# *Order-Consistent Programs are Cautiously Monotonic*

HUDSON TURNER
*Computer Science Department*
*University of Minnesota, Duluth*
hudson@d.umn.edu

**Abstract**

Some normal logic programs under the answer set (or stable model) semantics lack the appealing property of "cautious monotonicity." That is, augmenting a program with one of its consequences may cause it to lose another of its consequences. The syntactic condition of "order-consistency" was shown by Fages to guarantee existence of an answer set. This note establishes that order-consistent programs are not only consistent, but cautiously monotonic. From this it follows that they are also "cumulative." That is, augmenting an order-consistent program with some of its consequences does not alter its consequences. In fact, as we show, its answer sets remain unchanged.

## 1 Introduction

The answer set (or stable model) semantics of normal logic programs (Gelfond and Lifschitz 1988, 1991) does not satisfy cautious monotonicity. That is, even if atoms $a$ and $c$ are among the consequences of a program $P$, $a$ may fail to be a consequence of the program $P \cup \{c \leftarrow\}$. Here is an example due to Dix (1991), who has published many studies of such properties for various logic programming semantics.

$$a \leftarrow not\ b$$
$$b \leftarrow c, not\ a$$
$$c \leftarrow a$$

This program has only one answer set $\{a, c\}$, and so has $a$ and $c$ among its consequences. When augmented with the rule $c \leftarrow$ the program gains a second answer set $\{b, c\}$, and loses consequence $a$.

A syntactic condition known as "order-consistency" (Sato 1990) was shown by Fages (1994) to guarantee consistency of normal programs under the answer set semantics. In this note we establish that order-consistency guarantees another nice property: cautious monotonicity.

*Theorem 1* (*Cautious Monotonicity Theorem*)
If $P$ is an order-consistent program and atom $a$ belongs to every answer set for $P$, then every answer set for program $P \cup \{a \leftarrow\}$ is an answer set for $P$.

All normal programs under the answer set semantics have a property complementary to cautious monotonicity, commonly called "cut": augmenting a program with one of its consequences cannot cause it to gain a consequence. This is immediate, given the following easily proved fact.

*Fact 1*
If an atom $a$ belongs to an answer set $X$ for a program $P$, then $X$ is an answer set for program $P \cup \{a \leftarrow\}$.

Cut and cautious monotonicity together imply another nice property, called "cumulativity": augmenting a program with one of its consequences does not alter its consequences. Corresponding to this, we have the following result for order-consistent programs.

*Corollary 1* (*Cumulativity Corollary*)
If an atom $a$ belongs to every answer set for an order-consistent program $P$, then programs $P$ and $P \cup \{a \leftarrow\}$ have the same answer sets.

Semantic properties such as cumulativity, cut and cautious monotonicity were originally formulated more generally for analysis of consequence relations lacking the classic monotonicity property (Gabbay 1985, Makinson 1989, Kraus, Lehmann and Magidor 1990). Makinson's (1993) handbook article includes a survey of such properties for nonmonotonic logics used in AI, among them logic programming under the stable model (answer set) semantics.

The remainder of this note is devoted to a proof of the Cautious Monotonicity Theorem (and also, of course, to recalling the definitions involved in its statement). Here is a preliminary sketch. We first observe that adding a consequence to a "signed" program does not alter its answer sets. (This follows from results due to Dung (1992) and Schlipf.) We then recall a result from (Lifschitz and Turner 1994) that characterizes the answer sets $X$ for an order-consistent program $P$ in terms of families of signed programs whose answer sets correspond to a partition of $X$. In the proof we establish in addition that if an atom $a$ is a consequence of order-consistent program $P$, then $a$ is a consequence of the corresponding member of each of the families of signed programs. It follows that adding rule $a \leftarrow$ to $P$, and so to the corresponding member of each of the families of signed programs, does not affect the answer sets for the members of the families of signed programs. We can then conclude, by the Splitting Sequence Theorem of (Lifschitz and Turner 1994), that each answer set for $P \cup \{a \leftarrow\}$ is an answer set for $P$.

