dresden.de, tobias.philipp@tu-dresden.de

Abstract. In everyday life, it seems that we prefer some explanations for an observation over others because of our contextual background knowledge. Reiter already tried to specify a mechanism within logic that allows us to avoid explicitly considering all exceptions in order to derive a conclusion w.r.t. the usual case. In a recent paper, a contextual reasoning approach has been presented, which takes this contextual background into account and allows us to specify contexts within the logic. This approach is embedded into the Weak Completion Semantics, a Logic Programming approach that aims at adequately modeling human reasoning tasks. As this approach extends the underlying three-valued Łukasiewicz logic, some formal properties of the Weak Completion Semantics do not hold anymore. In this paper, we investigate the effects of this extension and present some surprising results about the complexity issues of contextual abduction.

1 Introduction

Consider the following scenario, extended and discussed in [10]:

> *If the brakes are pressed, then the car slows down. If the brakes are not OK, then car does not slow down. If the car accelerates, then the car does not slow down. If the road is slippery, then the car does not slow down. If the road is icy, then the road is slippery. If the road is downhill, then the car accelerates. If the car has snow chains on the wheels, then the road is not slippery for the car. If the car has snow chains on the wheels and the brakes are pressed, then the car does not accelerate when the road is downhill.*

[11] proposed to introduce licenses for inferences when modeling conditionals in human reasoning. [10] suggested to make these conditionals exception-tolerant in logic programs, by modeling the first conditional in the scenario above as *If the brakes are pressed and nothing abnormal is the case, then the car slows down*. Accordingly, we apply this idea to all conditionals in the previous scenario:

> *If the brakes are pressed (press) and nothing abnormal is the case ($\neg ab_1$), then the car slows down (slow_down). If the brakes are not OK ($\neg$brakes_ok), then something abnormal is the case w.r.t. ab_1. If the car accelerates (accelerate), then something abnormal is the case w.r.t. ab_1. If the road is slippery (slippery),*

* The authors are mentioned in alphabetical order.

> *then something abnormal is the case w.r.t.* ab_1*. If the road is icy (icy_road) and nothing abnormal is the case (*ab_2*), then the road is slippery. If the road is downhill (downhill) and nothing abnormal is the case (*ab_3*), then the car accelerates (accelerate). If the car has snow chains (snow_chain), then something abnormal is the case w.r.t.* ab_2*. If the car has snow chains (snow_chain) and the brakes are pressed (press), then something abnormal is the case w.r.t.* ab_3*.*

According to [10], when reasoning in such a scenario, abnormalities should be ignored, unless there is some reason to assume them to be true. As already observed and questioned by Reiter [9], the issue is whether it is possible to specify a logic-based mechanism that allows us to avoid explicitly considering all exceptions in order to derive a conclusion w.r.t. the usual case.

In this paper, we aim at modeling this idea within a logic programming approach, the *Weak Completion Semantics* (WCS) [3] and with the help of contextual reasoning [2]. WCS originates from [11], which unfortunately had some technical mistakes. These were corrected in [4] by using the three-valued Łukasiewicz logic. Since then, WCS has been successfully applied to various human reasoning tasks,summarized in [3]. [1] shows the correspondence between WCS and the Well-founded Semantics [12] and that the Well-founded Semantics does not adequately model Byrne's suppression task.

As has been shown in [2] modeling the famous Tweety example [9] under the Weak Completion Semantics leads to undesired results, namely that all exception cases have to be stated explicitly false. [2] proposes to extend the underlying three-valued Łukasiewicz Semantics and presents a contextual abductive reasoning approach. The above scenario is similar to Reiter's goal when he discussed the Tweety example, in the sense that it describes exceptions, which we don't want to explicitly consider.

Consider $\mathcal{P}_{car}$, representing the previous described scenario, including abnormalites:

$$\begin{array}{rlrl} slow\_down & \leftarrow press \wedge \neg ab_1. & slippery & \leftarrow icy\_road \wedge \neg ab_2. \\ ab_1 & \leftarrow slippery. & accelerate & \leftarrow downhill \wedge \neg ab_3. \\ ab_1 & \leftarrow \neg brakes\_ok. & ab_2 & \leftarrow snow\_chain. \\ ab_1 & \leftarrow accelerate. & ab_3 & \leftarrow snow\_chain \wedge press. \end{array}$$

Suppose that we observe *the brakes are pressed*, i.e. $\mathcal{O}_1 = \{press\}$: Under the WCS, we cannot derive from $\mathcal{P}_{car} \cup \mathcal{O}_1$ that *slow_down* is true, because we don't know whether ab_1 is false, which in turn cannot be derived to be false, because we don't know whether the road is slippery, the brakes are OK or the car accelerates. We need to explicitly state that *press* and *brakes_ok* are true whereas *icy_road*, *downhill* and *snow_chain* have to be assumed false such that we can derive that *slow_down* is true. However, if there is no evidence to assume that ab_1, ab_2 and ab_3 are true, we would like to assume the *usual case*, i.e. to avoid specifying explicitly that all abnormalities are not true.

Let us observe that the car does not slow down, i.e. $\mathcal{O}_2 = \{\neg slow\_down\}$. Given $\mathcal{P}_{car}$, we can either explain this observation by assuming that the brakes are not pressed, $\mathcal{E}_2 = \{press \leftarrow \bot\}$, that the road is icy, $\mathcal{E}_3 = \{icy\_road \leftarrow \top\}$, that the brakes are not OK, $\mathcal{E}_4 = \{brakes\_ok \leftarrow \bot\}$ or that the road is downhill and the car has no snow chain, $\mathcal{E}_5 = \{downhill \leftarrow \top,\ snow\_chain \leftarrow \bot\}$. We would like to express that the explanation that describes the usual case seems more likely: In this case $\mathcal{E}_2$ is the preferred explanation, as usually, when the car does not slow down, then the brakes are not pressed.

Only if there is some evidence that something abnormal is the case, i.e. if we observe that something else would suggest one of the other explanations, then some other explanation can be considered. For instance, if we observe additionally that the road is slippery, we would prefer $\mathcal{E}_3$ over the other explanations, or if we additionally observe that the road is downhill, we would prefer $\mathcal{E}_5$ over the other explanations. Let us check whether the closed world assumption (CWA) w.r.t. undefined atoms helps. Consider the program $\mathcal{P}_{car} \cup \{press \leftarrow \bot\}$:[1] *brakes_ok* is assumed to be false by CWA, which in turn makes ab_1 true, and therefore leads us to conclude that *slow_down* is false. However, in the usual case we would like to derive the contrary, namely that *slow_down* is true.

