TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute of Artificial Intelligence Knowledge Representation and Reasoning

Abduction in Logic Programming as Second-Order Quantifier Elimination

Christoph Wernhard

KRR Report 13-05

Mail to
Technische Universität Dresden
01062 Dresden

Bulk mail to
Technische Universität Dresden
Helmholtzstr. 10
01069 Dresden

Office
Room 2006
Nöthnitzer Straße 46
01187 Dresden

Internet
http://www.wv.inf.tu-dresden.de

Abduction in Logic Programming as Second-Order Quantifier Elimination

Christoph Wernhard

Technische Universität Dresden

Abstract. It is known that skeptical abductive explanations with respect to classical logic can be characterized semantically in a natural way as formulas with second-order quantifiers. Computing explanations is then just elimination of the second-order quantifiers. By using application patterns and generalizations of second-order quantification, like literal projection, the globally weakest sufficient condition and circumscription, we transfer these principles in a unifying framework to abduction with three non-classical semantics of logic programming: stable model, partial stable model and well-founded semantics. New insights are revealed about abduction with the partial stable model semantics.

1 Introduction

An abductive explanation is basically a formula X such that for given formulas F, the "background knowledge base", and G, the "observation", it holds that F and X together entail G and, in addition, X satisfies application specific further properties, for example, that it only contains symbols from a given vocabulary and that it is as weak as possible. For classical logic, the semantics of an abductive explanation in this sense can be characterized by a second-order formula as follows:

$$X \equiv \forall \mathit{SymbolsNotAllowedInTheExplanation}\ (F \rightarrow G). \qquad \text{(i)}$$

An explanation X can then be *computed* by performing *second-order quantifier elimination* on the second-order formula, that is, computing a formula which is equivalent to the given second-order formula but does not involve second-order quantifiers. If explanations are constrained to be minimal conjunctions of literals, this scheme also applies, but indirectly: the actual explanations are then obtained as the prime implicants of X. Variants of this understanding of abductive explanations are present in a number of works, e.g., [15, p312ff.],[24,7], but the relationship to second-order quantifier elimination seems to have been made explicit first in [5]. Abduction plays several important roles in logic programming, an area where it has been investigated extensively between the late 80s and the early 2000s [19,3]. Many of these approaches are oriented at deriving methods for computing explanations from methods for evaluating logic programs. Semantic characterizations, e.g., [20,8,25,1], are usually placed aside of methods, related to them by correctness properties and complexity results.

In contrast, the objective of the present work is to combine the second-order elimination approach with non-monotonic semantics of logic programming, resulting in a characterization of abductive explanations for logic programming semantics that is "constructive" in the sense that it maps the computation of explanations to problems of second-order quantifier elimination. As logic programming semantics we consider the popular stable model semantics and two related three-valued semantics, the well-founded and the partial stable model semantics[1]. We work with representations of these logic programming semantics in classical logic extended by second-order operators, based on known translations [23, Section 3.4.1][18]. Under this view, the stable model semantics appears as circumscription that is applied only to certain *occurrences* of predicates – those that are not subjected to negation as failure. Accordingly, a logic program can be represented by a classical formula where these occurrences are distinguished by special predicate names. A logic programming semantics then corresponds to a logical operator *sem* that is wrapped around a classical representation F of a program, such that $sem(F)$ expands into a formula of classical logic extended by second-order operators. The discrimination between different logic programming semantics is expressed by different such wrapping operators, allowing to embed programs considered under different semantics within a single classical formula. With respect to abduction, only a single entailment relation – classical entailment – is required, in contrast to other generic formalizations such as [8], where the discrimination is done "globally" by specific inference operators.

The link between the inherently classical second-order characterization of abductive explanations displayed above as (i) and the non-classical logic programming semantics will be provided by a lemma that states requirements under which the operators *sem* expressing non-monotonic context are "transparent" for explanations E, that is, it holds that $sem(F) \wedge E \equiv sem(F \wedge E)$. In the case of the investigated three-valued semantics, two related versions E, E' of the explanation are required, such that the established relationship is $sem(F) \wedge E \equiv sem(F \wedge E')$. To determine explanations with respect to a logic program, the abducibles, that is, the atoms that are allowed in explanations should not be submitted to the closed-world assumption, since, unless they occur in rule heads, they would then be just set to false by the non-monotonic semantics. We take this into account by using generalizations of the considered logic programming semantics that allow to specify a set of ground atoms as *open*, that is, not subjected to the closed-world assumption. These generalizations are quite straightforward: In the underlying representations of these semantics by circumscription, the open atoms just correspond to fixed – in contrast to minimized – predicate instances.

The entailment based notion of abductive explanation sketched at the beginning is called *skeptical* or *cautious*. In contrast, *credulous* or *brave* explanations, are constrained by the requirement that the background knowledge base combined with the explanation is *consistent* with observation. For the well-founded semantics every normal logic program has exactly a single model and thus both

[1] In the sense of [28,18], in contrast to contemporary work by Saccà and Zaniolo where *partial stable model* has been used for a related semantics. See [30, Introduction].

notions coincide. In this paper, we focus on the skeptical view for the other semantics. We consider finite normal ground programs, but the material should generalize to programs with disjunctive heads, negation as failure in the head, and first-order quantification, as indicated in [35,37].

As basic second-order operator we use *literal projection* [33], a generalization of predicate quantification. Its arguments make those symbols explicit that are "not quantified" and it allows, so-to-speak, to quantify just upon predicate occurrences with a specific polarity. The latter feature is used to model the considered three-valued logic programming semantics. The application pattern of second-order quantification in (i) is called *globally weakest sufficient condition (GWSC)* and specified in terms of projection. It is closely related to *weakest sufficient condition* [24,5]. Predicate quantification can be applied to express predicate circumscription [4]. We express circumscription by a dedicated second-order operator with a syntax analogous to projection [37]. We will develop the "constructive" characterizations of abductive explanations for the three considered logic programming semantics in parallel. This framework leads to clear formal conceptualizations of various subtle issues in abduction, such as notions of minimality and handling of negative facts in explanations. For abduction with the partial stable model semantics, the author is not aware of another thorough formal treatment. A distinguishing feature of that semantics is that it can be applied to deliver meaningful explanations for facts being observed as *undefined*.

This paper is an extended version of [38]. The rest of the main part is organized as follows: In Sect. 2 the background framework of classical propositional logic extended by certain second-order operators is specified. This is applied in Sect. 3 to characterize the considered logic programming semantics. In Sect. 4 definitions of abductive explanation and related concepts are given and in Sect. 5 the central concept of *globally weakest sufficient condition* is summarized. On this basis, the main results of the paper are developed in Sect. 6: Characterizations of abductive explanations and related concepts with respect to logic programming semantics as formulas with second-order operators. Related works are reviewed in Sect. 7 and possible ways to realize the approach in practice are sketched in the conclusion, Sect. 8. Proofs of the results in the main part of the paper and further investigations are provided in appendix sections.

2 Notation and Semantic Framework

Formulas, Literals, Scopes and Predicate Groups. We consider *formulas* of classical propositional logic, extended by operators for projection and circumscription. They are constructed from propositional *atoms*, truth value constants $\top, \bot$, the unary connective $\neg$, binary connectives $\wedge, \vee, \rightarrow, \leftarrow, \leftrightarrow$, as usual, and the two operators project and circ to express projection and circumscription. As meta-level notation we use n-ary versions of $\wedge$ and $\vee$. Based on the premise that the material developed here does in principle generalize to a first-order setting, we speak of propositional atoms, or synonymously Boolean variables, also as 0-ary *predicates*. A *literal* is a pair of an atom and a sign, where we write the positive (negative) literal with atom A as $+A$ ($-A$). The complement of a

literal L is denoted by $\overline{L}$. If S is a set of literals, then $\overline{S}$ denotes the set of the complements of the members of S. We call a formula that is an atom or a negated atom a *literal formula*, or, if no ambiguity arises, also briefly a *literal*. A *scope* is a set of literals. We assume a fixed propositional signature whose set of atoms is denoted by ATOMS. The sets of all literals, all positive literals, and all negative literals w.r.t. ATOMS are denoted by ALL, POS, NEG, respectively. An *atom scope* S is a scope such that $S = \overline{S}$. Since a literal is a member of an atom scope if and only if its complement is a member, as a shorthand, we represent an atom scope also just by the set of atoms of its members. To express logic programs and three-valued formulas by classical formulas we use a signature where each "original" predicate is available in different "copies", indicating whether an occurrence is subject to negation as failure or how it contributes to the three-valued reading. These "copies" are gathered into so-called *predicate groups*: In addition to the set of propositional atoms ATOMS, we assume a set of *source atoms* that play the role of atoms in other logics that we will represent in our classical framework. Each source atom A is associated with a number of corresponding atoms $A^0, \ldots, A^n \in \mathsf{ATOMS}$, where the superscripts indicate their *predicate group*. More precisely: We assume that ATOMS can be arranged as $\{A_1^0, A_1^1, \ldots, A_1^n, A_2^0, A_2^1, \ldots, A_2^n, A_3^1, \ldots, A_3^n, \ldots\}$ for some $n \geq 1$. For $k \in \{0, \ldots, n\}$, we call the set of all literals whose atom has superscript k the *predicate group* k, written just as the number k. An atom A_i^k is called the *correspondent* from group k of any atom A_i^j. Analogously we speak of correspondents of literals. An *ungrouped scope* is a scope that contains for each of its members all their correspondents. If no ambiguity arises, we write an ungrouped scope like a scope but with omitting the predicate group superscripts. For example, let $\mathsf{ATOMS} = \{p^0, p^1, q^0, q^1, r^0, r^1\}$. Then $1 = \{+p^1, +q^1, +r^1, -p^1, -q^1, -r^1\}$ is a predicate group, and $1 \cap \mathsf{POS} = \{+p^1, +q^1, +r^1\}$. The correspondent of p^1 from group 0 is p^0. An example for an ungrouped atom scope is $\{+p^0, +q^0, +p^1, +q^1, -p^0, -q^0, -p^1, -q^1\}$, which can be written as $\{p, q\}$. The atom scope $\{+p^1, +q^1, -p^1, -q^1\}$ can be written as $1 \cap \{p, q\}$.

Classical Semantics, Projection and Circumscription. An *interpretation* is a set of literals that contains for all atoms $A \in \mathsf{ATOMS}$ exactly one of $+A$ or $-A$. The satisfaction relation $\models$ between interpretations and formulas is defined with a clause for atoms and for each logical operator. For instance, for all interpretations I, scopes S, atoms A, and formulas F, G it holds that: $I \models A$ *iff* $+A \in I$; $I \models \neg F$ *iff* $I \not\models F$; $I \models F \wedge G$ *iff* $I \models F$ *and* $I \models G$; $I \models \mathsf{project}_S(F)$ *iff there is an interpretation* J *s.t.* $J \models F$ *and* $J \cap S \subseteq I$; $I \models \mathsf{circ}_S(F)$ *iff* $I \models F$ *and there is no interpretation* J *s.t.* $J \models F$ *and* $J \cap S \subset I \cap S$. Entailment and equivalence are then defined as usual: $F \models G$ *iff for all interpretations* I *it holds that if* $I \models F$ *then* $I \models G$; $F \equiv G$ *iff* $F \models G$ *and* $G \models F$.

