On the Minimality of Stable Models

Paolo Ferraris[1] and Vladimir Lifschitz[2]

[1] Google, USA
[2] Department of Computer Science, University of Texas at Austin, USA

Abstract. The class of logic programs covered by the original definition of a stable model has the property that all stable models of a program in this class are minimal. In the course of research on answer set programming, the concept of a stable model was extended to several new programming constructs, and for some of these extensions the minimality property does not hold. We are interested in syntactic conditions on a logic program that guarantee the minimality of its stable models. This question is addressed here in the context of the general theory of stable models of first-order sentences.

1 Introduction

A Prolog program with negation, viewed as a logical formula, usually has several minimal Herbrand models, and only one of them may reflect the actual behavior of Prolog. For instance, the propositional rule

$$p \leftarrow not\ q,$$

viewed as the formula

$$\neg q \rightarrow p \tag{1}$$

written in logic programming notation, has two minimal models, $\{p\}$ and $\{q\}$; the first of them is the "intended" model. Early research on the semantics of negation as failure [Bidoit and Froidevaux, 1987, Gelfond, 1987, Apt *et al.*, 1988, Van Gelder, 1988, Van Gelder *et al.*, 1988] was motivated by the need to distinguish between the intended model of a logic program and its other minimal models.

The definition of a stable model proposed in [Gelfond and Lifschitz, 1988] had a similar motivation. According to Theorem 1 from that paper, every stable model of a logic program is minimal. The converse, for programs with negation, is usually not true. One corollary to the fact that all stable models are minimal is that the collection of stable models of a program is an antichain: one stable model cannot be a subset of another.

In the course of research on answer set programming, the concept of a stable model was extended to several new programming constructs, and for some of these extensions the antichain property does not hold. Take, for instance, choice

rules, which play an important role in the language of LPARSE.[1] The set of stable models of the program consisting of the single choice rule

$$\{p\}$$

is $\{\emptyset, \{p\}\}$. It is not an antichain. If we identify this choice rule with the formula

$$p \vee \neg p, \qquad (2)$$

as proposed in [Ferraris, 2005], then we can say that the singleton $\{p\}$ is a stable but nonminimal model of (2).

The situation is similar for some cardinality constraints containing negation as failure. The one-rule LPARSE program

$$p \leftarrow \{not\ p\}\ 0$$

has the same stable models as the choice rule above, $\emptyset$ and $\{p\}$. According to [Ferraris, 2005], this program can be identified with the formula

$$\neg\neg p \rightarrow p. \qquad (3)$$

The singleton $\{p\}$ is a nonminimal stable model of (3).

Under what syntactic conditions on a logic program can we assert that every stable model of the program is minimal? What is the essential difference, for instance, between formula (1) on the one hand, and formulas (2) and (3) on the other? In this note we address this question in the context of the general theory of stable models proposed in [Ferraris et al., 2010]. The main definition of that paper, reproduced in the next section, describes the "stable model operator" $\mathrm{SM}_{\mathbf{p}}$, where $\mathbf{p}$ is a tuple of predicate constants. This operator turns any first-order sentence F into a conjunction of F with a second-order sentence. The stable models of F relative to the given choice of "intensional" predicates $\mathbf{p}$ are defined in [Ferraris et al., 2010] as models of $\mathrm{SM}_{\mathbf{p}}[F]$ in the sense of classical logic. The definition of $\mathrm{SM}_{\mathbf{p}}[F]$ is very similar to the definition of the parallel circumscription of $\mathbf{p}$ in F [McCarthy, 1986], which we will denote by $\mathrm{CIRC}_{\mathbf{p}}[F]$. The circumscription formula characterizes the models of F in which the extents of the predicates $\mathbf{p}$ are minimal. Thus the question that we are interested in can be stated as follows: Under what conditions is $\mathrm{CIRC}_{\mathbf{p}}[F]$ entailed by $\mathrm{SM}_{\mathbf{p}}[F]$?

2   Review: Circumscription and Stable Models

This review follows [Ferraris et al., 2010]. We assume that, in the definition of a formula, the propositional connectives $\bot$ (falsity), $\wedge$, $\vee$, $\rightarrow$ are chosen as primitives, and $\neg F$ is treated as an abbreviation for $F \rightarrow \bot$.