The two examples above show that neither the Weak Completion Semantics nor approaches that apply the closed world assumption w.r.t. undefined atoms can adequately model our intention. The contextual abductive reasoning approach presented in [2] proposes a way of modeling the usual case, i.e. ignoring abnormalities if there is no evidence to assume them to be true, and expressing a preference among explanations. This approach takes Pereira and Pinto's inspection points [8] in abductive logic programming as starting point. In this paper we investigate several problems in terms of complexity theory, and contrast these results with properties from abductive reasoning without context.

2 Background

We assume that the reader is familiar with logic and logic programming. The general notation and terminology is based on [6].

2.1 Contextual Logic Programs

Contextual logic programs are logic programs extended by the truth-functional operator ctxt, called *context* [2]. *(Propositional) contextual logic program clauses* are expressions of the forms $A \leftarrow L_1 \wedge \ldots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \ldots \wedge \mathsf{ctxt}(L_{m+p})$ (called *rules*), $A \leftarrow \top$ (called *facts*), and $A \leftarrow \bot$ (called *assumptions*). A is called *head* and $L_1 \wedge \ldots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \ldots \wedge \mathsf{ctxt}(L_{m+p})$ as well as $\top$ and $\bot$, standing for *true* and *false*, respectively, are called *body* of the corresponding clauses. A *contextual (logic) program* is a set of contextual logic program clauses. $\mathsf{atoms}(\mathcal{P})$ denotes the set of all atoms occurring in $\mathcal{P}$. 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 set of all atoms that are undefined in $\mathcal{P}$ is denoted by $\mathsf{undef}(\mathcal{P})$. The *definition of* A *in* $\mathcal{P}$ is defined as $\mathsf{def}(A,\mathcal{P}) = \{A \leftarrow body \mid A \leftarrow body \text{ is a rule or a fact occurring in } \mathcal{P}\}$. $\neg A$ is *assumed in* $\mathcal{P}$ iff $\mathcal{P}$ contains an assumption with head A and $\mathsf{def}(A,\mathcal{P}) = \emptyset$.

An *exception clause* is as a clause of the form $ab_j \leftarrow body$, $1 \leq j \leq m$, where $m \in \mathbb{N}$. A is an atom and the L_i with $1 \leq i \leq m+p$ are literals. Conceptually, we suggest to use the context connective within contextual programs as follows: The ctxt operator is applied upon every literal in the body of an exception clause in $\mathcal{P}$. We will omit the word 'contextual' when we refer to (logic) programs, if not stated otherwise.

[1] This notion of falsehood appears to be counterintuitive, but programs will be interpreted under (weak) completion semantics where the implication sign is replaced by an equivalence sign.

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

Consider the following transformation for a given program $\mathcal{P}$: (1) For all $A \leftarrow body_1, A \leftarrow body_2, \ldots, A \leftarrow body_n \in \mathcal{P}$, where $n \geq 1$, replace by $A \leftarrow body_1 \vee body_2 \vee \ldots \vee body_n$. (2) Replace all occurrences of $\leftarrow$ by $\leftrightarrow$. The resulting set is the *weak completion* of $\mathcal{P}$, denoted by $\mathsf{wc}\,\mathcal{P}$ [4].

Example 1 *Consider* $\mathcal{P} = \{s \leftarrow r, \quad r \leftarrow \neg p \wedge q, \quad q \leftarrow \bot, \quad s \leftarrow \top\}$. *The first two clauses are rules, the third is an assumption and the fourth is a fact. s and r are defined, whereas p and q are not defined in* $\mathcal{P}$, *i.e.* $\mathsf{undef}(\mathcal{P}) = \{p, q\}$. $\mathcal{P}$ *is acyclic, as it is acyclic w.r.t. the following level mapping:* $\ell(s) = 3, \ell(r) = 2$ *and* $\ell(p) = \ell(q) = 1$. *The weak completion of* $\mathcal{P}$ *is* $\mathsf{wc}\,\mathcal{P} = \{s \leftrightarrow r \vee \top,\ r \leftrightarrow \neg p \wedge q,\ q \leftrightarrow \bot\}$.

2.2 Three-Valued Łukasiewicz Logic Extended by the Context Connective

We consider the three-valued Łukasiewicz logic, 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 $\{\top, \bot, \mathrm{U}\}$, and is represented as a pair $I = \langle I^\top, I^\bot \rangle$ of two disjoint sets of atoms, where $I^\top = \{A \mid I(A) = \top\}$ and $I^\bot = \{A \mid I(A) = \bot\}$. Atoms which do not occur in $I^\top \cup I^\bot$ are mapped to U. The truth value of a given formula under I is determined according to the truth tables in Table 1. A *three-valued model* $\mathcal{M}$ of $\mathcal{P}$ is a three-valued interpretation such that $\mathcal{M}(A \leftarrow body) = \top$ for each $A \leftarrow body \in \mathcal{P}$. Let $I = \langle I^\top, I^\bot \rangle$ and $J = \langle J^\top, J^\bot \rangle$ be two interpretations. $I \subseteq J$ iff $I^\top \subseteq J^\top$ and $I^\bot \subseteq J^\bot$. I is a *minimal model* of $\mathcal{P}$ iff for no other model J of $\mathcal{P}$ it holds that $J \subseteq I$. I is the *least model* of $\mathcal{P}$ iff it is the only minimal model of $\mathcal{P}$. Example 2 shows the models of the program in Example 1.

Our suggestion to apply the ctxt operator upon every literal in the body of an exception clause in $\mathcal{P}$ with Table 1 implements the idea of the introduction that abnormalities should be assumed to be false, if there is no evidence to assume otherwise.

Example 2 $I_1 = \langle \{s\}, \{q, r\} \rangle$, $I_2 = \langle \{s, p\}, \{q, r\} \rangle$ *and* $I_3 = \langle \{s, q, r\}, \{p\} \rangle$ *are models of* $\mathcal{P}$. I_1 *is the least model of* $\mathsf{wc}\,\mathcal{P}$ *and* I_3 *is not a model of* $\mathsf{wc}\,\mathcal{P}$.