The formula $\mathsf{project}_S(F)$ whose semantics has just been defined with the $\models$ relationship is called the *literal projection*, or briefly *projection*, of formula F onto scope S. The *forgetting* in F about S is a notational variant where the scope is considered complementary [33,21]:

$$\mathsf{forget}_S(F) \stackrel{\text{def}}{=} \mathsf{project}_{\mathsf{ALL}-S}(F). \tag{ii}$$

Combined with first-order logic, projection generalizes second-order quantification, with respect to propositional logic quantified Boolean formulas (QBFs): A QBF $\exists p\, F$ can be expressed as $\mathsf{forget}_{\{+p,-p\}}(F)$ or as $\mathsf{project}_{\mathsf{ALL}-\{+p,-p\}}(F)$. If S is an *atom* scope, the semantic definition of projection is equivalent to: $I \models \mathsf{project}_S(F)$ *iff there is an interpretation* J *s.t.* $J \models F$ *and* $J \cap S = I \cap S$. *Literal* projection also allows to express, so-to-speak, quantification upon just the positive or negative occurrences of a Boolean variable in a formula. Intuitively, the projection of a formula F onto scope S is a formula that expresses about members of S the same as F, but expresses nothing about other literals. A projection of a propositional formula is equivalent to a formula in negation normal form in which only literals in the projection scope do occur. The latter formula is a *uniform interpolant* of the original formula with respect to the scope. A naive way to construct such a uniform interpolant – or to eliminate the projection operator – is indicated by the following equivalences, where $F[p\backslash\top]$ ($F[p\backslash\bot]$) denotes formula F with all occurrences of atom p replaced by $\top$ ($\bot$): (1.) $\mathsf{forget}_{\{+p,-p\}}(F) \equiv F[p\backslash\top] \vee F[p\backslash\bot]$. (2.) $\mathsf{forget}_{\{+p\}}(F) \equiv F[p\backslash\top] \vee (\neg p \wedge F)$. (3.) $\mathsf{forget}_{\{-p\}}(F) \equiv (p \wedge F) \vee F[p\backslash\bot]$. For formulas F and scopes S we define

$$F \Subset S \;\; \textit{iff} \;\; F \equiv \mathsf{project}_S(F). \tag{iii}$$

We use the symbol $\Subset$ also when introducing variables, e.g., "let $F \Subset S$ be a formula" for "let F be a formula such that $F \Subset S$". Projection provides a semantic account for systematically replacing atoms from a given predicate group to their correspondents from another one. Let i, j be different predicate groups. We define

$$\mathsf{rename}_{i\backslash j}(F) \;\stackrel{\text{def}}{=}\; \mathsf{forget}_i(F \wedge \textstyle\bigwedge_{A^i \in \mathsf{ATOMS}}(A^j \leftrightarrow A^i)). \tag{iv}$$

If F is a propositional formula, then $\mathsf{rename}_{i\backslash j}(F)$ is equivalent to F with all occurrences of atoms from group i replaced by their correspondents from j. We define $\mathsf{rename}_{[i_1\backslash j_1,\, ...,\, i_n\backslash j_n]}(F)$ as shorthand for $\mathsf{rename}_{i_n\backslash j_n}(...(\mathsf{rename}_{i_1\backslash j_1}(F))...)$.

The circ operator has the same argument types as $\mathsf{project}$ and has also been semantically defined above. It allows to express variants of parallel predicate circumscription where the effects on each atom are controlled by a scope argument [37]. Atoms that occur just in a *positive* literal in the scope are minimized, atoms that occur just in a *negative* literal are maximized, atoms that occur in *both polarities* are fixed and atoms that do *not at all* occur in the scope are varying. Thus, if F is a formula whose atoms are in disjoint sets P, Q and Z, then the *parallel predicate circumscription of* P *in* F *with fixed* Q *and varied* Z, traditionally written as $\mathrm{CIRC}[F; P; Z]$, can be expressed as $\mathsf{circ}_{(P\cap\mathsf{POS})\cup Q}(F)$.

3 Classically Represented Logic Programming Semantics

We consider finite normal logic programs that are ground, that is, finite sets of rules of the form

$$p \leftarrow q_1,\, \ldots,\, q_m,\, \mathsf{not}\;\; r_1,\, \ldots, \mathsf{not}\;\; r_n, \tag{v}$$

where $m, n \geq 0$ and p, q_i, r_i are source atoms. The *classical representation of a normal logic program* is a classical propositional sentence, obtained from the program by forming the conjunction of its members and replacing each source atom by its representative from the indicated group as well as replacing the connectives with classical ones, according to the following schema:

$$p^0 \leftarrow q_1^0 \wedge \ldots \wedge q_m^0 \wedge \neg r_1^1 \wedge \ldots \wedge \neg r_n^1. \tag{vi}$$

Information that was expressed in (v) by the positioning of an atom in a rule head versus the negative body is now captured instead by the predicate group.

Stable Model Semantics. For abductive reasoning we consider generalizations of the established logic programming semantics that allow to specify atoms as *open*, that is, not subjected to the closed world assumption. To this end, the operators that express the logic programming semantics have aside of a classical representation of a logic program also an ungrouped atom scope as argument that specifies the open atoms.[2] The logical operator stable renders the stable model semantics: For ungrouped atom scopes O and formulas F define

$$\mathsf{stable}_O(F) \stackrel{\text{def}}{=} \mathsf{rename}_{1\setminus 0}(\mathsf{circ}_{(0\cap \mathsf{POS})\cup 1\cup O}(F)). \tag{vii}$$

The circumscription scope in this definition specifies that all atoms from group 1 as well as all open atoms are fixed, while the remaining atoms from group 0 are minimized. This characterization of stable models in terms of circumscription originates from [23, Section 3.4.1] (see also [22,35]). It is expressed here not as a formula transformation but as a logical operator that expands into a classical formula with projection (for the renaming) and circumscription. The stable operator represents the stable model semantics in the following sense: If F is the classical representation of a normal logic program and O is an ungrouped atom scope, then the stable models of the program w.r.t. O are exactly the sets of atoms obtained by taking the set of the positive literals that are from group 0 in some model of $\mathsf{stable}_O(F)$, followed by dropping their signs and group superscripts. For example, the program $\{p \leftarrow \mathsf{not}\ q\}$ has $\{p\}$ as its single stable model, which can be obtained from the models of $\mathsf{stable}_{\{\}}(p^0 \leftarrow \neg q^1) \equiv (p^0 \wedge \neg q^0)$ as described. With respect to $O = \{q\}$, the program has the two stable models $\{p\}$ and $\{q\}$, corresponding to $\mathsf{stable}_{\{q\}}(p^0 \leftarrow \neg q^1) \equiv (p^0 \wedge \neg q^0) \vee (\neg p^0 \wedge q^0)$.

Partial Stable Model Semantics. Partial stable model and well-founded semantics associate three-valued models with a logic program. Predicate groups can be applied to express the three truth values $\mathsf{F}, \mathsf{U}, \mathsf{T}$ in terms of two truth values: An interpretation I over at least all atoms of groups 0 and 1 is said to *assign* to a source atom p the three-valued truth value F *iff* $I \models (\neg p^0 \wedge \neg p^1)$, U *iff* $I \models (\neg p^0 \wedge p^1)$, and T *iff* $I \models (p^0 \wedge p^1)$. The remaining possibility $I \models (p^0 \wedge \neg p^1)$ does not correspond to a three-valued truth value and models with this combination can be excluded with the axiom

$$\mathsf{cons} \stackrel{\text{def}}{=} \textstyle\bigwedge_{A^0 \in \mathsf{ATOMS}} (A^1 \leftarrow A^0), \tag{viii}$$

[2] It is well-know that specifying atoms as *open* in this sense can also be encoded in the standard versions of these semantics (see discussion of [17] in Sect. 7).

assuming ATOMS is finite. The logical operator pstable defined below renders the partial stable model semantics [29] by combining the translation of [18] into programs with stable models semantics with the translation of the stable model semantics shown above. Each of the two translations involves discrimination between two predicate groups, yielding four groups $0, 1, 2, 3$ in combination, which are reduced in the final value of pstable by renaming to groups 0 and 1. The models of $\mathsf{pstable}_O(F)$ represent the three-valued partial stable models by combining the values of atoms for predicate groups 0 and 1. In the definition of pstable we write the numbers denoting predicate groups in binary notation to indicate how the two involved translations are combined: The right digit corresponds to the group discrimination required by the translation into stable models, the left digit to the discrimination required by expressing the stable model semantics with circumscription. The arguments of pstable are like those of stable. For ungrouped atom scopes O and formulas F define

$$\mathsf{pstable}_O(F) \stackrel{\text{def}}{=} \mathsf{rename}_{[10\backslash 00,11\backslash 01]}(\mathsf{circ}_M(\mathsf{cons} \wedge \mathsf{rename}_{[01\backslash 11]}(F) \wedge \mathsf{rename}_{[01\backslash 10,00\backslash 01]}(F))), \tag{ix}$$

where $M = ((00 \cup 01) \cap \mathsf{POS}) \cup 10 \cup 11 \cup O$. To represent values of pstable, we write a conjunction C of literal formulas that contains for each atom $p \in \mathsf{ATOMS} \cap (0 \cup 1)$ either p or $\neg p$ as conjunct and is consistent with cons as pair $\langle \mathcal{T}, \mathcal{F} \rangle$ of two sets of source atoms, analogous to common notation for three-valued interpretations: $\mathcal{T}$ is the set of all p such that p^0 is a conjunct in C, and $\mathcal{F}$ is the set of all p such that $\neg p^1$ is a conjunct in C. For example, $(p^0 \wedge p^1 \wedge \neg q^0 \wedge \neg q^1 \wedge \neg r^0 \wedge r^1)$ would be written as $\langle \{p\}, \{q\} \rangle$. Compared to the stable model semantics, the partial stable model semantics yields additional models, caused, e.g., by atoms that are "undefined" since they are exempt from the closed world assumption or since they occur "paradoxically" in the head and negated in the body of some rule.

Example 1 (Partial Stable Model Semantics). Let $F = (p^0 \leftarrow q^0)$ and let $O = \{q\}$. Then (1) $\mathsf{stable}_O(F) \equiv (p^0 \wedge q^0) \vee (\neg p^0 \wedge \neg q^0)$ and (2) $\mathsf{pstable}_O(F) \equiv \langle \{\}, \{\} \rangle \vee \langle \{p, q\}, \{\} \rangle \vee \langle \{\}, \{p, q\} \rangle$. The first disjunct in (2), that is, $\langle \{\}, \{\} \rangle$, does not correspond to any disjunct in (1). As an example for a "paradoxical" occurrence consider $F' = (p^0 \leftarrow \neg p^1 \wedge \neg q^1)$. Then $\mathsf{stable}_{\{\}}(F') \equiv \bot$, that is, F' has no stable model. However, $\mathsf{pstable}_{\{\}}(F') \equiv \langle \{\}, \{q\} \rangle$.