Notation: if p and q are predicate constants of the same arity then $p \leq q$ stands for the formula $\forall \mathbf{x}(p(\mathbf{x}) \rightarrow q(\mathbf{x}))$, where $\mathbf{x}$ is a tuple of distinct object

---

[1] See http://www.tcs.hut.fi/Software/smodels/lparse.ps for a description of the language.

variables. If **p** and **q** are tuples $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$ of predicate constants then $\mathbf{p} \leq \mathbf{q}$ stands for the conjunction

$$(p_1 \leq q_1) \wedge \cdots \wedge (p_n \leq q_n),$$

and $\mathbf{p} < \mathbf{q}$ stands for $(\mathbf{p} \leq \mathbf{q}) \wedge \neg(\mathbf{q} \leq \mathbf{p})$. In second-order logic, we apply the same notation to tuples of predicate variables.

Let **p** be a list of distinct predicate constants.[2] The *circumscription operator with the minimized predicates* **p**, denoted by $\text{CIRC}_\mathbf{p}$, is defined as follows: for any first-order sentence F, $\text{CIRC}_\mathbf{p}[F]$ is the second-order sentence

$$F \wedge \neg \exists \mathbf{u}((\mathbf{u} < \mathbf{p}) \wedge F(\mathbf{u})),$$

where **u** is a list of distinct predicate variables of the same length as **p**, and $F(\mathbf{u})$ is the formula obtained from F by substituting the variables **u** for the constants **p**. Models of $\text{CIRC}_\mathbf{p}[F]$ will be called **p**-*minimal* models of F.

Let **p** be a list of distinct predicate constants $p_1, \ldots, p_n$. The *stable model operator with the intensional predicates* **p**, denoted by $\text{SM}_\mathbf{p}$, is defined as follows: for any first-order sentence F, $\text{SM}_\mathbf{p}[F]$ is the second-order sentence

$$F \wedge \neg \exists \mathbf{u}((\mathbf{u} < \mathbf{p}) \wedge F^*(\mathbf{u})),$$

where **u** is a list of n distinct predicate variables $u_1, \ldots, u_n$, and $F^*(\mathbf{u})$ is defined recursively:

- $p_i(\mathbf{t})^* = u_i(\mathbf{t})$ for any tuple $\mathbf{t}$ of terms;
- $F^* = F$ for any atomic formula F that does not contain members of **p**;[3]
- $(F \wedge G)^* = F^* \wedge G^*$;
- $(F \vee G)^* = F^* \vee G^*$;
- $(F \to G)^* = (F^* \to G^*) \wedge (F \to G)$;
- $(\forall x F)^* = \forall x F^*$;
- $(\exists x F)^* = \exists x F^*$.

Models of $\text{SM}_\mathbf{p}[F]$ will be called **p**-*stable* models of F.

It is clear that if F does not contain implication then $F^*(\mathbf{u})$ is identical to $F(\mathbf{u})$, $\text{SM}_\mathbf{p}[F]$ is identical to $\text{CIRC}_\mathbf{p}[F]$, and the class of **p**-stable models of F is identical to the class of **p**-minimal models of F.

Example 1. If F is (1) then $\text{CIRC}_{pq}[F]$ is

$$(\neg q \to p) \wedge \neg \exists uv(((u, v) < (p, q)) \wedge (\neg v \to u))$$

(u, v are propositional variables). This formula is equivalent to

$$(p \wedge \neg q) \vee (\neg p \wedge q),$$

---

[2] In this note, equality is not considered a predicate constant, so that it is not allowed to be a member of **p**.

[3] This includes the case when F is $\bot$.

so that the pq-minimal models of (1) are $\{p\}$ and $\{q\}$. To apply the operator SM_{pq} to formula (1) we need to remember that this formula is shorthand for

$$(q \rightarrow \bot) \rightarrow p.$$

So $F^*(u, v)$ is

$$(((v \rightarrow \bot) \wedge (q \rightarrow \bot)) \rightarrow u) \wedge ((q \rightarrow \bot) \rightarrow p),$$

which can be abbreviated as

$$((\neg v \wedge \neg q) \rightarrow u) \wedge (\neg q \rightarrow p).$$

Then $SM_{pq}[F]$ is

$$(\neg q \rightarrow p) \wedge \neg\exists uv(((u, v) < (p, q)) \wedge ((\neg v \wedge \neg q) \rightarrow u) \wedge (\neg q \rightarrow p)).$$

This formula is equivalent to $p \wedge \neg q$, so that the only pq-stable model of (1) is $\{p\}$.