2.3 Stenning and van Lambalgen Consequence Operator

We reason w.r.t. the Stenning and van Lambalgen consequence operator $\Phi_{\mathcal{P}}$ [11, 4]: Let I be an interpretation and $\mathcal{P}$ be a program. The application of Φ to I and $\mathcal{P}$, denoted by $\Phi_{\mathcal{P}}(I)$, is the interpretation $J = \langle J^\top, J^\bot \rangle$:

$$J^\top = \{A \mid \text{there is } A \leftarrow body \in \mathcal{P} \text{ such that } I(body) = \top\},$$
$$J^\bot = \{A \mid \text{there is } A \leftarrow body \in \mathcal{P} \text{ and for all } A \leftarrow body \in \mathcal{P},\ \text{we find } I(body) = \bot\}.$$

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	$\bot$

$\leftarrow$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	$\top$	$\top$
$\bot$	$\bot$	U	$\top$

$\leftrightarrow$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	$\top$	U
$\bot$	$\bot$	U	$\top$

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

Table 1. The truth tables for the connectives under the three-valued Łukasiewicz logic and for $\mathsf{ctxt}(L)$. L is a literal, $\top$, $\bot$, and U denote *true*, *false*, and *unknown*, respectively.

The least fixed point of $\Phi_{\mathcal{P}}$ is denoted by $\mathsf{lfp}\ \Phi_{\mathcal{P}}$, if it exists. Acyclic programs admit several nice properties: The $\Phi_{\mathcal{P}}$ operator is a contraction, has a least fixed point[2] that can be reached by iterating a finite number of times starting from any interpretation, and $\mathsf{lfp}\ \Phi_{\mathcal{P}}$ is a model of $\mathcal{P}$ [2]. We define $\mathcal{P} \models_{wcs} F$ iff $\mathcal{P}$ is acyclic and $\mathsf{lfp}\ \Phi_{\mathcal{P}} \models F$.

As has been shown in [4], for non-contextual programs, the least fixed point of $\Phi_{\mathcal{P}}$ is identical to the least model of the weak completion of $\mathcal{P}$, which always exists. As Example 3 shows this does not hold for contextual programs: The weak completion of contextual programs might have more than one minimal model.

Example 3 *$\mathcal{P} = \{s \leftarrow \neg r,\ r \leftarrow \neg p \wedge q,\ q \leftarrow \mathsf{ctxt}(\neg p)\}$ then $\mathsf{wc}\,\mathcal{P} = \{s \leftrightarrow r,\ r \leftrightarrow \neg p \wedge q,\ q \leftrightarrow \mathsf{ctxt}(\neg p)\}$. $\mathsf{lfp}\ \Phi_{\mathcal{P}} = \langle \{s\}, \{q, r\} \rangle$. $\langle \{q, r\}, \{p, s\} \rangle$ is also a minimal model of $\mathsf{wc}\,\mathcal{P}$.*

However, a minimal model that is different to the least fixed point of $\Phi_{\mathcal{P}}$, is not supported in the sense that if we iterate $\Phi_{\mathcal{P}}$ starting with this minimal model, then we will compute $\mathsf{lfp}\ \Phi_{\mathcal{P}}$. As $\mathsf{lfp}\ \Phi_{\mathcal{P}}$ is unique and the only supported minimal model of $\mathsf{wc}\,\mathcal{P}$, we define $\mathcal{P} \models_{wcs} F$ if and only if F holds in the least fixed point of $\Phi_{\mathcal{P}}$.

2.4 Complexity Classes

A *decision problem* is a problem where the answer is either *yes* or *no*. A natural correspondence to the decision problem is the word problem, where the word problem deals with the question *Does word w belong to language L*? Here, a word is a finite string over the alphabet Σ and a language is a possibly infinite set of words over Σ, where Σ^* denotes every word over Σ.

P is the class of decision problems that are solvable in polynomial time. NP is the class of decision problems, where the *yes* answers can be verified in polynomial time. Every decision problem can be specified by a language and every class can be specified by the set of languages it contains: P is the set of languages, for which it can be decided in polynomial time whether they are in P. Let R be a binary relation on strings. R is *balanced* if $(x, y) \in R$ implies $|y| \leq |x|^k$ for some $k \geq 1$. Let $L \subseteq \Sigma^*$ be a language. $L \in$ NP iff there is a polynomially decidable and a polynomial balanced relation R such that $L = \{x \mid (x, y) \in R$ for some $y\}$ [7]. That is, NP is the set of languages, for the ones that belong to this class, it can be decided in polynomial time whether they are in NP. Given that CONP $= \{\overline{L} \mid L \in$ NP$\}$, a language L is in the class DP iff there are two languages $L_1 \in$ NP and $L_2 \in$ CONP such that $L = L_1 \cap L_2$. PSPACE is the

[2] Note that for acyclic programs, the least fixed point of $\Phi_{\mathcal{P}}$ is also the unique fixed point of $\Phi_{\mathcal{P}}$.

set of languages for which it can be decided in polynomial space whether they are in PSPACE. The relation of the four classes is P $\subseteq$ NP $\subseteq$ DP $\subseteq$ PSPACE.

A language L is *polynomial-time reducible* to a language L', denoted as $L \leq_p L'$ if there is a polynomial-time computable function $f : \Sigma^* \mapsto \Sigma^*$ such that for every $x \in \Sigma^*$, $x \in L$ iff $f(x) \in L'$. Reductions are transitive, i.e. if $L_1 \leq_p L_2$ and $L_2 \leq_p L_3$ then $L_1 \leq_p L_3$ for all languages L_1, L_2 and L_3. Given that **C** is a complexity class, we say that a language L is **C**-*hard* if $L \leq_p L'$ for all $L' \in$ **C**. L is **C**-complete if L is in **C** and L is **C**-*hard*.

