Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

A Characterization of Stable Models using a Non-Monotonic Operator

F. Teusink

Computer Science/Department of Software Technology

CS-R9315 1993

A Characterization of Stable Models using a Non-Monotonic Operator

Frank Teusink
CWI
P.O. Box 4079, 1009 AB Amsterdam
The Netherlands
frankt@cwi.nl

Abstract

Stable models seem to be a natural way to describe the beliefs of a rational agent. However, the definition of stable models itself is not constructive. It is therefore interesting to find a constructive characterization of stable models, using a fixpoint construction. The operator we define, is based on the work of –among others– F. Fages. For this operator, every total stable model of a general logic program will coincide with the limit of some (infinite) sequence of interpretations generated by it. Moreover, the set of all stable models will coincide with certain interpretations in these sequences. Furthermore, we will characterize the least fixpoint of the Fitting operator and the well-founded model, using our operator.

1991 Mathematical Subject Classification: 68N17
1991 CR Classification: D.1.6, F.4.1, I.2.3
Keywords and Phrases: Logic Programming, Stable Models, Non-Monotonic Reasoning
Note: This Technical Report is an extended version of [Teu93]

1 Introduction

Stable models, as introduced in [GL88] and extended to three-valued models in [Prz90b], seem to be a natural candidate for providing general logic programs with a meaning. However, their definition is not constructive. The aim of this paper is to find a constructive characterization of stable models for general logic programs, using sequences of interpretations generated by iterating a non-deterministic non-monotonic operator. The non-deterministic behaviour of this operator is captured by using the notion of selection strategies. Our operator is based on the ideas of F. Fages [Fag91]. The main difference with the approach of Fages is, that our operator is less non-deterministic than his. As a result, our operator is more complex, but this enables us to define a notion of (transfinite) fairness with which we can characterize a class of stabilizing strategies that contain all total stable models. Moreover, the additional structure in our operator allows us to define various classes of strategies with nice properties. The difference of our operator with respect to the *backtracking fixpoint* introduced by D. Saccà and C. Zaniolo in [SZ90] is twofold: we find all stable models, instead of only all total stable models, and, if an inconsistency occurs, we use a non-deterministic choice over all possibilities for resolving that inconsistency, while their operator uses backtracking, which is just one particular possibility.

In the next section we give a short introduction on general logic programs and interpretations, and introduce some notations that will be used throughout the paper. Section 3 contains an explanation of (three-valued) well-supported models and stable models, and a generalization of Fages' Lemma, which establishes the equivalence between a subset of the set of (three-valued) well-supported models and the set of (three-valued) stable models. In section 4 we will introduce our operator $\mathcal{S}_P$, and prove that the sequences generated by this operator consist of well-supported interpretations. After this, we will show in sections 5, 6, 7 and 8 how to find total stable models, (three-valued) stable models, the least fixpoint of the Fitting operator and the well-founded model, respectively, using our operator. In section 9, we will take a short look at the complexity of the operator, and effective strategies for finding stable models.

2 Preliminaries and notations

A *general logic program* is a finite set of clauses $R : A \leftarrow L_1 \wedge \ldots \wedge L_k$, where A is an atom and L_i ($i \in [1..k]$) is a literal. A is called the *conclusion* of R, and $\{L_1, \ldots, L_k\}$ is called the *set of premises* of R. We write $concl(R)$ and $prem(R)$ to denote A and $\{L_1, \ldots, L_k\}$, respectively. For semantic purposes, a general logic program is equivalent to the (possibly infinite) set of ground instances of its clauses. In the following, we will only work with these infinite sets of ground clauses, and call them *programs*.

We use $\mathcal{B}_P$ to denote the Herbrand Base of a program P; A, A' and A_i represent typical elements of $\mathcal{B}_P$. Furthermore, $\mathcal{L}_P$ is the set of all literals of P; L, L' and L_i represent typical elements of $\mathcal{L}_P$. We use the following notations:

- for a literal L, $\neg L$ is the positive literal A, if $L = \neg A$, and the negative literal $\neg A$, if $L = A$, and
- for a set of literals S, we write
  - $\neg S$ to denote the set $\{\neg L \mid L \in S\}$,
  - $S^+ = \{A \mid A \in S\}$ to denote the set of all atoms that appear in positive literals of S,
  - $S^- = \{A \mid \neg A \in S\}$ to denote the set of all atoms that appear in negative literals of S, and
  - $S^\pm = S^+ \cup S^-$ to denote the set of all atoms that appear in literals of S.

A *two-valued* interpretation of a program P maps the elements of $\mathcal{B}_P$ on *true* or *false*. In this paper, we will use *three-valued* interpretations, in which an atom can also be mapped on *unknown*. They are defined as follows:

Definition 2.1 Let P be a program. An *interpretation* I of P is a set of elements from $\mathcal{L}_P$. An atom is *true* in I, if it is an element of I^+, it is *false* in I, if it is an element of I^-, and it is *unknown* in I, if it is not an element of $I^\pm$. If some atom is both *true* and *false* in I, then I is called *inconsistent*. If all atoms in $\mathcal{B}_P$ are either *true* or *false* (or both) in I, then I is called *total*. $\square$

Example 2.2 Consider program P_1 consisting of the clauses $p(a) \leftarrow \neg p(b)$, $p(b) \leftarrow \neg p(a)$ and $q(b) \leftarrow q(b)$. We have that $\mathcal{B}_{P_1}$ is the set $\{p(a), p(b), q(a), q(b)\}$. There are $2^8 = 256$ interpretations of P_1, $3^4 = 81$ of them are consistent, $3^4 = 81$ of them are total, and $2^4 = 16$ of them are consistent and total. $\circ$

Note, that a consistent total interpretation can be seen as a two-valued interpretation, because then no atom is both *true* and *false* and, because $I^\pm = \mathcal{B}_P$, no atom is unknown.