Example 2. Let F be the formula

$$\forall xy(p(x, y) \rightarrow q(x, y)) \wedge \forall xyz(q(x, y) \wedge q(y, z) \rightarrow q(x, z)).$$

Then $CIRC_q[F]$ is the conjunction of F with the formula

$$\neg\exists u((u < q) \wedge \forall xy(p(x, y) \rightarrow u(x, y)) \wedge \forall xyz(u(x, y) \wedge u(y, z) \rightarrow u(x, z)))$$

(u is a binary predicate variable). This conjunction expresses that q is the transitive closure of p. Furthermore, $SM_q[F]$ is the conjunction of F with the formula

$$\begin{aligned}
\neg\exists u((u < q) &\wedge \forall xy(p(x, y) \rightarrow u(x, y)) \\
&\wedge \forall xy(p(x, y) \rightarrow q(x, y)) \\
&\wedge \forall xyz(u(x, y) \wedge u(y, z) \rightarrow u(x, z)) \\
&\wedge \forall xyz(q(x, y) \wedge q(y, z) \rightarrow q(x, z))).
\end{aligned}$$

This conjunction is equivalent to $CIRC_q[F]$, and consequently it expresses the same relationship: q is the transitive closure of p.[4]

3   Critical Subformulas

Recall that an occurrence of a symbol in a formula is called *strictly positive* if it does not belong to the antecedent of any implication.[5]

About an occurrence of a formula G in a first-order formula F we say that it is *critical* if it is the antecedent of an implication $G \rightarrow H$ in F, and this implication

---

[4] Formula F in this example is a canonical theory in the sense of [Kim *et al.*, 2009, Section 4]. The fact that the stable models of this formula are identical to its minimal models is a special case of Proposition 2 from that paper.

[5] We do not add "and is not in a scope of a negation" because $\neg F$ is treated here as an abbreviation.

(i) is in the scope of a strictly positive $\vee$, or
(ii) is in the scope of a strictly positive $\exists$, or
(iii) belongs to the antecedents of at least two other implications.

Theorem 1. *For any first-order sentence F and any list $\mathbf{p}$ of predicate constants, if members of $\mathbf{p}$ do not occur in critical subformulas of F then every $\mathbf{p}$-stable model of F is $\mathbf{p}$-minimal.*

In other words, under the condition on critical subformulas above, $\mathrm{SM}_{\mathbf{p}}[F]$ entails $\mathrm{CIRC}_{\mathbf{p}}[F]$.

For example, formula (1) has no critical subformulas, so that the condition in the statement of the theorem holds in this case trivially. The same can be said about the formula from Example 2.[6] The second occurrence of p in (2) is critical (recall that $\neg p$ is shorthand for $p \rightarrow \bot$), as well as the first occurrence of p in (3). Consequently neither (2) nor (3) is covered by our theorem, which could be expected: each of these formulas has a nonminimal stable model.

Logic programs in the sense of [Gelfond and Lifschitz, 1988] are, from the perspective of [Ferraris *et al.*, 2010], conjunctions of the universal closures of formulas of the form

$$L_1 \wedge \cdots \wedge L_m \rightarrow A,$$

where $L_1, \ldots, L_m$ $(m \geq 0)$ are literals, and A is an atom. These formulas do not have critical subformulas. (Thus our Theorem 1 can be viewed as a generalization of Theorem 1 from [Gelfond and Lifschitz, 1988].) The same can be said about "disjunctive logic programs"—conjunctions of the universal closures of formulas of the form

$$L_1 \wedge \cdots \wedge L_m \rightarrow A_1 \vee \cdots \vee A_n, \tag{4}$$

where $L_1, \ldots, L_m$ are literals, and $A_1, \ldots, A_n$ are atoms $(m, n \geq 0)$.