3 Abduction in Contextual Logic Programs

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

$$\begin{aligned} &\{A \leftarrow \top \mid A \text{ is undefined in } \mathcal{P} \text{ or } A \text{ is head of an exception clause in } \mathcal{P}\} \\ \cup\ &\{A \leftarrow \bot \mid A \text{ is undefined in } \mathcal{P} \text{ and } \neg A \text{ is not assumed}^3 \text{ in } \mathcal{P}\}, \end{aligned}$$

Let an *observation* O be a non-empty set of literals. Abductive reasoning can be characterized as the problem to find an explanation $\mathcal{E} \subseteq \mathcal{A}$ such that O can be inferred by $\mathcal{P} \cup \mathcal{E}$ by deductive reasoning. Often, explanations are restricted to be *basic* and that they are consistent with $\mathcal{P}$. An explanation $\mathcal{E}$ is *basic*, if $\mathcal{E}$ cannot be explained by other facts or assumptions, i.e. $\mathcal{E}$ can only be explained by itself.[4] It is easy to see that given an acyclic logic program $\mathcal{P}$ and that $\mathcal{E} \subseteq \mathcal{A}$, the resulting program $\mathcal{P} \cup \mathcal{E}$ is acyclic as well. Further, as the $\Phi_{\mathcal{P}}$ operator always yields a least fixed point for acyclic programs, $\mathcal{P} \cup \mathcal{E}$ is guaranteed to be consistent. We will impose a further restriction on explanations such that explanations do not allow to change the context of the observation. Formally, this is defined using the following relation:

Definition 4. *The* strongly depends on – relation *w.r.t.* $\mathcal{P}$ *is the smallest transitive relation with the following properties:*

1. *If* $A \leftarrow L_1 \wedge \ldots \wedge L_m \wedge \mathsf{ctxt}(L_{m+1}) \wedge \ldots \wedge \mathsf{ctxt}(L_{m+p}) \in \mathcal{P}$*, then* A *strongly depends on* L_i *for all* $i \in \{1, \ldots, m\}$.
2. *If* L *strongly depends on* L'*, then* $\neg L$ *strongly depends on* L'.
3. *If* L *strongly depends on* L'*, then* L *strongly depends on* $\neg L'$. □

Example 5 *Given* $\mathcal{P} = \{p \leftarrow r, p \leftarrow \mathsf{ctxt}(q)\}$*,* p *strongly depends on* r *and* $\neg r$*,* $\neg p$ *strongly depends on* r *and* $\neg r$*.* p *does not strongly depend on* q*, neither on* $\mathsf{ctxt}(q)$*.*

We formalize the abductive reasoning process as follows:

Definition 6. *Given the contextual abductive framework* $\langle \mathcal{P}, \mathcal{A}, \models_{wcs} \rangle$ $\mathcal{E}$ *is a contextual explanation of* O *given* $\mathcal{P}$ *iff* $\mathcal{E} \subseteq \mathcal{A}$*,* $\mathcal{P} \cup \mathcal{E} \models_{wcs} O$*, and for all* $A \leftarrow \top \in \mathcal{E}$ *and* $A \leftarrow \bot \in \mathcal{E}$ *there exists an* $L \in O$*, such that* L *strongly depends on* A*.*

[3] It is not the case that $A \leftarrow \bot \in \mathcal{P}$ and this is the only clause where A is the head of in $\mathcal{P}$.

[4] If $\mathcal{P} = \{p \leftarrow q\}$ and $O = \{q\}$, then $\mathcal{E} = \{q \leftarrow \top\}$ is basic.

In the following, we abbreviate the contextual abductive framework, by referring to the *abductive problem* $\mathcal{AP} = \langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$. $\mathcal{E}$ is an explanation for the abductive problem $\mathcal{AP} = \langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$ iff $\mathcal{E}$ is a contextual explanation of $\mathcal{O}$ given $\mathcal{P}$.

Notice that $\mathcal{P} \cup \mathcal{E}$ is consistent since the resulting program is acyclic, and therefore a least fixed point of $\Phi_{\mathcal{P}}$ exists. We demonstrate the formalism by Example 7.

Example 7 *Let us consider again $\mathcal{P}_{car}$ from the introduction and recall that, if we know that 'the brakes are pressed' is true i.e. $press \leftarrow \top$, then under the Weak Completion Semantics, we cannot derive from $\mathcal{P} \cup \{press \leftarrow \top\}$ that 'slow_down' is true, because we don't know whether the road is slippery, the brakes are OK or the car accelerates. Given $\mathcal{P}_{car}$, $\mathcal{P}_{car}^{\mathsf{ctxt}}$ is defined as follows:*

$$\begin{array}{rlrl} slow\_down & \leftarrow press \wedge \neg ab_1. & slippery & \leftarrow icy\_road \wedge \neg ab_2. \\ ab_1 & \leftarrow \mathsf{ctxt}(slippery). & accelerate & \leftarrow downhill \wedge \neg ab_3. \\ ab_1 & \leftarrow \mathsf{ctxt}(\neg brakes\_ok). & ab_2 & \leftarrow \mathsf{ctxt}(snow\_chain). \\ ab_1 & \leftarrow \mathsf{ctxt}(accelerate). & ab_3 & \leftarrow \mathsf{ctxt}(snow\_chain) \wedge \mathsf{ctxt}(press). \end{array}$$

By iterating $\Phi_{\mathcal{P}_{car}^{\mathsf{ctxt}}}$ until the least fixed point is reached, we obtain $\langle \emptyset, \{ab_1, ab_2, ab_3\} \rangle$. All abnormality predicates are false, as nothing is known about 'slippery', 'brakes_ok', 'accelerate' and 'snow_chain'. According to Table 1, '$\mathsf{ctxt}(slippery)$', '$\mathsf{ctxt}(brakes\_ok)$', '$\mathsf{ctxt}(accelerate)$' and '$\mathsf{ctxt}(snow\_chain)$' are evaluated to false under $\langle \emptyset, \emptyset \rangle$, which in turn makes '$ab_1$', '$ab_2$' and '$ab_3$' false. Assume that we observe $\mathcal{O}_1 = \{press\}$. A contextual explanation $\mathcal{E}_1$ for $\mathcal{O}_1$ has to be a subset of the set of abducibles $\mathcal{A}$. $\mathcal{A}$ consists of the following facts and assumptions:

$$\begin{array}{rrrr} press \leftarrow \top. & brakes\_ok \leftarrow \top. & icy\_road \leftarrow \top. & ab_1 \leftarrow \top. \\ press \leftarrow \bot. & brakes\_ok \leftarrow \bot. & icy\_road \leftarrow \bot. & ab_2 \leftarrow \top. \\ downhill \leftarrow \top. & snow\_chain \leftarrow \top. & & ab_2 \leftarrow \top. \\ downhill \leftarrow \bot. & snow\_chain \leftarrow \bot. & & \end{array}$$