## 2 Normal Logic Programs

Begin with a set of symbols called *atoms*. A *rule* consists of three parts: an atom called the *head*, and two finite sets of atoms—the set of *positive subgoals* and the set of *negated subgoals*. The rule with head $a$, positive subgoals $b_1, \ldots, b_m$ and negated subgoals $c_1, \ldots, c_n$ is typically written

$$a \leftarrow b_1, \ldots, b_m, not\ c_1, \ldots, not\ c_n.$$

We denote the three parts of a rule $r$ by $head(r)$, $pos(r)$ and $neg(r)$; $atoms(r)$ stands for $\{head(r)\} \cup pos(r) \cup neg(r)$.

A *program* is a set of rules. For any program $P$, by $atoms(P)$ we denote the union of the sets $atoms(r)$ for all $r \in P$; the atoms in this set are said to *occur* in $P$.

A program $P$ is *positive* if, for every rule $r \in P$, $neg(r) = \emptyset$. The notion of an answer set is first defined for positive programs, as follows. A set $X$ of atoms is *closed* under a positive program $P$ if, for every rule $r \in P$ such that $pos(r) \subseteq X$, $head(r) \in X$. The *answer set* for a positive program $P$ is the least set of atoms closed under $P$.

Now let $P$ be an arbitrary program and $X$ a set of atoms. For each rule $r \in P$ such that $neg(r) \cap X = \emptyset$, let $r'$ be the rule defined by

$$head(r') = head(r)\,,\ pos(r') = pos(r)\,,\ neg(r') = \emptyset\,\cdot$$

The positive program consisting of all rules $r'$ obtained in this way is the *reduct* of $P$ relative to $X$, denoted by $P^X$. We say $X$ is an *answer set* for $P$ if $X$ is the answer set for $P^X$.

A program is *consistent* if it has an answer set. An atom is a *consequence* of a program $P$ if it belongs to all answer sets for $P$. We write $Cn(P)$ to denote the set of all consequences of $P$.

We'll want an auxiliary notion, related to the well-founded semantics of logic programs (Van Gelder, Ross and Schlipf 1991). For any program $P$, let $\Gamma_P$ be the operator that maps a set $X$ of atoms to the answer set for $P^X$. Clearly, the answer sets for $P$ are exactly the fixpoints of $\Gamma_P$. It is well-known that $\Gamma_P^2$ is a monotone operator whose least fixpoint, which we'll denote by $WF(P)$, is exactly the set of atoms true in the well-founded model of $P$.

## 3 Signed Programs are Cautiously Monotonic

A program $P$ is *cautiously monotonic* if, for all $a \in Cn(P)$,

$$Cn(P) \subseteq Cn(P \cup \{a \leftarrow\})\,\cdot$$

A program $P$ is *cumulative* if, for all $a \in Cn(P)$,

$$Cn(P) = Cn(P \cup \{a \leftarrow\})\,\cdot$$

We are interested in a stronger property: for all $a \in Cn(P)$, programs $P$ and $P \cup \{a \leftarrow\}$ have the same answer sets.

To see that this is indeed a stronger property, notice that adding the rule $a \leftarrow$ to program $\{a \leftarrow not\ a\}$ changes its answer sets, but not its consequences.

The notion of a "signing" of a program is due to Kunen (1989). A program $P$ is *signed* if there is a set $S$ of atoms such that, for every rule $r \in P$,

- if $head(r) \in S$ then $pos(r) \subseteq S$ and $neg(r) \cap S = \emptyset$,
- if $head(r) \notin S$ then $pos(r) \cap S = \emptyset$ and $neg(r) \subseteq S$.

The following program $P_1$ is signed.

$$a \leftarrow not\ b$$
$$b \leftarrow not\ a$$

Take $S = \{a\}$, for instance.

*Lemma 2* (*Signing Lemma*)
For any signed program $P$ and $a \in Cn(P)$, programs $P$ and $P \cup \{a \leftarrow\}$ have the same answer sets.

This is immediate, given the following two results.

*Proposition 1*
(Dung 1992) For any signed program $P$, $Cn(P) = WF(P)$.

*Proposition 2*
(Schlipf, personal communication) For any program $P$ and $a \in WF(P)$, programs $P$ and $P \cup \{a \leftarrow\}$ have the same answer sets.

Proposition 1 follows from a stronger result in (Dung 1992). Proposition 2 is apparently widely known, and plays a significant role in automated systems for answer set programming.