Proposition 1 below gives an example of a formula without critical subformulas that is "very different" from disjunctive logic programs. Recall that first-order formulas F and G are *strongly equivalent* to each other if, for any formula H, any occurrence of F in H, and any list $\mathbf{p}$ of distinct predicate constants, $\mathrm{SM}_{\mathbf{p}}[H]$ is equivalent to $\mathrm{SM}_{\mathbf{p}}[H']$, where H' is obtained from H by replacing the occurrence of F by G [Ferraris *et al.*, 2010, Section 5.2].

Proposition 1. *No conjunction of propositional formulas of the form (4) is strongly equivalent to $(p \rightarrow q) \rightarrow q$.*

If we drop any of conditions (i)–(iii) from the definition of a critical subformula then the assertion of the theorem will become incorrect. The need to include condition (i) is illustrated by formula (2). Formula (3) shows that (iii) is required. The need for (ii) follows from the following proposition:

---

[6] More generally, the examination of the definition of a canonical theory from [Kim *et al.*, 2009] shows that our condition on critical subformulas holds for all canonical theories.

Proposition 2. *Formula*

$$p(a) \wedge (q(a) \rightarrow p(b)) \wedge \exists x (p(x) \rightarrow q(x)) \tag{5}$$

has a pq-stable model that is not pq-minimal.

4   Proofs

In Lemmas 1–4, F is a first-order formula, $\mathbf{p}$ is a tuple of distinct predicate constants, and $\mathbf{u}$ is a tuple of distinct predicate variables of the same length as $\mathbf{p}$.

Lemma 1. *If F does not contain members of $\mathbf{p}$ then $F^*(\mathbf{u})$ is equivalent to F.*

Proof. Immediate by structural induction.

Lemma 2. *If all occurrences of members of $\mathbf{p}$ in F are strictly positive then the formula*

$$(\mathbf{u} \leq \mathbf{p}) \wedge F(\mathbf{u}) \rightarrow F^*(\mathbf{u})$$

is logically valid.

Proof. By structural induction. The only nontrivial case is when F has the form $G \rightarrow H$; G does not contain members of $\mathbf{p}$, and all occurrences of members of $\mathbf{p}$ in H are strictly positive. By the induction hypothesis, the formula

$$(\mathbf{u} \leq \mathbf{p}) \wedge H(\mathbf{u}) \rightarrow H^*(\mathbf{u}) \tag{6}$$

is logically valid. Assume $(\mathbf{u} \leq \mathbf{p}) \wedge F(\mathbf{u})$, that is,

$$(\mathbf{u} \leq \mathbf{p}) \wedge (G \rightarrow H(\mathbf{u})). \tag{7}$$

We need to derive $F^*(\mathbf{u})$, that is,

$$(G^*(\mathbf{u}) \rightarrow H^*(\mathbf{u})) \wedge (G \rightarrow H).$$

In view of Lemma 1, this formula is equivalent to

$$G \rightarrow (H^*(\mathbf{u}) \wedge H).$$

Assume G. Then, by (7),

$$(\mathbf{u} \leq \mathbf{p}) \wedge H(\mathbf{u}),$$

and, by (6), $H^*(\mathbf{u})$. The formula

$$(\mathbf{u} \leq \mathbf{p}) \wedge H^*(\mathbf{u}) \rightarrow H \tag{8}$$

is logically valid [Ferraris *et al.*, 2010, Lemma 5]. Consequently H follows as well.

Lemma 3. *If no occurrence of any member of* $\mathbf{p}$ *in* F *belongs to the antecedent of more than one implication then the formula*

$$(\mathbf{u} \leq \mathbf{p}) \wedge F^*(\mathbf{u}) \rightarrow F(\mathbf{u})$$

is logically valid.

Proof. By structural induction. The only nontrivial case is when F has the form $G \rightarrow H$; all occurrences of members of $\mathbf{p}$ in G are strictly positive, and no occurrence of any member of $\mathbf{p}$ in H belongs to the antecedent of more than one implication. By Lemma 2, the formula

$$(\mathbf{u} \leq \mathbf{p}) \wedge G(\mathbf{u}) \rightarrow G^*(\mathbf{u}) \tag{9}$$

is logically valid. By the induction hypothesis, the formula

$$(\mathbf{u} \leq \mathbf{p}) \wedge H^*(\mathbf{u}) \rightarrow H(\mathbf{u}) \tag{10}$$

is logically valid. Assume $(\mathbf{u} \leq \mathbf{p}) \wedge F^*(\mathbf{u})$, that is,

$$(\mathbf{u} \leq \mathbf{p}) \wedge (G^*(\mathbf{u}) \rightarrow H^*(\mathbf{u})) \wedge (G \rightarrow H). \tag{11}$$