Well-Founded Semantics. An interpretation is called *informationally less-or-equal-than* a second one if and only if each atom whose three-valued truth value assigned by the first interpretation is T or F has the same truth value assigned by the second one. Models that are minimal with respect to this relation can be characterized by circumscription upon the scope

$$\mathsf{imin\text{-}scope} \stackrel{\text{def}}{=} (0 \cap \mathsf{POS}) \cup (1 \cap \mathsf{NEG}). \tag{x}$$

If the models of a formula F satisfy cons, then the *informationally minimal* models of F are the models of $\mathsf{circ}_{\mathsf{imin\text{-}scope}}(F)$. If cons is used together with circumscription upon imin-scope, it can equivalently be placed inside or outside

of the circumscription operator: $\mathsf{circ}_{\mathsf{imin\text{-}scope}}(\mathsf{cons} \wedge F) \equiv \mathsf{cons} \wedge \mathsf{circ}_{\mathsf{imin\text{-}scope}}(F)$. Now, well-founded models are exactly the informationally minimal partial stable models [28], allowing to characterize the well-founded semantics as

$$\mathsf{wf}_O(F) \stackrel{\text{def}}{=} \mathsf{circ}_{\mathsf{imin\text{-}scope}}(\mathsf{pstable}_O(F)). \tag{xi}$$

An attractive feature of the well-founded semantics is that each normal logic program has exactly a single model. This property applies also to the generalized variant wf_O with specified open atoms. By the following proposition, the consequences G for which it holds that $G \Subset \mathsf{imin\text{-}scope}$ are for the well-founded semantics exactly the same as for the partial stable model semantics:

Proposition 2 (Consequences under Well-Founded and Partial Stable Model Semantics). *If O is a set of ungrouped atoms, and F, G are formulas such that $G \Subset \mathsf{imin\text{-}scope}$, then $\mathsf{wf}_O(F) \models G$ if and only if $\mathsf{pstable}_O(F) \models G$.*

The precondition $G \Subset \mathsf{imin\text{-}scope}$ of Prop. 2 is met for example by formulas $G = p^0$ and $G = \neg p^1$, which express that the truth value assigned to p is T and F, respectively, since $(\mathsf{cons} \wedge p^0) \equiv (\mathsf{cons} \wedge p^0 \wedge p^1)$ and $(\mathsf{cons} \wedge \neg p^1) \equiv (\mathsf{cons} \wedge \neg p^0 \wedge \neg p^1)$. The precondition fails for $G = (\neg p^0 \wedge p^1)$, which expresses that the value assigned to p is U.

4 Basic Concepts of Abduction

An *abductive setting* gathers the parameters of abductive reasoning problems:

Definition 3 (Abductive Setting). An *abductive setting* is a tuple $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ of (1.) a logical operator *sem* with two arguments (an ungrouped atom scope and a formula), the *programming semantics*, (2.) an ungrouped atom scope O, the *open scope*, (3.) an ungrouped scope $S \subseteq O$, the *explanation scope*, (4.) a formula F, the *background*, and (5.) a formula G, the *observation*.

This is similar to *abductive framework* [19], but here also the observation is included. The *programming semantics* is an operator like stable that specifies the logic programming semantics to be used. The *open scope* specifies the atoms that are to be considered open with respect to the logic programming semantics. The *explanation scope* specifies the vocabulary along with associated polarities that is available for explanations. It is equal to or a subset of the open scope, and thus must not necessarily be an atom scope, that is, it can contain literals but not their complements. *Background* and *observation* are formulas representing the background theory presentation and the observation, respectively.

Since we will focus on explanations that are conjunctions of literals, we provide convenient notation for these: A *conjunctive clause* is a consistent conjunction of literal formulas, with the empty conjunction $\top$ as special case. Let C, D be conjunctive clauses. We write $C \models D$ as $D \subseteq C$, and $(C \models D \text{ and } C \not\equiv D)$ as $D \subset C$. A conjunctive clause C is called *positive* (*negative*, resp.) if and only if $C \Subset \mathsf{POS}$ ($C \Subset \mathsf{NEG}$, resp.). In this paper we adhere to the skeptical view of explanations, rendered in the following definition:

Definition 4 (Explanation, Factual Explanation). Let $\mathfrak{A}=\langle sem,O,S,F,G\rangle$ be an abductive setting. An *explanation for* $\mathfrak{A}$ is a formula $H \Subset S \cap 0$ such that $sem_O(F \wedge H) \models G$. An explanation that is a conjunctive clause is called *factual*.

A positive factual explanation can be combined in a particularly simple way with a logic program: If F is the classical representation of a normal logic program and C is a positive factual explanation, then $(F \wedge C)$ is again a classical representation of a normal logic program, the original program with the positive literals of the explanation added as facts. Different ways to combine *negative* literals in explanations with programs are discussed in Sect. D. Certain abductive settings have the property that conjunctive clauses which extend a factual explanation and are in the explanation scope are also explanations, formally:

Definition 5 (Factual Explanation Monotonicity). An abductive setting $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ is called *factual explanation monotonic* if and only if whenever C is a factual explanation for $\mathfrak{A}$, then any conjunctive clause $D \Subset S \cap 0$ such that $C \subseteq D$ is also a factual explanation for $\mathfrak{A}$.

This property justifies to represent *all* factual explanations of some abductive setting compactly just by the set of *minimal factual explanations*, that is, those factual explanations that do not properly extend some other explanation:

Definition 6 (Minimal Factual Explanation). Let $\mathfrak{A}$ be an abductive setting. A *minimal factual explanation for* $\mathfrak{A}$ is a factual explanation C for $\mathfrak{A}$ such that there does not exist another factual explanation D for $\mathfrak{A}$ with $D \subset C$.

A further notion of "minimality" for factual explanations is obtained by considering just *complete* explanations, explanations that contain for each atom A^0 occurring in of the explanation scope either A^0 or $\neg A^0$, and compare them with respect to their *positive* member literals: $C \leq D$ *iff* $\mathsf{project}_{\mathsf{POS}}(D) \models \mathsf{project}_{\mathsf{POS}}(C)$. We call factual explanations that are minimal in this sense *smallest*. There is a one-one correspondence of the smallest explanations to a certain subset of the minimal explanations (Prop. C27). Smallest explanations can be combined with the background by adding their positive literals as facts, which yields a normal logic program, and removing from the open scope all members whose atom occurs in the explanation scope, independently of the particular explanation (see Sect. C and D).

In the literature, it is often required that the combination of explanation and background is consistent. For reasons explicated in Sect. Bthis is specified here as a separate property:

Definition 7 (Background Consistent Explanation). An explanation H for an abductive setting with semantics *sem*, open scope O and background F is called *background consistent* if and only if $sem_O(F \wedge H)$ is satisfiable.

Integrity constraints, that is, rules with empty head, are in the literature on abduction in logic programming often assigned a special role. We consider here just normal logic programs, which, however, allow to encode constraints with respect to the *consistency view* [19] by rules with a head atom that indicates failure and is added negated to the observation [8, Sect. 3].

5 The Globally Weakest Sufficient Condition

The *globally weakest sufficient condition (GWSC)* [37] is the application pattern of second-order quantification by which explanations with respect to classical logic are characterized as in (i) in the introduction. We specify it formally in terms of literal projection, such that also polarity can be constrained:

Definition 8 (Globally Weakest Sufficient Condition). The *globally weakest sufficient condition (GWSC)* of formula G on scope S within formula F, in symbols $\mathsf{gwsc}_S(F, G)$, is defined as $\mathsf{gwsc}_S(F, G) \stackrel{\text{def}}{=} \neg\mathsf{project}_{\overline{S}}(F \wedge \neg G)$.

The following alternate characterization provides intuition on the relationship to abductive explanations: The GWSC of G on S within F is the weakest formula $H \Subset S$ such that $F \wedge H \models G$. More precisely:

Proposition 9 (Alternate Characterization of the GWSC). *For all formulas F, G, H and scopes S it holds that $H \equiv \mathsf{gwsc}_S(F, G)$ if and only if: (1.) $H \Subset S$, (2.) $H \models G$, and (3.) for all formulas $H' \Subset S$ such that $F \wedge H' \models G$ it holds that $H' \models H$.*

The following property implies that a GWSC on scope S can be expressed as a propositional formula in negation normal form that only involves literals from S:

Proposition 10 (Scope Closedness of the GWSC). *For all formulas F, G and scopes S it holds that $\mathsf{gwsc}_S(F, G) \Subset S$.*

The *GWSC* is closely related to *weakest sufficient conditions (WSCs)*, devised in [24] for propositional logic and adapted to first-order logic in [5]. Aside of the consideration of polarity, GWSCs differ from WSCs in the sense of [24] in that for a given formula and scope only GWSCs are unique up to equivalence [37].

6 Abduction with Logic Programming Semantics

The GWSC basically relates to classical semantics. How can it be applied with non-classical logic programming semantics? Lemma 11 below, about "extension transparency", provides the required link. It states requirements that allow a formula to be moved between the context of the non-classical semantics in the argument of the *sem* operator – where the formula "extends" a logic program – and a classical context outside the *sem* operator. Based on this lemma, we then develop characterizations of abductive explanations in terms of the GWSC for the considered logic programming semantics.

The involved lemma, theorem and propositions will be expressed in a generic way, where the differences relating to the particular semantics are factorized out into three auxiliary concepts that expand differently, depending on the semantics indicated by their first argument. The first of these concepts, CF, represents the circumscribed formulas in the definitions of stable and $\mathsf{pstable}$. It is thus defined for formulas F as $\mathsf{CF}(\mathsf{stable}, F) \stackrel{\text{def}}{=} F$ and $\mathsf{CF}(\mathsf{pstable}, F) \stackrel{\text{def}}{=} (\mathsf{cons} \wedge \mathsf{rename}_{[01\backslash 11]}(F) \wedge$

$\mathsf{rename}_{[01\setminus 11,00\setminus 01]}(F)$). The second concept, IG, is used to project intermediate results onto specific predicate groups and is defined as $\mathsf{IG}(\mathsf{stable}) \stackrel{\text{def}}{=} 0$ and $\mathsf{IG}(\mathsf{pstable}) \stackrel{\text{def}}{=} \mathsf{imin\text{-}scope} = (0 \cap \mathsf{POS}) \cup (1 \cap \mathsf{NEG})$. The third concept, IC, is required for three-valued semantics to express a polarity dependent mapping between the predicate groups in conjunctive clauses of explanations and of intermediate results. For stable the value of IC is the unaltered argument, for $\mathsf{pstable}$ it is obtained by switching the group of all negative literals to 1: IC is defined for conjunctive clauses $C = (\bigwedge_{i=1}^{m} A_i^0 \wedge \bigwedge_{i=1}^{n} \neg B_i^0)$, where $m, n \geq 0$ and $C \Subset 0$, as $\mathsf{IC}(\mathsf{stable}, C) \stackrel{\text{def}}{=} C$ and $\mathsf{IC}(\mathsf{pstable}, C) \stackrel{\text{def}}{=} (\bigwedge_{i=1}^{m} A_i^0 \wedge \bigwedge_{i=1}^{n} \neg B_i^1)$.

Lemma 11 (Extension Transparency). *Let $sem \in \{\mathsf{stable}, \mathsf{pstable}\}$, let F be a formula, let O be an atom scope, and let G be a formula such that $G \Subset (0 \cap (O \cup \mathsf{NEG})) \cup 1$. Then $sem_O(F \wedge G) \equiv sem_O(F) \wedge \mathsf{CF}(sem, G)$.*

We apply this lemma mostly to formulas G satisfying the stronger condition $G \Subset 0 \cap O$, which means that G can be expressed in terms of open atoms from group 0. The weaker precondition in the lemma results in the course of the proof (Sec. A). It will be used in Sect. 8 to justify a way in which stable model computation invoked on the background combined with the negated observation can be applied to compute explanations. Based on Lemma 11, Theorem 12 below can be proven. It shows for the stable model and the partial stable model semantics that factual explanations are – modulo conversion by IC – exactly the conjunctive clauses in the explanation scope that imply the GWSC of the program representation wrapped in the semantics operator and of the observation. For the well-founded semantics, the equivalence to the partial stable model semantics with respect to explanations for "defined" observations is stated, which follows from Prop. 2.