## 4 Order-Consistent Programs

For any program $P$ and atom $a$, $P_a^+$ and $P_a^-$ are the smallest sets of atoms such that $a \in P_a^+$ and, for every rule $r \in P$,

- if $head(r) \in P_a^+$ then $pos(r) \subseteq P_a^+$ and $neg(r) \subseteq P_a^-$,
- if $head(r) \in P_a^-$ then $pos(r) \subseteq P_a^-$ and $neg(r) \subseteq P_a^+$.

Intuitively, $P_a^+$ is the set of atoms on which atom $a$ depends positively in $P$, and $P_a^-$ is the set of atoms on which atom $a$ depends negatively in $P$.

A *level mapping* is a function from atoms to ordinals.

A program $P$ is *order-consistent* if there is a level mapping $\lambda$ such that $\lambda(b) < \lambda(a)$ whenever $b \in P_a^+ \cap P_a^-$. That is, if $a$ depends both positively and negatively on $b$, then $b$ is mapped to a lower stratum.

*Theorem 2* (*Fages' Theorem*)
(Fages 1994) Order-consistent programs are consistent.

The following program $P_2$ is order-consistent.

$$a \leftarrow not\ b$$
$$b \leftarrow not\ a$$
$$c \leftarrow a$$
$$c \leftarrow b$$

Consider, for example, the level mapping $\lambda(a) = \lambda(b) = 0$, $\lambda(c) = 1$.

Clearly every signed program is order-consistent. As program $P_2$ illustrates, the converse does not hold.

## 5 Call-Consistent Programs are not Cautiously Monotonic

For finite programs, order-consistency is equivalent to a well-known, simpler condition: a program $P$ is *call-consistent* if for all $a \in atoms(P)$, $a \notin P_a^-$. That is, no atom depends negatively on itself.

The following (infinite) program is call-consistent, but not order-consistent.

$$a_m \leftarrow not\ c, not\ a_n \qquad (0 \leq m < n)$$

This program has no answer set, so $c$ and $a_0$ are among its consequences. Adding the rule $c \leftarrow$ produces a single answer set $\{c\}$ and thus eliminates consequence $a_0$. This shows that not all call-consistent programs are cautiously monotonic.

One may wonder at this point if all *consistent* call-consistent programs are cautiously monotonic. Consider adding the following rules to the previous example.

$$c \leftarrow a$$
$$a \leftarrow not\ b$$
$$b \leftarrow not\ a$$

The resulting program has a single answer set $\{a, c\}$. Adding the rule $c \leftarrow$ yields a second answer set $\{b, c\}$.

## 6 Splitting Sequences

In order to "decompose" an order-consistent program into a family of signed programs and reason about the result, we need some machinery. The definitions given in this section and the next simplify (slightly) those from (Lifschitz and Turner 1994), which applied also to non-normal programs (with classical negation and disjunction).

A *splitting set* for a program $P$ is any set $U$ of atoms such that, for every rule $r \in P$, if $head(r) \in U$ then $atoms(r) \subseteq U$.

It is clear that for any program $P$, both $\emptyset$ and $atoms(P)$ are splitting sets. For program $P_2$ from Section 4, another splitting set is $\{a, b\}$.

Let $U$ and $X$ be sets of atoms and $P$ a program. The set of rules $r \in P$ such that $atoms(r) \subseteq U$ is denoted by $b_U(P)$. For each rule $r \in P \setminus b_U(P)$ such that $pos(r) \cap U \subseteq X$ and $neg(r) \cap X = \emptyset$, take the rule $r'$ defined by

$$head(r') = head(r)\,,\ pos(r') = pos(r) \setminus U\,,\ neg(r') = neg(r) \setminus U\,.$$

The program consisting of all rules $r'$ obtained in this way is denoted by $e_U(P, X)$.

For example, if $U = \{a, b\}$ then $b_U(P_2)$ is exactly the signed program $P_1$ considered previously, and $e_U(P_2, \{a\}) = \{c \leftarrow\} = e_U(P_2, \{b\})$.