Our goal is to prove $G(\mathbf{u}) \rightarrow H(\mathbf{u})$. From $G(\mathbf{u})$, the first conjunctive term of (11), and (9), $G^*(\mathbf{u})$. Then, by the second conjunctive term of (11), $H^*(\mathbf{u})$. Then $H(\mathbf{u})$ follows by (10).

Lemma 4. *If members of* $\mathbf{p}$ *do not occur in critical subformulas of* F *then the formula*

$$\mathbf{u} \leq \mathbf{p} \wedge F \wedge F(\mathbf{u}) \rightarrow F^*(\mathbf{u})$$

is logically valid.

Proof. By induction on F. There are three nontrivial cases: when F is $G \vee H$, $G \rightarrow H$, or $\exists x G(x)$. If F is $G \vee H$ or $\exists x G(x)$ then the antecedents of all implications occurring in F are critical and consequently do not contain members of $\mathbf{p}$. Thus all occurrences of members of $\mathbf{p}$ in F are strictly positive, and the assertion to be proved follows from Lemma 2. Let F be $G \rightarrow H$. In formula G, no occurrence of any member of $\mathbf{p}$ belongs to the antecedent of more than one implication, because otherwise the antecedent of the innermost implication containing that occurrence would be critical in F. By Lemma 3, it follows that the formula

$$(\mathbf{u} \leq \mathbf{p}) \wedge G^*(\mathbf{u}) \rightarrow G(\mathbf{u}) \tag{12}$$

is logically valid. By the induction hypothesis, the formula

$$\mathbf{u} \leq \mathbf{p} \wedge H \wedge H(\mathbf{u}) \rightarrow H^*(\mathbf{u}) \tag{13}$$

is logically valid. Assume

$$\mathbf{u} \leq \mathbf{p} \wedge F \wedge F(\mathbf{u}); \tag{14}$$

our goal is to derive $F^*(\mathbf{u})$, that is,

$$(G^*(\mathbf{u}) \to H^*(\mathbf{u})) \wedge F.$$

The second conjunctive term is immediate from (14). To prove the first conjunctive term, assume $G^*(\mathbf{u})$. Then, by the first conjunctive term of (14) and (12), $G(\mathbf{u})$. Consequently, by the third conjunctive term of (14), $H(\mathbf{u})$. On the other hand, the formula

$$(\mathbf{u} \le \mathbf{p}) \wedge G^*(\mathbf{u}) \to G$$

is logically valid [Ferraris *et al.*, 2010, Lemma 5]; hence G, and, by the second conjunctive term of (14), H. Then, by (13), $H^*(\mathbf{u})$.

Proof of Theorem 1. Take a sentence F such that members of $\mathbf{p}$ do not occur in critical subformulas of F. We need to show that $\mathrm{SM}_{\mathbf{p}}[F]$ entails $\mathrm{CIRC}_{\mathbf{p}}[F]$. Assume that

$$F \wedge \neg\exists\mathbf{u}((\mathbf{u} < \mathbf{p}) \wedge F^*(\mathbf{u})) \tag{15}$$

but

$$(\mathbf{u} < \mathbf{p}) \wedge F(\mathbf{u}).$$

Then, by Lemma 4, $F^*(\mathbf{u})$, which contradicts (15).

The proof of Proposition 1 below refers to reducts in the sense of [Ferraris, 2005]. It uses two facts about them. One is a characterization of strong equivalence in terms of reducts:

Lemma 5. *Propositional formulas F and G are strongly equivalent to each other iff, for every set X of atoms, the reducts F^X and G^X are equivalent to each other in the sense of classical logic.*

This is part of the statement of Proposition 2 from [Ferraris, 2005].[7]

The second fact is a property of disjunctive logic programs:

Lemma 6. *If Π is a conjunction of propositional formulas of form (4) then, for any sets X, Y and Z of atoms such that $X \subseteq Y \subseteq Z$, if $Z \models \Pi$ and $X \models \Pi^Y$ then $X \models \Pi^Z$.*

This observation is made in [Eiter *et al.*, 2005] (and expressed there using somewhat different terminology, in terms of SE-models instead of reducts).