Theorem 12 (Factual Explanation in Terms of GWSC). *Let $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ be an abductive setting. Let $C \Subset 0$ be a conjunctive clause. If $sem \in \{\mathsf{stable}, \mathsf{pstable}\}$, then the following two statements are equivalent:*
1. C is a factual explanation for $\mathfrak{A}$.
2. $C \Subset S$ and $\mathsf{IC}(sem, C) \models \mathsf{gwsc}_{S \cap \mathsf{IG}(sem)}(sem_O(F), G)$.
If $sem = \mathsf{wf}$ and $G \Subset \mathsf{imin\text{-}scope}$, then (1.) is equivalent to:
3. C is a factual explanation for $\langle \mathsf{pstable}, O, S, F, G\rangle$.

Since $\mathsf{gwsc}_S(\mathsf{pstable}(F), G) \equiv \mathsf{gwsc}_S(\mathsf{pstable}(F), \mathsf{cons} \wedge G)$, in abductive settings with $\mathsf{pstable}$ the observation G can be equivalently replaced by any formula G' such that $(\mathsf{cons} \wedge G') \equiv (\mathsf{cons} \wedge G)$. In particular, an observation $(p^0 \wedge p^1)$, which expresses that p is T, can be replaced by just p^0, and $(\neg p^0 \wedge \neg p^1)$, which expresses that p is F, by just $\neg p^1$. The following example illustrates a case where the factual explanations with stable differ from those with $\mathsf{pstable}$ and wf.

Example 13 (Abduction with Different Semantics I). Let $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ be an abductive setting, where $O = S = \{a, b\}$, $F = (p^0 \leftarrow a^0 \wedge b^0) \wedge (p^0 \leftarrow a^0 \wedge \neg b^1)$, and $G = p^0$. If $sem = \mathsf{stable}$, there is a single minimal factual explanation for $\mathfrak{A}$, namely a^0. If $sem \in \{\mathsf{pstable}, \mathsf{wf}\}$, there are two: First,

$(a^0 \wedge b^0)$, second $(a^0 \wedge \neg b^0)$. To see that a^0 is then not an explanation, consider that $\mathsf{pstable}_{\{a,b\}}(F \wedge a^0) \equiv \langle\{a\},\{\}\rangle \vee \langle\{a,b,p\},\{\}\rangle \vee \langle\{a,p\},\{b\}\rangle$.

The following comprehensive example demonstrates further differences of the three considered logic programming semantics with respect to abduction, in particular a case where a meaningful explanation for a fact being observed as *undefined* is only obtained with the partial stable model semantics.

Example 14 (Abduction with Different Semantics II). Assume a domain with two persons a, b, one of them, b, being "the barber". For $x, y \in \{a, b\}$ let sxy stand for "x shaves y", let mx, fx stand for "x is male" and "x is female", respectively. In addition let ss stand for "barbers are self-shavers". The following program F expresses: "a person that is male and does not shave himself is shaved by b", "if barbers are self-shavers, then b shaves himself", and "all persons are either female or male": $F = (sba^0 \leftarrow ma^0 \wedge \neg saa^1) \wedge (sbb^0 \leftarrow mb^0 \wedge \neg sbb^1) \wedge (sbb^0 \leftarrow ss^0) \wedge (fa^0 \leftarrow \neg ma^1) \wedge (ma^0 \leftarrow \neg fa^1) \wedge (fb^0 \leftarrow \neg mb^1) \wedge (mb^0 \leftarrow \neg fb^1)$. Let $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ be an abductive setting, where $O = S = \{ma, mb, ss\}$. Let us first consider the partial stable model semantics, i.e., assume $sem = \mathsf{pstable}$. A distinguishing feature of this semantics is that it allows to compute explanations for the *undefinedness* of observations: Let $G = (\neg sbb^0 \wedge sbb^1)$. Then $G \notin \mathsf{imin\text{-}scope}$ and G expresses that "sbb is U". As the single minimal factual explanation for $\mathfrak{A}$ we then obtain $(mb^0 \wedge \neg ss^0)$. Since the well-founded model is a partial stable model, this is also an explanation w.r.t. the well-founded semantics. However, there are explanations w.r.t. the well-founded semantics that are not explanations w.r.t. the partial stable model semantics. Here for example the "empty" explanation $\top$, since the well-founded model of F w.r.t. O is $\langle\{\},\{saa\}\rangle$, where the value of sbb is U. Notice that in the example only the explanation w.r.t. the partial stable model semantics provides the desired information about the reasons for sbb being undefined, i.e., that the barber is male and that "barbers are self-shavers" is false. For "defined" observations G, i.e., if $G \Subset \mathsf{imin\text{-}scope}$, explanations w.r.t. the partial stable model semantics and the well-founded semantics coincide. In the case $G = sbb^0$, expressing that the value of sbb is T, we obtain ss^0 as single minimal factual explanation. In the case $G = \neg sbb^1$, expressing that the value of sbb is F, we obtain $(\neg mb^0 \wedge \neg ss^0)$. Let us now consider the stable model semantics, i.e., assume $sem = \mathsf{stable}$. For the observation $G = sbb^0$, the minimal factual explanations then are ss^0 and mb^0, the first one coinciding with the partial stable model semantics. For the observation $G = \neg sbb^0$, the only minimal factual explanation is just $\neg ss^0$. The dependency of $\neg mb^0$ in the explanation obtained for the partial stable model semantics, introduced through the "paradoxical" rule $(sbb^0 \leftarrow mb^0 \wedge \neg sbb^1)$, is not taken into account by the stable model semantics.

All the three considered logic programming semantics are factual explanation monotonic, which follows from Theorem 12:

Proposition 15 (Factual Explanation Monotonicity of Considered Logic Programming Semantics). *An abductive setting $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ where $sem \in \{\mathsf{stable}, \mathsf{pstable}, \mathsf{wf}\}$ is factual explanation monotonic.*

Theorem 12 gives a characterization of factual explanations in terms of conjunctive clausal implicants of some particular GWSC. A straightforward consequence is that *minimal* factual explanations correspond to *prime* implicants of that GWSC, as stated in the following proposition. Recall that a *prime implicant* of a formula F is a conjunctive clause C such that $C \models F$ and there does not exist another conjunctive clause D such that $D \models F$ and $D \subset C$.

Proposition 16 (Minimal Factual Explanations and Prime Implicants). *Let $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ be an abductive setting. Let $C \Subset 0$ be a conjunctive clause. Then the following two statements are equivalent for $sem \in \{\mathsf{stable}, \mathsf{pstable}\}$:*
1. C is a minimal factual explanation for $\mathfrak{A}$.
2. $\mathsf{IC}(sem, C)$ is a prime implicant of $\mathsf{gwsc}_{S \cap \mathsf{IG}(sem)}(sem_O(F), G)$.
If $sem = \mathsf{wf}$ and $G \Subset \mathsf{imin\text{-}scope}$, then (1.) is equivalent to:
3. C is a minimal factual explanation for $\langle \mathsf{pstable}, O, S, F, G\rangle$.

From Prop. 10 it follows that $\mathsf{gwsc}_{S \cap \mathsf{IG}(sem)}(sem_O(F), G) \Subset S \cap \mathsf{IG}(sem)$, and thus, if $sem = \mathsf{pstable}$, then $\mathsf{gwsc}_{S \cap \mathsf{IG}(sem)}(sem_O(F), G)$ is equivalent to a formula in DNF with only consistent disjuncts, where the positive literal formulas are from group 0 and the negative ones from group 1. To convert such a DNF into prime implicants form, i.e., the disjunction of all its prime implicants, it suffices to remove subsumed conjunctive clauses. The following example illustrates the relationship of prime implicants and minimal explanations for the partial stable model semantics.

Example 17 (Prime Implicants Form with Partial Stable Models). Consider the setting of Examp. 13. Then $\mathsf{gwsc}_{S \cap \mathsf{IG}(\mathsf{pstable})}(\mathsf{pstable}_O(F), G) \equiv (a^0 \wedge b^0) \vee (a^0 \wedge \neg a^1) \vee (a^0 \wedge \neg b^1) \vee (b^0 \wedge \neg b^1)$, where the latter formula is in prime implicants form. To obtain the minimal factual explanations, we remove the two disjuncts $(a^0 \wedge \neg a^1)$ and $(b^0 \wedge \neg b^1)$, which would become inconsistent after renaming from group 1 to group 0. This requirement of consistency is implicit in Prop. 16 with the precondition that $C \Subset 0$ is a conjunctive clause.

7 Related Work

As indicated in the introduction in the context of the second-order characterization (i) of classical abductive explanations, similar characterizations have been formulated in a number of works. With respect to non-monotonic semantics, the author is only aware of a second-order characterization for default logic in [32], where a translation of default abduction problems into QBFs is specified such that the models of the resulting QBF correspond to the explanations. The relationship to second-order quantifier *elimination* is not made explicit there. In [7] a QBF characterization of the existence of consistent abductive explanations with respect to classical propositional logic is shown. Only positive explanations, that is, sets of atoms, are permitted. To achieve this, *literal* projection is encoded as Boolean quantification in [7]. Otherwise, the presented schema is essentially (i) conjoined with a condition that ensures background consistency. In [7] also a

QBF representation of the stable model semantics is given, but its interplay with abduction is not considered there. In [8] abduction for stable model and well-founded semantics is formalized and complexity results for associated decision problems are given. The role of QBFs there is that hardness results are proven with translations from decision problems for QBFs with certain quantifier prefixes into the abductive decision problems. Negative literals in explanations are not considered in [8].

Several works on computing *credulous* abductive explanations with respect to the stable model semantics are based on the approach of [20]. Similarities to the present work include the consideration of open abducibles and the relationship of minimal explanations to prime implicants. Computation of skeptical explanations can be performed with the credulous approach in a trivial way: Computing all stable models of the background and possible explanations, independently of the observation, and inspecting these afterwards. In [17] it is shown how the computation of credulous explanations with respect to the stable model semantics can be expressed as computation of stable models of programs with integrity rules. The knowledge base is a normal logic program. To encode that abducibles are open, for each abducible p rules $(p \leftarrow \mathsf{not}\ p')$ and $(p' \leftarrow \mathsf{not}\ p)$ are added, where p' is a fresh symbol. Finally, the observation q is added as an integrity constraint $(\bot \leftarrow \mathsf{not}\ q)$. There is a one-one correspondence between stable models of the resulting program and explanations of q. As noted in [25], a major drawback of this method is that it involves the actual computation of *all* explanations, not taking into account that the minimal ones provide a succinct representation of them. A variant of [17] is described in [16], where a generalization of the stable model semantics to rules with literals instead of just atoms, as well as disjunctive heads and negation as failure in the head is considered. Computation of explanations is there encoded similarly to [17], except that the openness of abducibles p is expressed by rules $(p \mid \mathsf{not}\ p \leftarrow \top)$. Minimality with respect to the set of abducibles is taken into account [16, Corollary 3.3], but in a way that just suggests to compute first the models and only afterwards extract explanations and compare them with respect to minimality. In [25], the approach of [20] is improved by discerning redundant explanations. Explanations correspond to sets of *literals*. It is shown that the set of all explanations can be represented by the set of minimal explanations, and that minimal explanations to correspond to prime implicants. Again, only credulous explanations are considered.