$\mathcal{E}_1 = \{press \leftarrow \top\}$ is the only contextual explanation for $\mathcal{O}_1$. The least fixed point of the program together with the corresponding explanation is as follows:

$$\mathsf{lfp}\,(\Phi_{\mathcal{P} \cup \mathcal{E}_1}) = \langle \{slow\_down, press\}, \{ab_1, ab_2, ab_3\} \rangle.$$

Assume that we observe that the car does not slow down, i.e. $\mathcal{O}_2 = \{\neg slow\_down\}$. The only contextual explanation for $\mathcal{O}_2$ is $\mathcal{E}_2 = \{press \leftarrow \bot\}$. $\mathsf{lfp}\,(\Phi_{\mathcal{P} \cup \mathcal{E}_2})$ is as follows:

$$\langle \emptyset, \{slow\_down, press, ab_1, ab_2, ab_3\} \rangle,$$

and indeed this model entails '$\neg$slow_down'. Note that neither $\mathcal{E}_3 = \{icy\_road \leftarrow \top\}$ nor $\mathcal{E}_4 = \{brakes\_ok \leftarrow \bot\}$ can be contextual explanations for $\mathcal{O}_2$, because the additional condition for contextual explanations, that 'for all $A \leftarrow \top \in \mathcal{E}$ and for all $A \leftarrow \bot \in \mathcal{E}$ there exists an $L \in \mathcal{O}$, such that L strongly depends on A,' does not hold: '$\neg$slow_down' strongly depends on 'press' but it does not strongly depend on 'brakes_ok' neither does it strongly depend on 'icy_road'. Assume that we additionally observe that the road is slippery:

$$\mathcal{O}_3 = \mathcal{O}_2 \cup \{slippery\}.$$

As 'slippery' strongly depends on 'icy_road', $\mathcal{E}_3$ is a contextual explanation for $\mathcal{O}_3$.

$\mathsf{lfp}\,(\Phi_{\mathcal{P}_{car}^{\mathsf{ctxt}} \cup \mathcal{E}_3}) = \langle \{icy\_road, slippery, ab_1, \}, \{slow\_down, ab_2, ab_3\} \rangle$ *and entails both '$\neg$slow_down' and 'slippery'. $\mathcal{E}_3$ is the only contextual explanation for $\mathcal{O}_3$.*

Example 8 *Let us extend the scenario from the introduction as follows:*

> *If the engine shaft rotates (rotate_E) and nothing is abnormal (ab_4), then the wheels rotate (rotate_W). If the clutch is not pressed (¬press_clutch), then something abnormal is the case w.r.t. ab_4. If the wheels rotate (rotate_W) and nothing is abnormal (ab_4), then the shaft rotates. If the wheels rotate then something is the case w.r.t. ab_1. If the wheels engine shaft rotates then something is the case w.r.t. ab_1.*

$\mathcal{P}_{car}^{\mathsf{ctxt}}$ *is extended by the following clauses:*

$$\begin{array}{ll} rotate\_W \leftarrow rotate\_E \wedge \neg ab_4. & ab_4 \leftarrow \mathsf{ctxt}(\neg press\_clutch). \\ rotate\_E \leftarrow rotate\_W \wedge \neg ab_4. & ab_1 \leftarrow \mathsf{ctxt}(rotate\_W). \\ & ab_1 \leftarrow \mathsf{ctxt}(rotate\_E). \end{array}$$

Even though there is a cycle in this program, by iterating Φ w.r.t. this program, the following least fixed point is reached: $\langle \emptyset, \{ab_1, ab_2, ab_3, ab_4\}\rangle$.

Example 8 shows that some programs with cycles still can reach a least fixed point. We assume that acyclicity only needs to be restricted w.r.t. the literals within ctxt.

4 Complexity of Consistency of Contextual Abductive Problems

A contextual abductive problem $\mathcal{A} = \langle \mathcal{P}, \mathcal{A}, \mathcal{O}\rangle$ is *consistent* if there is an explanation for $\mathcal{A}$. We will now investigate the complexity of deciding consistency. First, we show that computing the least fixed point of $\Phi_{\mathcal{P}}$ for acyclic contextual programs can be done in polynomial time. From this, we can easily show that consistency is in NP. Hardness follows analogously to [5].

For showing that $\Phi_{\mathcal{P}}$ can be computed in polynomial time, observe that several nice properties of $\Phi_{\mathcal{P}}$ do not hold if we consider contextual programs. For instance, for logic programs that do not contain the context connective, ctxt, the least fixed point of $\Phi_{\mathcal{P}}$ is monotonously increasing if we add facts and assumptions whose head is undefined. As the following example demonstrates, this does not hold for contextual programs.

Example 9 *Consider $\mathcal{P} = \{p \leftarrow \mathsf{ctxt}(r)\}$ where $\mathsf{lfp}\,\Phi_{\mathcal{P}} = \langle \emptyset, \{p\}\rangle$.*
However, $\langle \emptyset, \{p\}\rangle \not\subseteq \langle \{r, p\}, \emptyset\rangle = \mathsf{lfp}\,\Phi_{\mathcal{P}\cup\{r\leftarrow\top\}}$.

$\Phi_{\mathcal{P}}$ is non-monotonic even for acyclic programs as the following example demonstrates:

Example 10 *Given $\mathcal{P} = \{p \leftarrow \mathsf{ctxt}(q)\}$, $I_1 = \langle \emptyset, \emptyset\rangle \subseteq \langle \emptyset, \{p, q\}\rangle = I_2$, and $F = \{q \leftarrow \top\}$. Then $\Phi_{\mathcal{P}}(I_1) = \langle \emptyset, \{p\}\rangle \subseteq \langle \{q\}, \{p\}\rangle = \Phi_{\mathcal{P}\cup F}(I_2)$.*
However, $\mathsf{lfp}\,\Phi_{\mathcal{P}}(I_1) = \langle \emptyset, \{p\}\rangle \not\subseteq \langle \{p, q\}, \emptyset\rangle = \mathsf{lfp}\,\Phi_{\mathcal{P}\cup F}(I_2)$. □

We can establish a weak form of monotonicity for a logic program $\mathcal{P}$ that is acyclic w.r.t. ℓ: If the atom A is true (false, resp.) after the nth application of $\Phi_{\mathcal{P}}$ starting from the empty interpretation, and $\ell(A) \leq n$, then A remains true (false, resp.). We define $\Phi_{\mathcal{P}} \uparrow 0 = \langle \emptyset, \emptyset\rangle$ and $\Phi_{\mathcal{P}} \uparrow (n+1) = \Phi_{\mathcal{P}}(\Phi_{\mathcal{P}} \uparrow n)$ for all $n \in \mathbb{N}$.