A (*transfinite*) *sequence* is a family whose index set is an initial segment of ordinals, $\{\alpha : \alpha < \mu\}$. A sequence $\langle U_\alpha \rangle_{\alpha<\mu}$ of sets is *monotone* if $U_\alpha \subseteq U_\beta$ whenever $\alpha < \beta$, and *continuous* if, for each limit ordinal $\alpha < \mu$, $U_\alpha = \bigcup_{\gamma<\alpha} U_\gamma$.

A *splitting sequence* for a program $P$ is a nonempty, monotone, continuous sequence $\langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $P$ such that $\bigcup_{\alpha<\mu} U_\alpha = atoms(P)$.

For example, $\langle \{a, b\}, \{a, b, c\} \rangle$ is a splitting sequence for program $P_2$.

Let $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for a program $P$. A *solution* to $P$ (with respect to $U$) is a sequence $\langle X_\alpha \rangle_{\alpha<\mu}$ of sets of atoms such that

- $X_0$ is an answer set for $b_{U_0}(P)$,
- for any $\alpha$ such that $\alpha + 1 < \mu$, $X_{\alpha+1}$ is an answer set for

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(P), \bigcup_{\gamma\leq\alpha} X_\gamma\right),$$

- for any limit ordinal $\alpha < \mu$, $X_\alpha = \emptyset$.

Notice, for example, that program $P_2$ has two solutions with respect to splitting sequence $\langle \{a,b\}, \{a,b,c\}\rangle$: $\langle \{a\}, \{c\}\rangle$ and $\langle \{b\}, \{c\}\rangle$. They correspond to the two answer sets for $P_2$, as described in the following general theorem.

*Theorem 3* (*Splitting Sequence Theorem*)
(Lifschitz and Turner 1994) Let $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for a program $P$. A set $X$ of atoms is an answer set for $P$ if and only if

$$X = \bigcup_{\alpha<\mu} X_\alpha$$

for some solution $\langle X_\alpha \rangle_{\alpha<\mu}$ to $P$ with respect to $U$.

Let $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for a program $P$. A sequence $\langle X_\alpha \rangle_{\alpha<\mu}$ of sets of atoms "decomposes" $P$ into the following family of programs.

$$b_{U_0}(P) \tag{1}$$

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(P), \bigcup_{\gamma\leq\alpha} X_\gamma\right) \qquad (\alpha+1<\mu) \tag{2}$$

Every atom occurring in (1) belongs to $U_0$, and for every $\alpha + 1 < \mu$, every atom occurring in (2) belongs to $U_{\alpha+1} \setminus U_\alpha$. Consequently, the members of any solution are answer sets for a family of programs no two of which have an atom in common.

## 7 Signed Components of Order-Consistent Programs

We are interested in the syntactic form of the programs (1) and (2) whose answer sets can be members of a solution to an order-consistent program $P$. It is clear that each rule of each of these programs is obtained from a rule of $P$ by removing some of its subgoals. A more specific claim can be made using the following terminology.

For any program $P$ and set $X$ of atoms, let $rm(P, X)$ be the program obtained from $P$ by removing from each of the rules of $P$ all subgoals, both positive and negated, that belong to $X$. For any program $P$ and splitting sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ for $P$, the programs

$$b_{U_0}(P),$$
$$rm\left(b_{U_{\alpha+1}}(P) \setminus b_{U_\alpha}(P), U_\alpha\right) \qquad (\alpha+1<\mu)$$

will be called the *U-components* of $P$.

It is easy to see that, for any set $X$ of atoms,

$$e_{U_\alpha}(b_{U_{\alpha+1}}(P), X) \subseteq rm(b_{U_{\alpha+1}}(P) \setminus b_{U_\alpha}(P), U_\alpha)\,.$$

Consequently, each of the programs (1) and (2) is a subset of the corresponding $U$-component of $P$.

In (Lifschitz and Turner 1994) we showed that a program is stratified if and only if it has a splitting sequence $U$ such that all $U$-components are positive. We also established the following characterization of order-consistent programs.

*Proposition 3*
(Lifschitz and Turner 1994) A program $P$ is order-consistent if and only if it has a splitting sequence $U$ such that all $U$-components of $P$ are signed.

For example, if $U = \langle\{a, b\}, \{a, b, c\}\rangle$, then the $U$-components of $P_2$ are the signed programs $P_1$ and $\{c \leftarrow\}$.