A characterization of stable models in terms of circumscription is presented in [10] as a transformation $\mathsf{SM}(F)$ on classical formulas F. In contrast to the stable operator, based on [23], the predicate occurrences that are affected by circumscription are identified in [10] by their syntactic position within the formula, such that classically equivalent formulas are not necessarily equivalent with respect to the logic programming semantics. Interestingly, an analog to Lemma 11 is shown in [10, Sect. 5.1]: $\mathsf{SM}(F \wedge G) \equiv \mathsf{SM}(F) \wedge G$ *whenever* G *has no strictly positive occurrences* [read: each occurrence is negative, i.e. is in NEG, or is subjected to negation as failure, i.e., is from group 1] *of intensional predicates* [read: predicates that are not open, i.e., are not in O]. Observe that if 0 and 1 are the

only predicate groups, then $\mathsf{NEG} \cup 1 \cup O = (0 \cap (O \cup \mathsf{NEG})) \cup 1$, matching exactly the precondition upon G of Lemma 11.

Abduction with respect the well-founded semantics has been elaborated in [1] for programs with a second type of negation, so-called explicit negation, and integrity constraints. A semantic characterization of explanations is specified, and a computation method is described and proven correct. Explicit negation and "coherency" in [1] at least superficially correspond to predicate group 1 and the cons axiom, although a detailed comparison still needs to be done. Concerning abduction with respect to the partial stable model semantics, the present author is not aware of a thorough previous investigation.

8 Conclusion

We have seen that abductive explanations with respect to different logic programming semantics can be characterized semantically as formulas with second-order operators. This provides a solid basis for subtle issues such as abduction with the partial stable model semantics, and, as further described in the appendices, alternate kinds of minimality, the handling of negative facts in explanations, and abductive consequences. A distinguishing feature of such characterizations is that they can be directly processed by elimination of the second-order operators, that is, computing for a given formula with these operators an equivalent formula without them. Approaches to second-order quantifier elimination include, with respect to first-order and modal logics, the resolution-based SCAN [12,14] and the direct methods [4,11]. Of course, with respect to full first-order logic, these methods are inherently incomplete. Further relevant techniques stem from knowledge compilation [34] and SAT solving, where Boolean variable elimination is an important preprocessing technique [6,27].

From an algorithmic point of view, the elimination approach suggests two possible ways to divide the computation of explanations into subtasks. Consider the computation of all minimal background consistent factual explanations with respect to the stable model semantics. According to Prop. 16, the core expression then is $\mathsf{gwsc}_{S \cap 0}(\mathsf{stable}_O(F), G)$. Explanations can be computed by expanding the gwsc and stable operators, eliminating the resulting second-order quantifiers, and postprocessing the result by computing prime implicants and removing explanations that are not background consistent. A naive implementation that proceeds in this way and allows small experiments is provided with [36][3]. The second way to divide the computation begins with computing $\mathsf{stable}_O(F)$ with a dedicated system for stable models. Lemma 11 justifies to take positive observations into account at this stage: If G contains only positive atoms from group 0, then $\mathsf{gwsc}_{S \cap 0}(\mathsf{stable}_O(F), G) \equiv \neg \mathsf{project}_{\overline{S} \cap 0}(\mathsf{stable}_O(F) \wedge \neg G) \equiv \neg \mathsf{project}_{\overline{S} \cap 0}(\mathsf{stable}_O(F \wedge \neg G))$. Combinations of stable model computation with second-order quantifier elimination have been developed [9,13], but it needs to be investigated whether they can be used for the computations suggested here.

[3] Available at `http://cs.christophwernhard.com/toyelim/`

On the agenda for future work are also further applications of the semantic aspects of the characterizations. For example, relationships to concepts of equivalence of logic programs, in particular abductive equivalence [31] and uniform equivalence. Can complexity results be read-off from the characterizations? Are there useful relationships between abduction *with respect to non-monotonic semantics* and the many other applications of GWSC and WSCs [24,5,37] as well as the further similar concept of *perfect rewriting* [2]?

Acknowledgments. The author wishes to thank anonymous reviewers for bringing related work to attention.

References

1. Alferes, J.J., Pereira, L.M., Swift, T.: Abduction in well-founded semantics and generalized stable models via tabled dual programs. Theory and Pract. Log. Program. 4(4), 383–428 (2004)
2. Calvanese, D., Giacomo, G.D., Lenzerini, M., Vardi, M.Y.: View-based query processing: On the relationship between rewriting, answering and losslessness. Theor. Comp. Sci. 371(3), 169–182 (2007)
3. Denecker, M., Kakas, A.C.: Abduction in logic programming. In: Kakas, A.C., Sadri, F. (eds.) Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part I. LNCS, vol. 2407, pp. 402–436 (2002)
4. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing circumscription revisited: A reduction algorithm. J. Autom. Reasoning 18(3), 297–338 (1997)
5. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing strongest necessary and weakest sufficient conditions of first-order formulas. In: IJCAI-01. pp. 145–151. Morgan Kaufmann (2001)
6. Eén, N., Biere, A.: Effective preprocessing in SAT through variable and clause elimination. In: SAT'05. LNCS, vol. 3569, pp. 61–75 (2005)
7. Egly, U., Eiter, T., Tompits, H., Woltran, S.: Solving advanced reasoning tasks using quantified Boolean formulas. In: AAAI-00. pp. 417–422. AAAI Press (2000)
8. Eiter, T., Gottlob, G., Leone, N.: Abduction from logic programs: Semantics and complexity. Theor. Comp. Sci. 189(1–2), 129–177 (1997)
9. Eiter, T., Wang, K.: Semantic forgetting in answer set programming. Artif. Intell. 172(14), 1644–1672 (2008)
10. Ferraris, P., Lee, J., Lifschitz, V.: Stable models and circumscription. Artif. Intell. 175(1), 236–263 (2011)
11. Gabbay, D.M., Schmidt, R.A., Szałas, A.: Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications. College Publications(2008)
12. Gabbay, D., Ohlbach, H.J.: Quantifier elimination in second-order predicate logic. In: KR'92. pp. 425–435. Morgan Kaufmann (1992)
13. Gebser, M., Kaufmann, B., Schaub, T.: Solution enumeration for projected Boolean search problems. In: CPAIOR 2009. LNCS, vol. 5547, pp. 71–86 (2009)
14. Goranko, V., Hustadt, U., Schmidt, R.A., Vakarelov, D.: SCAN is complete for all Sahlqvist formulae. In: RelMiCS 7. LNCS, vol. 3051, pp. 149–162 (2004)
15. Inoue, K.: Linear resolution for consequence finding. Artif. Intell. 56(2–3), 301–353 (1992)

16. Inoue, K., Sakama, C.: Negation as failure in the head. J. Log. Program. 35(1), 39–78 (1998)
17. Iwayama, N., Satoh, K.: Computing abduction by using TMS with top-down expectation. J. Log. Program. 44, 179–206 (2000)
18. Janhunen, T., Niemelä, I., Seipel, D., Simons, P., You, J.H.: Unfolding partiality and disjunctions in stable model semantics. ACM Trans. Comput. Log. 7(1), 1–37 (2006)
19. Kakas, A.C., Kowalski, R.A., Toni, F.: The role of abduction in logic programming. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.) Handbook of Logic in Artifical Intelligence, vol. 5, pp. 235–324. Oxford University Press (1998)
20. Kakas, A.C., Mancarella, P.: Generalized stable models: A semantics for abduction. In: ECAI-90. pp. 385–391. Pitman (1990)
21. Lang, J., Liberatore, P., Marquis, P.: Propositional independence – formula-variable independence and forgetting. J. of Artif. Intell. Res. 18, 391–443 (2003)
22. Lifschitz, V.: Twelve definitions of a stable model. In: ICLP 2008. LNCS, vol. 5366, pp. 37–51 (2008)
23. Lin, F.: A Study of Nonmonotonic Reasoning. Ph.D. thesis, Stanford Univ. (1991)
24. Lin, F.: On strongest necessary and weakest sufficient conditions. Artif. Intell. 128(1–2), 143–159 (2001)
25. Lin, F., You, J.H.: Abduction in logic programming: A new definition and an abductive procedure based on rewriting. Artif. Intell. 140(1/2), 175–205 (2002)
26. Lobo, J., Uzcátegui, C.: Abductive consequence relations. Artif. Intell. 89(1–2), 149–171 (1997)
27. Manthey, N.: Coprocessor 2.0 – a flexible CNF simplifier. In: SAT '12. LNCS, vol. 7317, pp. 436–441 (2012)
28. Przymusinski, T.: Well-founded semantics coincides with three-valued stable semantics. Fundam. Inform. 13(4), 445–464 (1990)
29. Przymusinski, T.: Stable semantics for disjunctive programs. New Gen. Comput. 9(3/4), 401–424 (1991)
30. Saccà, D., Zaniolo, C.: Deterministic and non-deterministic stable models. J. Log. Comput. 7(5), 555–579 (1997)
31. Sakama, C., Inoue, K.: Equivalence issues in abduction and induction. J. Applied Logic 7(3), 318–328 (2009)
32. Tompits, H.: Expressing default abduction problems as quantified Boolean formulas. AI Commun. 16(2), 89–105 (2003)
33. Wernhard, C.: Literal projection for first-order logic. In: JELIA 08. LNCS (LNAI), vol. 5293, pp. 389–402 (2008)
34. Wernhard, C.: Tableaux for projection computation and knowledge compilation. In: TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 325–340 (2009)
35. Wernhard, C.: Circumscription and projection as primitives of logic programming. In: Tech. Comm. ICLP 2010. LIPIcs, vol. 7, pp. 202–211 (2010)
36. Wernhard, C.: Computing with logic as operator elimination: The ToyElim system. In: WLP 2011. pp. 94–98. Infsys Res. Rep. 1843-11-06, TU Wien (2011)
37. Wernhard, C.: Projection and scope-determined circumscription. J. Symb. Comput. 47(9), 1089–1108 (2012)
38. Wernhard, C.: Abduction in logic programming as second-order quantifier elimination. In: FroCoS 2013. LNCS (LNAI), vol. 8152, pp. 103–119. Springer (2013)

The following appendix sections contain proofs of Lemma 11 and Theorem 12 (Sect. A), and discussions of background consistency (Sect. B), smallest factual explanations (Sect. C), the possibilities to incorporate negative facts in explanations into programs (Sect. D) and abductive consequences (Sect. E).

A Proofs

This appendix section contains proofs of Lemma 11 and Theorem 12. In the proof of Lemma 11 properties of circumscription are applied, for which we need some auxiliary definitions and propositions. We define scope-determined circumscription equivalently to the definition in Sect. 2 but in terms of an auxiliary intermediate operator raise [37].

Definition A18 (Raising and Scope-Determined Circumscription). For interpretations I, scopes S and formulas F, define:

(i) $I \models \mathsf{raise}_S(F)$ iff_{def} there exists an interpretation J such that $J \models F$ and $J \cap S \subset I \cap S$.