Lemma 11. *Let $\mathcal{P}$ be a logic program that is acyclic w.r.t. a level mapping ℓ. Let $I_n = \langle I_n^\top, I_n^\perp \rangle = \Phi_\mathcal{P} \uparrow n$ for all $n \in \mathbb{N}$. If $n < m$, then:*

$$I_n^\top \cap \{A \mid \ell(A) \le n\} \subseteq I_m^\top \quad \textit{and} \quad I_n^\perp \cap \{A \mid \ell(A) \le n\} \subseteq I_m^\perp.$$

Proof. We show the claim by induction on n. For the induction base case, this is straightforward as $I_0^\top = I_0^\perp = \emptyset$. For the induction step, assume the claim holds for n:

$$I_n^\top \cap \{A \mid \ell(A) \le n\} \subseteq I_m^\top \quad \text{for all } m \in \mathbb{N} \text{ with } n < m, \tag{1}$$
$$I_n^\perp \cap \{A \mid \ell(A) \le n\} \subseteq I_m^\perp \quad \text{for all } m \in \mathbb{N} \text{ with } n < m. \tag{2}$$

– To show: $I_{n+1}^\top \cap \{A \mid \ell(A) \le n+1\} \subseteq I_k^\top$ for all $k \in \mathbb{N}$ with $n+1 \le k$.
  1. We show it by contradiction, i.e. assume that i) $A \in I_{n+1}^\top$, ii) $\ell(A) \le n+1$ and iii) $A \notin I_k^\top$.
  2. As i), there is $A \leftarrow body \in \mathcal{P}$ with the property that $I_n(body) = \top$.
  3. As $\mathcal{P}$ is acyclic, $\ell(L) < \ell(A)$ for all literals L appearing in *body*. For all L the following holds:
     (a) if $L = B$, then $B \in I_n^\top$ and as ii) $\ell(B) < n$, by (1), $B \in I_{k-1}^\top$.
     (b) if $L = \neg B$, then $B \in I_n^\perp$ and as ii) $\ell(B) < n$, by (2), $B \in I_{k-1}^\perp$.
  4. By 3a and 3b follows that $I_{k-1}(body) = \top$. Accordingly, $A \in I_k^\top$ which contradicts iii).

– To show: $I_{n+1}^\perp \cap \{A \mid \ell(A) \le n+1\} \subseteq I_k^\perp$ for all $k \in \mathbb{N}$ with $n+1 \le k$.
  1. Again, we show by contradiction, i.e. assume that i) $A \in I_{n+1}^\perp$, ii) $\ell(A) \le n+1$ and iii.) $A \notin I_k^\perp$.
  2. As i), there is $A \leftarrow body \in \mathcal{P}$, and we find that $I_n(body) = \perp$ for all $A \in body \in \mathcal{P}$. As $\mathcal{P}$ is acyclic, $\ell(L) < \ell(A)$ for all literals L appearing in *body*. For at least one L in each *body* the following holds:
     (a) if $L = B$, then $B \in I_n^\perp$ and as ii) $\ell(B) < n$, by (1), $B \in I_{k-1}^\perp$.
     (b) if $L = \neg B$, then $B \in I_n^\top$ and as ii) $\ell(B) < n$, by (2), $B \in I_{k-1}^\top$.
  3. By 2a and 2b follows that $I_{k-1}(body) = \perp$ for all $A \in body \in \mathcal{P}$. Accordingly, $A \in J_k^\perp$ which contradicts iii).

Proposition 12. *Computing* $\mathsf{lfp}\, \Phi_\mathcal{P}$ *can be done in polynomial time for acyclic logic programs $\mathcal{P}$.*

Proof. By [2, Corollary 4] the least fixed point can be obtained from finite applications of $\Phi_\mathcal{P}$, i.e. there is n such that $\Phi_\mathcal{P} \uparrow n = \Phi_\mathcal{P} \uparrow m$ for all $m > n$. We show that n is polynomially restricted in $\mathcal{P}$ as follows: The number of atoms appearing in $\mathcal{P}$ is polynomially restricted in the length of the string $\mathcal{P}$. Consequently, we can assume a maximum level m such that $\ell(A) \le m$ for all atoms A appearing in $\mathcal{P}$. We now compute $\Phi_\mathcal{P} \uparrow m$ which can be done in polynomial time. By Lemma 11, we know that $\Phi_\mathcal{P}$ is monotonic after m steps. Afterwards, we can add only polynomially many atoms to $I^\top$ or $I^\perp$ using $\Phi_\mathcal{P}$. Hence, after polynomial iterations, we have reached the least fixed point.

Theorem 13. *Deciding, whether a contextual problem $\langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$ has an explanation is* NP-complete.

Proof. We show that the problem belongs to NP, and afterwards we show NP-*hard*.

To show NP-membership, observe that explanations are polynomially bounded by the abductive framework. Then, showing NP-membership only requires to show that checking whether a set $\mathcal{E}$ is an explanation. This is done as follows:

1. $\mathcal{E}$ is a consistent subset of $\mathcal{A}$: This can be done in polynomial time [5].
2. $\mathcal{P} \cup \mathcal{E} \models_{wcs} \mathcal{O}$: Computing $\mathcal{M} = \mathsf{lfp}\, \Phi_{\mathcal{P} \cup \mathcal{E}}$ can be done in polynomial time (Proposition 12). The last step is to check whether $\mathcal{P} \cup \mathcal{E} \models_{wcs} L$ for all $L \in \mathcal{O}$, can be done as follows. For all literals $L \in \mathcal{O}$, if $L = A$, then check if $A \in I^\top$ and if $L = \neg A$, then check if $A \in I^\perp$
3. for all $A \leftarrow \top \in \mathcal{E}$ and for all $A \leftarrow \perp \in \mathcal{E}$, respectively, there exists $L \in \mathcal{O}$ such that L strongly depends on $A \leftarrow \top$ and $A \leftarrow \perp$, respectively: The strongly depends on relation for every two literals can be checked in $|\mathcal{P}|$ steps, and thus the computation can be done in polynomial time.