As discussed in (Lifschitz and Turner 1994), Proposition 3 and the Splitting Sequence Theorem can be used to derive Fages' Theorem from a similar—and easier—result for signed programs. Below they are used instead in the proof that order-consistent programs are cautiously monotonic.

## 8 Proof of Cautious Monotonicity Theorem

*Restatement of Theorem 1.* If $P$ is an order-consistent program and $a \in Cn(P)$, then every answer set for program $P \cup \{a \leftarrow\}$ is an answer set for $P$.

*Proof*
Assume $P$ is order-consistent and $a \in Cn(P)$. Let $X$ be an answer set for $P \cup \{a \leftarrow\}$. Since $P$ is order-consistent, so is $P \cup \{a \leftarrow\}$. By Proposition 3, there is a splitting sequence $U = \langle U_\alpha\rangle_{\alpha<\mu}$ for $P \cup \{a \leftarrow\}$ such that all $U$-components of $P \cup \{a \leftarrow\}$ are signed. Notice that $U$ is also a splitting sequence for $P$, and that all $U$-components of $P$ are signed as well. By the Splitting Sequence Theorem, there is a solution $\langle X_\alpha\rangle_{\alpha<\mu}$ to $P \cup \{a \leftarrow\}$ with respect to $U$ such that $X = \bigcup_{\alpha<\mu} X_\alpha$. We complete the proof by showing that $\langle X_\alpha\rangle_{\alpha<\mu}$ is a solution to $P$ with respect to $U$. (From this it follows, again by the Splitting Sequence Theorem, that $X$ is an answer set for $P$.)

Observe that any splitting sequence can be "extended" by inserting $\emptyset$ at its beginning. That is, since $\langle U_\alpha\rangle_{\alpha<\mu}$ is a splitting sequence for $P$ and $P \cup \{a \leftarrow\}$, so is $\langle U'_\alpha\rangle_{\alpha<\mu+1}$, where

- $U'_0 = \emptyset$,
- for all natural numbers $n$ such that $n + 1 < \mu$, $U'_{n+1} = U_n$,
- for all ordinals $\alpha$ such that $\omega \leq \alpha < \mu$, $U'_\alpha = U_\alpha$,
- $U'_\mu = atoms(P)$.

Notice that since all $U$-components of $P$ and $P \cup \{a \leftarrow\}$ are signed, so are all $U'$-components. For convenience then, we will assume, without loss of generality, that $U_0 = \emptyset$. Under this assumption, any atom that occurs in $P$ belongs to one of the sets $U_{\alpha+1} \setminus U_\alpha$ (for some $\alpha$ such that $\alpha + 1 < \mu$).

Let $\alpha$ be such that $a \in U_{\alpha+1} \setminus U_\alpha$. For all $\beta + 1 < \mu$ such that $\beta \neq \alpha$,

$$e_{U_\beta}\left(b_{U_{\beta+1}}(P), \bigcup_{\gamma \leq \beta} X_\gamma\right) \quad = \quad e_{U_\beta}\left(b_{U_{\beta+1}}(P \cup \{a \leftarrow\}), \bigcup_{\gamma \leq \beta} X_\gamma\right).$$

Hence, we can show that $\langle X_\alpha \rangle_{\alpha<\mu}$ is a solution to $P$ with respect to $U$ simply by showing that $X_{\alpha+1}$ is an answer set for

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(P), \bigcup_{\gamma \leq \alpha} X_\gamma\right). \tag{3}$$

We will do this by showing that (3) has the same answer sets as

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(P \cup \{a \leftarrow\}), \bigcup_{\gamma \leq \alpha} X_\gamma\right).$$

First, notice that the latter program is the same as

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(P), \bigcup_{\gamma \leq \alpha} X_\gamma\right) \cup \{a \leftarrow\}.$$

So it is enough to show that adding the rule $a \leftarrow$ to (3) does not affect its answer sets. Since (3) is a signed program, we can use the Signing Lemma: it remains only to show that atom $a$ is among the consequences of (3).