(ii) $I \models \mathsf{circ}_S(F)$ iff_{def} $I \models F \wedge \neg\mathsf{raise}_S(F)$.

Proposition A20.ii below provides an alternate characterization of raising that refers to the *biscope* and *uniscope* of a scope, two disjoint subsets into which a scope can be partitioned: The biscope contains those members of the scope whose complement is also a member of the scope (thus they are "*bi*-polar" members). The uniscope contains the remaining members of the scope, that is, those whose complement is not also a member of the scope (thus they are "*uni*-polar" members). The following definitions provide formal notation for this:

Definition A19 (Biscope and Uniscope Partitions of a Scope). For scopes S define:

(i) $\mathsf{bsc}(S) \stackrel{\text{def}}{=} S \cap \overline{S}$.

(ii) $\mathsf{usc}(S) \stackrel{\text{def}}{=} S - \overline{S}$.

Proposition A20 (Properties of Raising). *Let S be a scope, let F, G be formulas and let I be an interpretation. It then holds that:*

(i) *If $F \models G$, then $\mathsf{raise}_S(F) \models \mathsf{raise}_S(G)$.*

(ii) *$I \models \mathsf{raise}_S(F)$ if and only if there exists an interpretation J such that $I \models F$, $J \cap \mathsf{bsc}(S) = I \cap \mathsf{bsc}(S)$, and $J \cap \mathsf{usc}(S) \subset I \cap \mathsf{usc}(S)$.*

Lemma 11 (Extension Transparency) Let $sem \in \{\mathsf{stable}, \mathsf{pstable}\}$, let F be a formula, let O be an atom scope, and let G be a formula such that $G \Subset (0 \cap (O \cup \mathsf{NEG})) \cup 1$. Then $sem_O(F \wedge G) \equiv sem_O(F) \wedge \mathsf{CF}(sem, G)$.

Proof. We show the lemma separately for the two cases that *sem* is stable and $\mathsf{pstable}$. The essential difference of the two proofs is that formulas "extending" the logic program can be moved directly outside the stable operator, while they have to be subjected to a systematic renaming of negative literals to predicate

group 1 for the partial stable model semantics. Thus, the proofs differ essentially just in that in the case of pstable these renaming transformations are woven in.

Case *sem* = stable**.** Define SC as shorthand for the circumscription scope in the definition of stable, that is, $+0 \cup 1 \cup (O \cap 0)$. We first show that the precondition $G \Subset (0 \cap (O \cup \mathsf{NEG})) \cup 1$ implies the following equivalence:

$$\mathsf{raise}_{SC}(F \wedge G) \wedge G \equiv \mathsf{raise}_{SC}(F) \wedge G. \tag{xii}$$

The left-to-right direction of equivalence (xii) follows from Prop. A20.i. The right-to-left direction is shown in the following table. Assume (1), the precondition of the proposition. Let I be a model of the right side of equivalence (xii), that is, assume (2) and (3). We derive with step (10) that I is also a model of the left side.

(1)	$G \Subset (0 \cap (O \cup \mathsf{NEG})) \cup 1$.	assumption
(2)	$I \models \mathsf{raise}_{SC}(F)$.	assumption
(3)	$I \models G$.	assumption
(4)	There exists a J such that	
(5)	$\quad J \models F$,	
(6)	$\quad J \cap 1 = I \cap 1$,	
(7)	$\quad J \cap O \cap 0 = I \cap O \cap 0$, and	
(8)	$\quad J \cap 0 \cap \mathsf{POS} \subset I \cap 0 \cap \mathsf{POS}$.	by (2), Prop. A20.ii
(9)	$J \models G$.	by (6)–(8), (3), (1)
(10)	$I \models \mathsf{raise}_{SC}(F \wedge G)$.	by (6)–(9), Prop. A20.ii

Now, the proposition can be shown with the following equivalences, obtained by expanding or contracting operators and applying the equivalence (xii):

(11)	$\mathsf{stable}_O(F \wedge G)$	
(12)	$\equiv \mathsf{rename}_{[0\setminus 1]}(\mathsf{circ}_{SC}(F \wedge G))$	
(13)	$\equiv \mathsf{rename}_{[0\setminus 1]}(F \wedge \neg\mathsf{raise}_{SC}(F \wedge G) \wedge G)$	by equiv. (xii)
(14)	$\equiv \mathsf{rename}_{[0\setminus 1]}(F \wedge \neg\mathsf{raise}_{SC}(F) \wedge G)$	
(15)	$\equiv \mathsf{rename}_{[0\setminus 1]}(\mathsf{circ}_{SC}(F)) \wedge G$	
(16)	$\equiv \mathsf{stable}_O(F) \wedge G$	
(17)	$\equiv \mathsf{stable}_O(F) \wedge \mathsf{CF}(\mathsf{stable}, G)$.	

Case *sem* = pstable**.** Define SCP as shorthand for the circumscription scope in the definition of pstable, that is, $((0 \cup 1) \cap \mathsf{POS}) \cup 2 \cup 3 \cup O$. We first show that the precondition $G \Subset (0 \cap (O \cup \mathsf{NEG})) \cup 1$ implies the following equivalence:

$$\begin{aligned} &\mathsf{raise}_{SCP}(\mathsf{CF}(\mathsf{pstable}, F \wedge G)) \wedge \mathsf{CF}(\mathsf{pstable}, G) \\ \equiv\ &\mathsf{raise}_{SCP}(\mathsf{CF}(\mathsf{pstable}, F)) \wedge \mathsf{CF}(\mathsf{pstable}, G). \end{aligned} \tag{xiii}$$

Since clearly $\mathsf{CF}(\mathsf{pstable}, F \wedge G) \models \mathsf{CF}(\mathsf{pstable}, F)$, the left-to-right direction of equivalence (xiii) follows from Prop. A20.i. The right-to-left direction is shown in the following table. Assume (1), the precondition of the proposition. Let G' as specified in (2). Steps (3) and (4) then follow from (2) and (1). Let I be a model of the right side of equivalence (xiii), that is, assume (4) and (5). We derive with step (18) that I is also a model of the left side.

(1)	$G \Subset 0 \cap (O \cup \mathsf{NEG})) \cup 1.$	assumption
(2)	$G' \stackrel{\text{def}}{=} \mathsf{rename}_{[1\setminus 3]}(G) \wedge \mathsf{rename}_{[1\setminus 2,0\setminus 1]}(G).$	definition
(3)	$\mathsf{CF}(\mathsf{pstable}, G) \equiv \mathsf{cons} \wedge G'.$	by (2), (1)
(4)	$G' \Subset (((0 \cup 1) \cap (O \cup \mathsf{NEG})) \cup 2 \cup 3.$	by (2), (1)
(5)	$I \models \mathsf{raise}_{SCP}(\mathsf{CF}(\mathsf{pstable}, F)).$	assumption
(6)	$I \models \mathsf{CF}(\mathsf{pstable}, G).$	assumption
(7)	$I \models G'.$	by (6), (3)
(8)	There is a J such that	
(9)	$\quad J \models \mathsf{CF}(\mathsf{pstable}, F),$	
(10)	$\quad J \cap 2 = I \cap 2,$	
(11)	$\quad J \cap 3 = I \cap 3,$	
(12)	$\quad J \cap O \cap (0 \cup 1) = I \cap O \cap (0 \cup 1),$	
(13)	$\quad J \cap (0 \cup 1) \cap \mathsf{POS} \subset I \cap (0 \cup 1) \cap \mathsf{POS}.$	by (5), Prop. A20.ii
(14)	$J \models G'.$	by (10)–(13), (7), (4)
(15)	$J \models \mathsf{cons}.$	by (9)
(16)	$J \models \mathsf{CF}(\mathsf{pstable}, G).$	by (15), (14), (3)
(17)	$J \models \mathsf{CF}(\mathsf{pstable}, F \wedge G).$	by (16), (9)
(18)	$I \models \mathsf{raise}_{SCP}(\mathsf{CF}(\mathsf{pstable}, F \wedge G)).$	by (17), (10)–(14), Prop. A20.ii

To derive step (14) we applied that for all formulas F, scopes S and interpretations I, J it holds that if $F \Subset S$, $I \models F$ and $I \cap S \subseteq J$, then $J \models F$. Now, the proposition can be shown with the following equivalences, obtained by expanding or contracting operators and, to obtain the equivalence of (23) to (22), by equivalence (xiii).

(19)		$\mathsf{pstable}_O(F \wedge G)$
(20)	$\equiv$	$\mathsf{rename}_{[2\setminus 0,3\setminus 1]}(\mathsf{circ}_{SCP}(\mathsf{CF}(\mathsf{pstable}, F \wedge G)))$
(21)	$\equiv$	$\mathsf{rename}_{[2\setminus 0,3\setminus 1]}(\mathsf{CF}(\mathsf{pstable}, F \wedge G) \wedge \neg\mathsf{raise}_{SCP}(\mathsf{CF}(\mathsf{pstable}, F \wedge G)))$
(22)	$\equiv$	$\mathsf{rename}_{[2\setminus 0,3\setminus 1]}(\mathsf{CF}(\mathsf{pstable}, F) \wedge \neg\mathsf{raise}_{SCP}(\mathsf{CF}(\mathsf{pstable}, F \wedge G)) \wedge \mathsf{CF}(\mathsf{pstable}, G))$
(23)	$\equiv$	$\mathsf{rename}_{[2\setminus 0,3\setminus 1]}(\mathsf{CF}(\mathsf{pstable}, F) \wedge \neg\mathsf{raise}_{SCP}(\mathsf{CF}(\mathsf{pstable}, F)) \wedge \mathsf{CF}(\mathsf{pstable}, G))$
(24)	$\equiv$	$\mathsf{rename}_{[2\setminus 0,3\setminus 1]}(\mathsf{circ}_{SCP}(\mathsf{CF}(\mathsf{pstable}, F))) \wedge \mathsf{CF}(\mathsf{pstable}, G)$
(25)	$\equiv$	$\mathsf{pstable}_O(F) \wedge \mathsf{CF}(\mathsf{pstable}, G).$

In the proofs of the cases of Theorems 12 and E29 that apply to the partial stable model semantics we will represent the systematic polarity dependent renaming of predicate groups in conjunctive clauses with a shorthand as follows: Let

$$C = (\bigwedge_{i=1}^{m} A_i^0 \wedge \bigwedge_{i=1}^{n} \neg B_i^0),$$

be a conjunctive clause, where $m, n \geq 0$. We then write C also as $C[+0, -0]$, and write C with the negated atoms replaced by their correspondents from group 1, that is,

$$(\bigwedge_{i=1}^{m} A_i^0 \wedge \bigwedge_{i=1}^{n} \neg B_i^1),$$

as $C[+0, -1]$. The other combinations, $C[+1, -0]$ and $C[+1, -1]$, are understood analogously. The following proposition shows conversions from CF and conjunctive clauses $C[+0, -0]$, $C[+0, -1]$, $C[+1, -1]$ in the presence of cons.

Proposition A21 (Properties of Conjunctive Clauses over Group Settings). *For all conjunctive clauses* $C[+0,-0]$ *it holds that:* $\mathsf{cons} \wedge C[+0,-1] \equiv \mathsf{cons} \wedge C[+0,-0] \wedge C[+1,-1] \equiv \mathsf{CF}(\mathsf{pstable}, C[+0,-0])$.