It remains to show that consistency is NP-*hard*. As already consistency with no context connective is NP-*hard* [5], it easily follows that consistency is NP-*hard*.

5 Complexity of Skeptical Reasoning with Abductive Explanations

We are not only interested in deciding whether an observation can be explained, but what can be inferred from the possible explanations. We distinguish between skeptical and credulous reasoning: Given an abductive problem $\mathcal{AP} = \langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$, *F follows skeptically from* $\mathcal{AP}$ iff $\mathcal{AP}$ is consistent, and for all explanations $\mathcal{E}$ for $\mathcal{AP}$ it holds that $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$. The formula *F follows credulously from* $\mathcal{AP}$ iff there exists an explanation $\mathcal{E}$ of $\mathcal{AP}$ and $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$.

Proposition 14. *Deciding if $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$ does not hold for all explanations $\mathcal{E}$ given $\mathcal{AP}$ is* NP-complete*.*

Proof. To show that the problem is in NP, we guess a $\mathcal{E} \subseteq \mathcal{A}$ for $\mathcal{AP}$ and check in polynomial time whether $\mathcal{E}$ is an explanation for $\mathcal{O}$ and whether $\mathcal{P} \cup \mathcal{E} \not\models_{wcs} F$. This can be done in polynomial time.
To show that the problem is NP-*hard*, we use the result from Theorem 13, by reducing consistency to the problem above, i.e. reduce the question whether a contextual problem $\langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$ has an explanation to the question whether there exists an explanation $\mathcal{E}$ such that $\mathcal{P} \cup \mathcal{E} \not\models_{wcs} \neg(A \leftarrow A)$ for all $A \in \mathsf{atoms}(\mathcal{P})$ given $\mathcal{AP}$. The correctness of the construction follows because for every interpretation I, it holds that $I \not\models \neg(A \rightarrow A)$.

Proposition 15. *Let $L \subseteq \Sigma^*$ be a language. Then L is* NP-complete *iff $\overline{L}$ is* coNP-complete*.*

Proof. See [7, Proposition 10.1].

Proposition 16. *Deciding if $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$ holds for all explanations $\mathcal{E}$ given $\mathcal{AP}$ is* coNP-complete*.*

Proof. The opposite problem is shown to be NP-*complete* by Proposition 14. By Proposition 15, deciding the above problem is coNP-*complete*.

Theorem 17. *The question, whether F follows skeptically from an abductive problem $\langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$ is DP-complete.*

Proof. We first show that the problem belongs to DP, and afterwards we show that it is DP-*hard*. Let $\mathcal{A}P = \langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$ be an abductive problem and F a formula. $\mathcal{P} \cup \mathcal{E} \models_{wcs} F$ for all explanations $\mathcal{E}$ for $\mathcal{A}P$ iff i.) $\mathcal{A}P$ is consistent and ii.) F follows from all explanations $\mathcal{E}$ for $\mathcal{A}P$.
By Theorem 13, i.) is in NP and by Proposition 16, ii.) is in coNP. Hence, deciding whether F follows skeptically from $\mathcal{A}P$ is in DP.
Let P be a decision problem in DP. P consists of two decision problems P_1 and P_2, where $P_1 \in$ NP and $P_2 \in$ coNP by the definition of the class DP. By Theorem 13, i.) is NP-*complete*, thus we know that P_1 is polynomially reducible to consistency. By Proposition 16 ii.) is coNP-*complete*, thus P_2 is polynomially reducible to it. Hence, P can be polynomially reduced to the combined problem i) and ii.). Hence, whether F follows skeptically from $\langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$ is DP-*hard*.

6 Skeptical Reasoning with Minimal Abductive Explanations

Often, one is interested in reasoning w.r.t. minimal explanations, i.e. there is no other contextual explanation $\mathcal{E}' \subset \mathcal{E}$ for an observation $\mathcal{O}$. If explanations are monotonic, i.e. the addition of further facts and assumptions are still an explanation, then checking minimality can be done in polynomial time [5]: It is enough to check that $\mathcal{E} \setminus \{A \leftarrow \bot\}$ and $\mathcal{E} \setminus \{A \leftarrow \top\}$ is not an explanation for all $A \leftarrow \top \in \mathcal{E}$ and $A \leftarrow \bot \in \mathcal{E}$. We cannot even guarantee that explanations are monotonic for logic programs without the context operator as Example 18 shows. Yet, if the set of abducibles is restricted to the set of facts and assumptions w.r.t. the undefined atoms in $\mathcal{P}$, i.e. $\mathcal{A} = \{A \leftarrow \top \mid A \in \mathsf{undef}(\mathcal{P})\} \cup \{A \leftarrow \bot \mid A \in \mathsf{undef}(\mathcal{P})\}$ then explanations are indeed monotonic [5].

Example 18 *Given $\mathcal{P} = \{p \leftarrow q \wedge r, p \leftarrow \neg q, q \leftarrow \bot\}$ and observation $\mathcal{O} = \{p\}$. $\mathcal{E}_1 = \{q \leftarrow \top, r \leftarrow \top\}$ is an explanation for $\mathcal{O}$. $\mathcal{E}_1 \supset \{q \leftarrow \top\}$ is not an explanation for $\mathcal{O}$, but $\mathcal{E}_2 = \emptyset \subseteq \{q \leftarrow \top\} \subseteq \mathcal{E}_1$ is again an explanation for $\mathcal{O}$.*

Yet, restricting the the set of abducibles, does not make explanations monotonic if we consider contextual programs, as Example 19 shows.

Example 19 *Given $\mathcal{P} = \{p \leftarrow q, p \leftarrow \mathsf{ctxt}(r)\}$ and observation $\mathcal{O} = \{p\}$. Then, $\mathcal{E} = \{q \leftarrow \top\}$ is a contextual explanation for $\mathcal{O}$, but $\{q \leftarrow \top, r \leftarrow \top\} \supset \mathcal{E}$ not anymore, because r does not strongly depend on p.*

As Example 20 shows, given that $\mathcal{E}$ is a contextual explanation for $\mathcal{O}$, we cannot simply iterate over all $A \leftarrow \top \in \mathcal{E}$ ($A \leftarrow \bot \in \mathcal{E}$, resp.) and check whether $\mathcal{E} \setminus \{A \leftarrow \top\}$ ($\mathcal{E} \setminus \{A \leftarrow \bot$, resp.) is a contextual explanation for $\mathcal{O}$. If this would be the case, then we could decide whether $\mathcal{E}$ is a minimal contextual explanation in polynomial time [5]. Instead, we might have to check all subsets of $\mathcal{E}$, for which there are $2^{|\mathcal{E}|}$ many, i.e. this might have to be done exponentially in time.