Proof of Proposition 1. Let F stand for $(p \to q) \to q$, and assume that Π is a conjunction of propositional formulas of form (4) that is strongly equivalent to F. It is easy to check that $\{p, q\} \models F$ and that

$$\emptyset \models F^{\{p\}}, \quad \emptyset \not\models F^{\{p,q\}}.$$

---

[7] The definition of strong equivalence in [Ferraris, 2005] is somewhat different from the definition given above, which is taken from [Ferraris *et al.*, 2010]. But in application to propositional formulas the two definitions are equivalent to each other, because, as discussed in these papers, each definition is equivalent in this case to the provability of $F \leftrightarrow G$ in the logic of here-and-there.

It follows that $\{p, q\} \models \Pi$ (because strongly equivalent formulas are classically equivalent) and, by Lemma 5, that

$$\emptyset \models \Pi^{\{p\}}, \; \emptyset \not\models \Pi^{\{p,q\}}.$$

But this is impossible by Lemma 6: take $X = \emptyset$, $Y = \{p\}$, and $Z = \{p, q\}$.

Proof of Proposition 2 (Hint). The Herbrand interpretation $\{p(a), p(b), q(a)\}$ is a *pq*-stable model of (5) that is not *pq*-minimal.

5   Conclusion

In this note, we gave a syntactic condition that ensures the minimality of all stable models of a first-order sentence. The condition is expressed in terms of critical subformulas. It shows that in the propositional case all possible exceptions to the general principle that stable models are minimal are similar to the examples given in the introduction: they contain an implication in the scope of a disjunction, as (2), or three implications nested within each other, as (3). In the presence of variables, the restriction on disjunctions has to be extended to existential quantifiers.

Acknowledgements

References

[Apt *et al.*, 1988]Apt, K., Blair, H., Walker, A.: Towards a theory of declarative knowledge. In: Minker, J. (ed.) Foundations of Deductive Databases and Logic Programming, pp. 89–148. Morgan Kaufmann, San Mateo (1988)

[Bidoit and Froidevaux, 1987]Bidoit, N., Froidevaux, C.: Minimalism subsumes default logic and circumscription in stratified logic programming. In: Proceedings LICS 1987, pp. 89–97 (1987)

[Eiter *et al.*, 2005]Eiter, T., Tompits, H., Woltran, S.: On solution correspondences in answer-set programming. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), pp. 97–102 (2005)

[Ferraris *et al.*, 2010]Ferraris, P., Lee, J., Lifschitz, V.: Stable models and circumscription[8]. Artificial Intelligence 175, 236–263 (2011)

[Ferraris, 2005]Ferraris, P.: Answer sets for propositional theories. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 119–131. Springer, Heidelberg (2005)

---

[8] `http://peace.eas.asu.edu/joolee/papers/smcirc.pdf`

[Gelfond and Lifschitz, 1988]Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R., Bowen, K. (eds.) Proceedings of International Logic Programming Conference and Symposium, pp. 1070–1080. MIT Press, Cambridge (1988)

[Gelfond, 1987]Gelfond, M.: On stratified autoepistemic theories. In: Proceedings of National Conference on Artificial Intelligence (AAAI), pp. 207–211 (1987)

[Kim *et al.*, 2009]Kim, T.-W., Lee, J., Palla, R.: Circumscriptive event calculus as answer set programming. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), pp. 823–829 (2009)

[McCarthy, 1986]McCarthy, J.: Applications of circumscription to formalizing common sense knowledge. Artificial Intelligence 26(3), 89–116 (1986)

[Van Gelder *et al.*, 1988]Van Gelder, A., Ross, K., Schlipf, J.: Unfounded sets and well-founded semantics for general logic programs. In: Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Austin, Texas, March 21-23, pp. 221–230. ACM Press, New York (1988)

[Van Gelder, 1988]Van Gelder, A.: Negation as failure using tight derivations for general logic programs. In: Minker, J. (ed.) Foundations of Deductive Databases and Logic Programming, pp. 149–176. Morgan Kaufmann, San Mateo (1988)