Theorem 12 (Factual Explanation in Terms of GWSC) *Let* $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ *be an abductive setting. Let* $C \Subset 0$ *be a conjunctive clause. If* $sem \in \{\mathsf{stable}, \mathsf{pstable}\}$*, then the following two statements are equivalent:*
1. C *is a factual explanation for* $\mathfrak{A}$*.*
2. $C \Subset S$ *and* $\mathsf{IC}(sem, C) \models \mathsf{gwsc}_{S \cap \mathsf{IG}(sem)}(sem_O(F), G)$*.*
If $sem = \mathsf{wf}$ *and* $G \Subset \mathsf{imin\text{-}scope}$*, then (1.) is equivalent to:*
3. C *is a factual explanation for* $\langle \mathsf{pstable}, O, S, F, G\rangle$*.*

Proof. We show the proposition separately for the three cases that *sem* is stable, pstable and wf.

Case $sem = \mathsf{stable}$**.** For this case we show the theorem generalized to arbitrary explanations instead of just *factual* explanations. We prove that under the precondition that $S \subseteq O$, which holds by the definition of abductive setting (Def. 3), the following two statements are equivalent for all formulas H:

$$H \text{ is an explanation for } \mathfrak{A}, \text{ and} \tag{xiv}$$
$$H \Subset S \cap 0 \text{ and } H \models \mathsf{gwsc}_{S \cap 0}(\mathsf{stable}_O(F), G). \tag{xv}$$

This equivalence implies the theorem since for all conjunctive clauses C it holds that $\mathsf{IC}(\mathsf{stable}, C) = C$ and $\mathsf{IG}(\mathsf{stable}) = 0$. Consider the following table. We assume as step (1) the precondition $S \subseteq O$, ensured by Def. 3. In addition, we assume as step (2) that $H \Subset S \cap 0$. For the right side of the theorem this condition is explicitly stated, for the left side it follow from the definition of explanation (Def. 4). Step (3) follows from these assumptions.

(1)	$S \subseteq O$.	assumption
(2)	$H \Subset S \cap 0$.	assumption
(3)	$H \Subset O \cap 0$.	by (1), (2)

Equivalence of (xiv) and (xv) now follows since under the assumptions just made H is an explanation for $\mathfrak{A}$ if and only if $H \models \mathsf{gwsc}_{S \cap 0}(\mathsf{stable}_O(F), G)$:

(4)	H is an explanation for $\mathfrak{A}$	
(5)	iff $\mathsf{stable}_O(F \wedge H) \models G$	by (2), Def. 4
(6)	iff $\mathsf{stable}_O(F) \wedge H \models G$	by (3), Lem. 11
(7)	iff $H \models \mathsf{gwsc}_{S \cap 0}(\mathsf{stable}_O(F), G)$.	by (2), Prop. 9

Case $sem = \mathsf{pstable}$**.** Let $C = C[+0,-0]$ be a conjunctive clause. Recall that by definition $\mathsf{IC}(\mathsf{pstable}, C[+0,-0]) = C[+0,-1]$. Thus statement (1.) in the theorem expands into

$$C[+0,-0] \text{ is a factual explanation for } \mathfrak{A},$$

and statement (2.) expands into

$$C[+0,-1] \models \mathsf{gwsc}_{S \cap \mathsf{IG}(\mathsf{pstable})}(\mathsf{pstable}_O(F), G).$$

Consider the following table. We assume as step (1) the precondition $S \subseteq O$, ensured by Def. 3. In addition, we assume as steps (2) and (3) that $C[+0, -0]$ is a conjunctive clause and that $C[+0, -0] \Subset S \cap 0$. For the right side of the theorem these conditions are explicitly stated, for the left side they follow from the definition of factual explanation (Def. 4). Steps (4) and (5) follow from these assumptions.

(1)	$S \subseteq O$.	assumption
(2)	$C[+0, -0]$ is a conjunctive clause.	assumption
(3)	$C[+0, -0] \Subset S \cap 0$.	assumption
(4)	$C[+0, -0] \Subset O \cap 0$.	by (1), (3)
(5)	$C[+0, -1] \Subset S \cap \mathsf{IG}(\mathsf{pstable})$.	by (3), since $\mathsf{IG}(\mathsf{pstable}) = \mathsf{imin\text{-}scope}$ $= (0 \cap \mathsf{POS}) \cup (1 \cap \mathsf{NEG})$

We conclude the proof for the case of $sem = \mathsf{pstable}$ by showing that under the assumptions just made the following equivalences hold:

(6)		$C[+0, -0]$ is a factual explanation for $\mathfrak{A}$	
(7)	iff	$\mathsf{pstable}(F \wedge C[+0, -0]) \models G$	by (2), (3), Def. 4
(8)	iff	$\mathsf{pstable}(F) \wedge C[+0, -1] \models G$	by (4), Lem. 11, Prop. A21
(9)	iff	$C[+0, -1] \models \mathsf{gwsc}_{S \cap \mathsf{IG}(\mathsf{pstable})}(\mathsf{pstable}_O(F), G)$.	by (5), Prop. 9

To derive step (8) we applied that for all formulas F and ungrouped atom scopes O it holds that $\mathsf{pstable}_O(F) \models \mathsf{cons}$.

Case $sem = \mathsf{wf}$**.** This case follows easily from the case $sem = \mathsf{pstable}$, the definition of factual explanation and Prop. 2.

B Background Consistency as a Separate Property

Background consistency is in the literature on abduction often required as an inherent property of explanations. Our specification as a separate property (Def. 7) is motivated by the follwing rationales: In a classical setting, the GWSC is the unique *weakest* explanation. From that point of view, requiring background consistency from explanations is redundant, since explanations fail to have this property only if no other explanations exist. A second rationale is that in explanation monotonic settings background consistency can be straightforwardly ensured with a "postprocessing" operation applied to the GWSC, as justified by the following proposition:

Proposition B22 (Picking Background Consistent Explanations). *Let $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ be an abductive setting. If $\langle sem, O, S, F, \bot\rangle$ is factual explanation monotonic, then the following statements are equivalent:*
1. C is a background consistent factual explanation for $\mathfrak{A}$ and there does not exist another background consistent factual explanation D for $\mathfrak{A}$ such that $D \subset C$.
(2.) C is a background consistent minimal factual explanation for $\mathfrak{A}$.

Proposition B22 can be applied to ensure that the factual explanations which are background consistent and *minimal compared to the other background consistent factual explanations* – i.e., those explanations which are typically desired

as final output of the explanation computation – can be obtained by picking the background consistent explanations from the minimal (but not necessarily background consistent) factual explanations obtained, e.g., as prime implicants from the GWSC according to Prop. 16.

C Smallest Factual Explanations

As already indicated in Sect. 4, aside of the concept of *minimal* factual explanation (Def. 6), that is, "minimality" with respect to a set of *literals*, another concept of "minimality" can be associated with factual explanations: "minimality" with respect to the set of *positive* literals. We call factual explanations that are "minimal" in the latter sense *smallest factual explanations*. The precise definition is based on the concept of *complete* conjunctive clause:

Definition C23 (Complete Conjunctive Clause). A conjunctive clause C is called *complete* for an abductive setting with explanation scope S if and only if $C \Subset S \cap 0$ and there is no other conjunctive clause $D \Subset S \cap 0$ such that $C \subset D$.

Definition C24 (Smallest Factual Explanation). A *smallest factual explanation* for an abductive setting $\mathfrak{A}$ is a complete factual explanation C for $\mathfrak{A}$ such that there does not exists a complete factual explanation D for $\mathfrak{A}$ with $\mathsf{project}_{\mathsf{POS}}(D) \subset \mathsf{project}_{\mathsf{POS}}(C)$.

Example C25 (Smallest and Minimal Factual Explanations). Let $\mathfrak{A} = \langle \mathsf{stable}, O, S, F, p^0 \rangle$, where $O \stackrel{\text{def}}{=} S \stackrel{\text{def}}{=} \{a, b, c, d\}$ and

$$\begin{aligned} F \stackrel{\text{def}}{=}\ & p^0 \leftarrow a^0 \wedge \neg b^1 \wedge \\ & p^0 \leftarrow a^0 \wedge \neg c^1 \wedge \\ & p^0 \leftarrow a^0 \wedge d^0. \end{aligned}$$

Then there are three minimal factual explanations for $\mathfrak{A}$: $(a^0 \wedge \neg b^0)$, $(a^0 \wedge \neg c^0)$, and $(a^0 \wedge d^0)$. There is only a single smallest factual explanation for $\mathfrak{A}$: $(a^0 \wedge \neg b^0 \wedge \neg c^0 \wedge \neg d^0)$.

Clearly, the set of all factual explanations for some abductive setting can be determined from the set of the minimal factual explanations, and vice versa. The smallest factual explanations can be determined from all factual explanations (and thus also from the minimal factual explanations). However, it is not in general possible to determine all factual explanations from the smallest factual explanations. There is a one-one correspondence of the smallest factual explanations to just to a subset of the minimal factual explanations. Proposition C27 below makes this precise. It is preceded by an auxiliary definition, the notation $\mathsf{fillneg}(\mathfrak{A}, C)$ for the conjunctive clause that is complete for $\mathfrak{A}$ and obtained by extending C with negative literals:

Definition C26 (Fillneg). Let $\mathfrak{A}$ be an abductive setting with explanation scope S. For conjunctive clauses $C \Subset S \cap 0$,

$$\mathsf{fillneg}(\mathfrak{A}, C) \stackrel{\text{def}}{=} C \wedge \bigwedge_{\{A^0 \mid -A^0 \in S \cap 0 \text{ and } A^0 \text{ is not a positive literal in } C\}} \neg A^0.$$

Proposition C27 (Smallest and Minimal Factual Explanations). *Let $\mathfrak{A}$ be a factual explanation monotonic abductive setting. Then the following statements are equivalent:*

1. *C is a smallest factual explanation for $\mathfrak{A}$.*
2. *There is a minimal factual explanation D for $\mathfrak{A}$ such that $C = \mathsf{fillneg}(\mathfrak{A}, D)$ and there does not exist another minimal factual explanation E for $\mathfrak{A}$ with $\mathsf{project}_{\mathsf{POS}}(E) \subset \mathsf{project}_{\mathsf{POS}}(D)$.*

D Representing Negative Facts by Closing Atoms

The definition of *explanation* (Def. 4) involves the conjunction $(F \wedge H)$ of the explanation H with the classical representation of the logic program F. If H is a factual explanation $A_1^0 \wedge \ldots \wedge A_m^0 \wedge \neg B_1^0 \wedge \ldots \wedge \neg B_n^0$, then positive as well as negative literals in group 0 are conjoined. The positive literals can be viewed as positive facts, rules with empty body, matching the syntactic restrictions for a normal logic program. If the negative literals are conjoined in the same way, they represent unary constraints, that is, rules with empty head and a single positive body literal, breaking the restrictions for a normal logic program.

The following property allows to express the negative literals in explanations in another way, by removing them from the open scope, such that the syntactic constraints of normal logic programs can be preserved:

$$sem_O(F \wedge \neg A^0) \equiv sem_{O-\{A\}}(F). \tag{xvi}$$

This property holds for the stable and partial stable model semantics if A does not occur in the head of a rule of the program represented by F, which can be expressed semantically as $F \equiv \mathsf{forget}_{\{+A^0\}}(F)$.