Example 20 *Given* $\mathcal{P} = \{p \leftarrow r \wedge \neg t, t \leftarrow \mathsf{ctxt}(q), t \leftarrow \mathsf{ctxt}(s), p \leftarrow r \wedge q \wedge s\}$*. Assume that* $\mathcal{O} = \{p\}$*:* $\mathcal{E}_1 = \{r \leftarrow \top\}$ *and* $\mathcal{E}_2 = \{r \leftarrow \top, q \leftarrow \top, s \leftarrow \top\}$ *are both contextual explanations for* $\mathcal{O}$*. As* $\mathcal{E}_1 \subset \mathcal{E}_2$ *holds,* $\mathcal{E}_1$ *is a minimal contextual explanation, whereas* $\mathcal{E}_2$ *is not. However, note that none of* $\mathcal{E}_2 \setminus \{r \leftarrow \top\}$*,* $\mathcal{E}_2 \setminus \{q \leftarrow \top\}$ *or* $\mathcal{E}_2 \setminus \{s \leftarrow \top\}$ *is a contextual explanation for* $\mathcal{O}$*.*

Still, we can show an upper bound for the complexity of deciding minimality:

Theorem 21. *The question, whether a set* $\mathcal{E}$ *is a minimal explanation for an abductive problem* $\langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$ *is in* PSPACE.

Proof. Given that $\langle \mathcal{P}, \mathcal{A}, \mathcal{O} \rangle$ is an abductive problem, we need to check all subsets of $\mathcal{E}$, in order to decide whether $\mathcal{E}$ is a minimal explanation for $\mathcal{O}$. As we don't need to store the subsets of $\mathcal{E}$ as soon as we have tested them, deciding whether $\mathcal{E}$ is minimal can be done polynomial in space.

7 Conclusion

This paper investigates contextual abductive reasoning, a new approach embedded within the Weak Completion Semantics. We first show with the help of an example the limitations of the Weak Completion Semantics, when we want to express the preference of the usual case over the exception cases. Furthermore, we cannot syntactically specify contextual knowledge in the logic programs as they have been presented so far.

After that, we introduce contextual programs together with contextual abduction, we show how the previous limitations can be solved. This contextual reasoning approach allows us to indicate contextual knowledge and express the preference among explanations, depending on the context.

However, as has already been shown previously in [2], some advantageous properties which hold for programs under the Weak Completion Semantics, do not hold for contextual programs. For instance, the $\Phi_{\mathcal{P}}$ operator is not necessarily monotonic. Further, if a contextual program contains a cycle, it might not even have a fixed point.

In this paper, we first show that even though $\Phi_{\mathcal{P}}$ is not monotonic, the least fixed point can still be computed in polynomial time for acyclic contextual programs. Thereafter, we show that whether an observation has a contextual explanation, is NP-complete. Furthermore, by examining the complexity of skeptical reasoning, deciding whether something follows skeptically from an observation is DP-complete. Unfortunately, explanations might not be monotonic in contextual abduction anymore, a property that holds in abduction for non-contextual programs [5]. We can however show that deciding whether a contextual explanation is minimal lies in PSPACE.

The approach discussed here brings up a number of interesting questions: In the end of Section 2.3, we have shown that the weak completion of contextual programs might have more than only one minimal model. It seems that a possible characterization for the model computed by the $\Phi_{\mathcal{P}}$ operator, is the only minimal model for which all undefined atoms in $\mathcal{P}$ are mapped to unknown. Yet, another aspect which arises from Section 6, is whether skeptical reasoning with minimal explanations is PSPACE-hard. Further, we would like to investigate how a development of a neural network perspective for reasoning with contextual programs could be done.

Acknowledgements We are grateful to the constructive comments and ideas of the three reviewers. The Graduate Academy at TU Dresden supported Tobias Philipp.

References

1. E.-A. Dietz, S. Hölldobler, and C. Wernhard. Modeling the suppression task under weak completion and well-founded semantics. *Journal of Applied Non-Classical Logics*, 24(1-2):61–85, 2014.
2. E.-A. Dietz Saldanha, S. Hölldobler, and L. M. Pereira. Contextual reasoning: Usually birds can abductively fly. In *Proceedings of the 14th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2017)*, 2017. to appear.
3. S. Hölldobler. Weak completion semantics and its applications in human reasoning. In U. Furbach and C. Schon, editors, *Proceedings of the 1st Workshop on Bridging the Gap between Human and Automated Reasoning (Bridging-2015)*, CEUR Workshop Proceedings, pages 2–16. CEUR-WS.org, 2015.
4. S. Hölldobler and C. D. Kencana Ramli. Logic programs under three-valued Łukasiewicz semantics. In P. M. Hill and D. S. Warren, editors, *Proceedings of the 25th International Conference on Logic Programming (ICLP 2009)*, volume 5649 of *LNCS*, pages 464–478, Heidelberg, 2009. Springer.
5. S. Hölldobler, T. Philipp, and C. Wernhard. An abductive model for human reasoning. In *Logical Formalizations of Commonsense Reasoning, Papers from the AAAI 2011 Spring Symposium*, AAAI Spring Symposium Series Technical Reports, pages 135–138, Cambridge, MA, 2011. AAAI Press.
6. J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag New York, Inc., New York, NY, USA, 1984.
7. C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
8. L. M. Pereira and A. M. Pinto. Inspecting side-effects of abduction in logic programs. In M. Balduccini and T. C. Son, editors, *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning: Essays in honour of Michael Gelfond*, volume 6565 of *LNAI*, pages 148–163. Springer, 2011.
9. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.
10. K. Stenning, L. Martignon, and A. Varga. Adaptive reasoning: integrating fast and frugal heuristics with a logic of interpretation. *Decision*, in press.
11. K. Stenning and M. van Lambalgen. *Human Reasoning and Cognitive Science*. A Bradford Book. MIT Press, Cambridge, MA, 2008.
12. A. Van Gelder, K. A. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):619–649, 1991.