Take $V_0 = U_\alpha$, $V_1 = U_{\alpha+1}$ and $V_2 = atoms(P)$, The sequence $V = \langle V_0, V_1, V_2 \rangle$ is a splitting sequence for $P$. We construct a solution to $P$ with respect to $V$ as follows. Take $Y_0 = \bigcup_{\gamma \leq \alpha} X_\gamma$. It is straightforward, using the Splitting Sequence Theorem, to verify that $Y_0$ is an answer set for $b_{V_0}(P)$. Notice that $e_{V_0}(b_{V_1}(P), Y_0)$ is exactly the program (3). Since (3) is signed, it is consistent. Let $Y_1$ be one of its answer sets. Since $P$ is order-consistent, so is $e_{V_1}(b_{V_2}(P), Y_0 \cup Y_1)$, and, by Fages' Theorem, it too is consistent. Let $Y_2$ be one of its answer sets. By this construction, the sequence $\langle Y_0, Y_1, Y_2 \rangle$ is a solution to $P$ with respect to $V$. By the Splitting Sequence Theorem, $Y = Y_0 \cup Y_1 \cup Y_2$ is an answer set for $P$. Since $a \in Cn(P)$, $a \in Y$. It follows that $a \in Y_1$. And since $Y_1$ was an arbitrarily chosen answer set for (3), we conclude that $a$ is among its consequences. □

## Acknowledgements

Thanks to T.C. Son for hinting at this question, and to Michael Gelfond for reminding me of a related conjecture in (Baral and Gelfond 1994). Thanks also to Son, Michael, and Vladimir Lifschitz for comments on an earlier draft, and to Jürgen Dix and John Schlipf for helpful correspondence.

## References

Baral, C. and Gelfond, M. (1994). Logic programming and knowledge representation, *Journal of logic programming* **19,20**: 73–148.

Dix, J. (1991). Classifying semantics of logic programs, *in* A. Nerode, W. Marek and V. S. Subrahmanian (eds), *Logic Programming and Non-Monotonic Reasoning: Proc. of the First Int'l Workshop*, pp. 166–180.

Dung, P. M. (1992). On the relations between stable and well-founded semantics of logic programs, *Theoretical Computer Science* **105**: 222–238.

Fages, F. (1994). Consistency of Clark's completion and existence of stable models, *Journal of Methods of Logic in Computer Science* **1**(1): 51–60.

Gabbay, D. (1985). Theoretical foundations for non-monotonic reasoning, *in* K. Apt (ed.), *Expert Systems, Logics, and Models of Concurrent Systems*, Springer-Verlag.

Gelfond, M. and Lifschitz, V. (1988). The stable model semantics for logic programming, *in* R. Kowalski and K. Bowen (eds), *Logic Programming: Proc. of the Fifth Int'l Conf. and Symp.*, pp. 1070–1080.

Gelfond, M. and Lifschitz, V. (1991). Classical negation in logic programs and disjunctive databases, *New Generation Computing* **9**: 365–385.

Kraus, S., Lehmann, D. and Magidor, M. (1990). Nonmonotonic reasoning, preferential models and cumulative inference, *Artificial Intelligence* **44**(1): 167–207.

Kunen, K. (1989). Signed data dependencies in logic programs, *Journal of Logic Programming* **7**(3): 231–245.

Lifschitz, V. and Turner, H. (1994). Splitting a logic program, *in* P. Van Hentenryck (ed.), *Proc. of Eleventh Int'l Conf. on Logic Programming*, pp. 23–37.

Makinson, D. (1989). General theory of cumulative inference, *in* M. Reinfrank, J. de Kleer, M. Ginsberg and E. Sandewall (eds), *Non-Monotonic Reasoning: 2nd International Workshop (Lecture Notes in Artificial Intelligence 346)*, Springer-Verlag, pp. 1–17.

Makinson, D. (1993). General patterns in nonmonotonic reasoning, *in* D. Gabbay, C. Hogger and J. Robinson (eds), *The Handbook of Logic in AI and Logic Programming*, Vol. 3, Oxford University Press, pp. 35–110.

Sato, T. (1990). Completed logic programs and their consistency, *Journal of Logic Programming* **9**: 33–44.

Van Gelder, A., Ross, K. and Schlipf, J. (1991). The well-founded semantics for general logic programs, *Journal of the ACM* **38**(2): 620–650.