A disadvantage of applying equivalence (xvi) to represent negative facts by altering the open scope is that the respective open scope then depends on the particular explanation. The following property, which again holds for the stable as well as the partial stable model semantics, justifies that also positive facts can be removed from the open scope:

$$sem_O(F \wedge A^0) = sem_{O-\{A\}}(F \wedge A^0). \tag{xvii}$$

By equivalences (xvi) and (xvii), for all *complete* factual explanations, their conjunction with the background can be represented as a normal logic program with respect to the *same* open scope, the open scope that is obtained from the original open scope by subtracting the explanation scope. If the original open scope and the explanation scope are identical, the open scope obtained by subtracting is empty, corresponding to the "standard" variant of the respective logic programming semantics, with no open predicates.

E Abductive Consequences

Consider the following example from [26]: Let $\mathfrak{A} = \langle \mathsf{stable}, O, S, F, G\rangle$, where $O = S = \{\mathit{rainedLastNight}\}$,

$$F = \begin{array}{ll} \mathit{grass\_is\_wet}^0 \leftarrow \mathit{rained\_last\_night}^0 & \wedge \\ \mathit{do\_not\_bike\_to\_work}^0 \leftarrow \mathit{rained\_last\_night}^0, & \end{array}$$

and $G = \mathit{grassIsWet}^0$. Then $\mathit{rainedLastNight}^0$ is the only minimal factual explanation for $\mathfrak{A}$. When added to the background, this explanation has aside of the observation also $\mathit{doNotBikeToWork}^0$ as a consequence. The notion of *abductive consequence* [26] takes account of such "collateral" consequences of the background when it is combined with explanations of some observation. We consider here a specific variant of this concept: A formula is an *factual-skeptical abductive consequence* of some observation if and only if for all factual explanations of the observation the formula is a consequence of the background combined with the explanation. More precisely:

Definition E28 (Factual-Skeptical Abductive Consequence). Let $\mathfrak{A} = \langle \mathit{sem}, O, S, F, G\rangle$ be an abductive setting. A formula H is a *factual-skeptical abductive consequence* of $\mathfrak{A}$ if and only if for all factual explanations C for $\mathfrak{A}$ it holds that $\mathit{sem}_O(F \wedge C) \models H$.

The following theorem shows how factual-abductive consequences can be characterized with the GWSC. It follows from Theorem 12, Lemma 11 and for the well-founded semantics from Prop. 2.

Theorem E29 (Abductive Consequences). *Let $\mathfrak{A} = \langle \mathit{sem}, O, S, F, G\rangle$ be an abductive setting. If $\mathit{sem} \in \{\mathsf{stable}, \mathsf{pstable}\}$, then the following two statements are equivalent:*
1. H is a factual-skeptical abductive consequence of $\mathfrak{A}$.
2. $\mathit{sem}_O(F) \wedge \mathsf{gwsc}_{S \cap \mathsf{IG}(\mathit{sem})}(\mathit{sem}_O(F), G) \models H$.
If $\mathit{sem} = \mathsf{wf}$, $G \Subset \mathsf{imin\text{-}scope}$ and $H \Subset \mathsf{imin\text{-}scope}$, then (1.) is equivalent to:
3. H is a factual-skeptical abductive consequence of $\langle \mathsf{pstable}, O, S, F, G\rangle$.

Proof. We show the proposition separately for the three cases that *sem* is stable, pstable and wf.

Case $\mathit{sem} = \mathsf{stable}$**.** Recall that $\mathsf{IG}(\mathsf{stable}) = 0$, thus statement (2.) of the theorem expands into

$$\mathsf{stable}_O(F) \wedge \mathsf{gwsc}_{S \cap 0}(\mathsf{stable}_O(F), G) \models H.$$

Let W be a shorthand for $\mathsf{gwsc}_{S \cap 0}(\mathsf{stable}_O(F), G)$. Assume the precondition of the theorem:

(1) $S \subseteq O$ by Def. 3

We reformulate the left side of the theorem by expanding definitions and applying Theorem 12 and Lemma 11 and then show the equivalence to the right side:

(2) H is a factual-skeptical abductive consequence of $\mathfrak{A}$
(3) iff For all factual explanations E for $\mathfrak{A}$ it holds that $\mathsf{stable}_O(F \wedge E) \models H$
(4) iff For all conjunctive clauses $E \Subset S \cap 0$ such that $E \models W$ it holds that $\mathsf{stable}_O(F \wedge E) \models H$ by (1), Thm. 12
(5) iff For all conjunctive clauses $E \Subset S \cap 0$ such that $E \models W$ it holds that $\mathsf{stable}_O(F) \wedge E \models H$. by (1), Lem. 11
(6) iff $(\mathsf{stable}_O(F) \wedge W) \models H$.

Clearly (6) implies (5). It remains to show that also (5) implies (6). Assume (5). Let $(W_1 \vee \ldots \vee W_n)$ be a formula that is equivalent to W and in disjunctive normal form with conjunctive clauses $W_1, \ldots, W_n \Subset S \cap 0$. The existence of such a formula follows from Prop. 10. Then, by (5), for each $i \in \{1, \ldots, n\}$ it follows that $(\mathsf{stable}_O(F) \wedge W_i) \models H$, which implies (6).

Case $sem = \mathsf{pstable}$**.** Let W be a shorthand for $\mathsf{gwsc}_{S \cap \mathsf{IG}(\mathsf{pstable})}(\mathsf{pstable}_O(F), G)$. Assume the precondition of the theorem:

(1) $S \subseteq O$ by Def. 3

We reformulate the left side of the theorem by expanding definitions and applying Theorem 12 and Lemma 11 and then show the equivalence to the right side:

(2) H is a factual-skeptical abductive consequence of $\mathfrak{A}$
(3) iff For all factual explanations $E[+0, -0]$ for $\mathfrak{A}$ it holds that $\mathsf{pstable}_O(F \wedge E[+0, -0]) \models H$
(4) iff For all conjunctive clauses $E[+0, -0] \Subset S \cap 0$ such that $E[+0, -1] \models W$ it holds that $\mathsf{pstable}_O(F \wedge E[+0, -0]) \models H$ by (1), Thm. 12
(5) iff For all conjunctive clauses $E[+0, -0]$ in $S \cap 0$ such that $E[+0, -1] \models W$ it holds that $\mathsf{pstable}_O(F) \wedge E[+0, -1] \models H$. by (1), Lem. 11
(6) iff $\mathsf{pstable}_O(F) \wedge W \models H$.

Clearly (6) implies (5). It remains to show that also (5) implies (6). Assume (5). Let

$$W_1[+0, -1] \vee \ldots \vee W_m[+0, -1] \vee W_{m+1}[+0, -1] \vee \ldots \vee W_n[+0, -1],$$

where $n \geq m \geq 0$, be a formula that is equivalent to W and is in disjunctive normal form with conjunctive clauses $W_1[+0, -1], \ldots, W_n[+0, -1] \Subset S \cap \mathsf{IG}(\mathsf{pstable})$, such that the conjunctions

$$W_1[+0, -0], \ldots, W_m[+0, -0]$$

are conjunctive clauses (hence consistent) and

$$W_{m+1}[+0, -0] \ldots, W_n[+0, -0]$$

are inconsistent conjunctions of literal formulas. The existence of such a DNF follows from Prop. 10 since any projection of a propositional formula is equivalent to a propositional formula in negation normal form whose literals are all in the projection scope. Then, by (5), for each $i \in \{1, \ldots, m\}$ it follows that $\mathsf{pstable}_O(F) \wedge W_i[+0, -1] \models H$. Step (6) then follows since $\mathsf{cons} \wedge (W_1[+0, -1] \vee \ldots \vee W_n[+0, -1]) \equiv \mathsf{cons} \wedge (W_1[+0, -1] \vee \ldots \vee W_m[+0, -1])$ and $\mathsf{pstable}_O(F) \models \mathsf{cons}$.

Case $sem = \mathsf{wf}$**.** Assume the preconditions of the theorem:

(1) $S \subseteq O$. by Def. 3
(2) $G \Subset \mathsf{imin\text{-}scope}$. assumption
(3) $H \Subset \mathsf{imin\text{-}scope}$. assumption

We show the equivalence of both sides of the proposition by expanding and contracting the definition of factual-skeptical abductive consequence, and by applying Prop. 2 and Theorem 12:

(4) H is a factual-skeptical abductive consequence of $\mathfrak{A}$
(5) iff For all factual explanations E for $\mathfrak{A}$ it holds that $\mathsf{wf}_O(F \wedge E) \models H$
(6) iff For all factual explanations E for $\mathfrak{A}$ it holds that $\mathsf{pstable}_O(F \wedge E) \models H$ by (3), Prop. 2
(7) iff For all factual explanations E for $\langle \mathsf{pstable}, O, S, F, G\rangle$ it holds that $\mathsf{pstable}_O(F \wedge E) \models H$ by (2), (1), Prop. 12
(8) iff H is a factual-skeptical abductive consequence of $\langle \mathsf{pstable}, O, S, F, G\rangle$.

The following example shows a case where the abductive explanations and consequences differ, depending on whether the stable model or the partial stable model/well-founded semantics is used.

Example E30 (Abductive Consequences). Let $\mathfrak{A} = \langle sem, O, S, F, G\rangle$ be an abductive setting, where $O = S = \{a, b\}$, $F = (p^0 \leftarrow s^0 \wedge a^0) \wedge (p^0 \leftarrow b^0) \wedge (q^0 \leftarrow b^0) \wedge (r^0 \leftarrow \neg s^1) \wedge (s^0 \leftarrow \neg r^1) \wedge (t^0 \leftarrow \neg t^1 \wedge r^0)$, and $G = p^0$. Intuitively, the rules $(r^0 \leftarrow \neg s^1)$ and $(s^0 \leftarrow \neg r^1)$ express that s or r must hold, and $(t^0 \leftarrow \neg t^1 \wedge r^0)$ that r leads to inconsistency with the stable model semantics, or to undefinedness with the partial stable model semantics, respectively. It holds that $\mathsf{gwsc}_{S \cap \mathsf{IG}(\mathsf{stable})}(F, G) \equiv (a^0 \vee b^0)$. Thus, if $sem = \mathsf{stable}$, then a^0 and b^0 are the two minimal factual explanations for $\mathfrak{A}$. Since $\mathsf{stable}_{\{a,b\}}(F) \wedge (a^0 \vee b^0) \not\models q^0$ it does not hold that q^0 is an abductive consequence of $\mathfrak{A}$. With *partial* stable model and well-founded semantics this is different. It holds that $\mathsf{gwsc}_{S \cap \mathsf{IG}(\mathsf{pstable})}(F, G) \equiv ((a^0 \wedge \neg a^1) \vee b^0)$. If $sem \in \{\mathsf{pstable.wf}\}$, then, since $(a^0 \wedge \neg a^0)$ is inconsistent and thus not a conjunctive clause, b^0 is the only minimal factual abductive explanation for $\mathfrak{A}$. Since $\mathsf{pstable}_{\{a,b\}}(F) \models \mathsf{cons}$, it holds that $\mathsf{pstable}_{\{a,b\}}(F) \wedge ((a^0 \wedge \neg a^1) \vee b^0) \equiv \mathsf{pstable}_{\{a,b\}}(F) \wedge b^0 \models q^0$, and thus q^0 is a factual abductive consequence of $\mathfrak{A}$.