3 Well-Supported and Stable Models

In this section we will introduce well-supported models and stable models. Our definition of well-supported models is an extension (to three-valued models) of the definition given in [Fag91]. Our definition of three-valued stable models follows the definition given in [Prz90b]. First, we will introduce *well-supported models*, because they follow quite naturally from the intuitive idea of the meaning of a program. After this we will give the definition of *stable models*, which is quite elegant. In the remainder of this section we generalize of Fages' Lemma [Fag91], which states that the class of total stable models and the class of total well-supported models coincide, to three-valued models.

So, let's take a look at the intuitive idea of the meaning of a program. First of all, an interpretation should be consistent; it doesn't make sense to have atoms that are both true and false. Furthermore, one can see a clause in a program as a statement saying that the conclusion of that clause should be true if that clause is applicable.

Definition 3.1 Let P be a program, let I be an interpretation of P and let R be a clause in P. R is *applicable* in I, if $prem(R) \subseteq I$. R is *inapplicable* in I, if $\neg prem(R) \cap I \neq \emptyset$. We call $\neg prem(R) \cap I$ the *blocking-set* of R in I. □

Now, a *model* of a program P is a consistent interpretation I of P such that, for every clause in P that is applicable in I, the conclusion of that clause is true in I, and an atom is false in I only if all clauses with that atom as conclusion are inapplicable in I. Note, that we have to state explicitly that I has to be consistent, because in our definition an interpretation can be inconsistent.

In a model of P, atoms can be true, even if there is no reason for that atom being true. However, an atom should only be true, if there is some kind of "explanation" for the fact that that atom is true. This concept of "explanation" will be formalized using the notion of *support order*.

Definition 3.2 Let P be a program and let I be an interpretation of P. A partial order $<$ on the elements of $\mathcal{L}_P$ is a *support order* on I, if, for all $A \in I^+$, there exists a clause R in P with conclusion A such that R is applicable in I and, for all $A' \in prem(R)^+$, $A' < A$. □

If, for some positive literal L that is true in M, we gather all literals L' such that $L' <^* L$ ($<^*$ is the transitive closure of $<$), then this set constitutes some kind of explanation for the fact that L is true in M.

Example 3.3 Consider program P_2 consisting of the clauses $p \leftarrow q \wedge r$, $q \leftarrow$ and $r \leftarrow \neg s$. One of the models of P_2 is $\{p, q, r, \neg s\}$, and $\{q < p, r < p\}$ is a support order on this model. We can read this support order as follows: p is true because r and q are true, q is always true, r is true because s is false, and s is false because there is no reason why s should be true. ∘

However, such an explanation can be rather awkward, either because it refers to the conclusion itself, or because it contains an infinite number of literals.

Example 3.4 Consider program P_3 consisting of the clauses $p \leftarrow q$ and $q \leftarrow p$. One of the models of P_3 is $\{p, q\}$, and $\{p < q, q < p\}$ is a support order on this model. However, the explanation 'p is true because q is true and q is true because p is true', is not a meaningful explanation for the fact that p is true. ∘

Example 3.5 Consider program P_4 consisting of the clauses $p(x) \leftarrow p(s(x))$ and $p(0) \leftarrow$. One of the models of P_4 is $\{p(s^i(0)) \mid i \geq 0\}$, and the partial order $\{p(s^{i+1}(0)) < p(s^i(0)) \mid i \geq 0\}$ is a support order on this model. However, any explanation for the fact that $p(0)$ is true in M_4, would be infinite. This seems to be rather counterintuitive. $\circ$

Models for which every support order contains these cyclic or infinite explanations, should not be considered as giving a correct meaning to a program. This can be achieved by using the fact that a support order is well-founded if and only if it doesn't contain cyclic or infinite explanations. Now, we can give the definition of *well-supported models*.

Definition 3.6 Let P be a program, and let M be a model of P. M is a *well-supported model* of P, if there exists a well-founded support order on M. $\square$

Example 3.7 Consider the program P_1 (example 2.2). The interpretations $\{p(a), \neg p(b), \neg q(a), \neg q(b)\}$ and $\{p(a), \neg p(b), \neg q(a), \neg q(b)\}$ are well-supported models of P_1. $\circ$

Another characterization of the meaning of a program is given by the definition of *stable models*. In the two-valued case, this definition uses the fact that the meaning of positive logic programs (in which the bodies of the clauses contain only positive literals) is well understood; it is given by the unique *two-valued minimal model* of the program. This definition of stable models has been generalized by T. Przymusinski to three-valued stable models [Prz90b]. In this definition, he uses the notion of (three-valued) truth-minimal models, and a program transformation.

Definition 3.8 Let P be a positive program and let M be a model of P. M is a *truth-minimal model* of P, if there does not exist a model M' (other than M) of P such that $M'^+ \subseteq M^+$ and $M'^- \supseteq M^-$. $\square$

Definition 3.9 Let P be a program and let I be an interpretation of P. The program $\frac{P}{I}$ is obtained from P by replacing every negative literal L in the body of a clause in P that is true (resp. false; resp. unknown) in I by the proposition $\mathbf{t}$ (resp. $\mathbf{f}$; resp. $\mathbf{u}$). $\square$

Now, we are able to give the definition of a *stable model*.

Definition 3.10 Let P be a program and let M be an interpretation of P. M is a *stable model* of P, if M is a truth-minimal model of $\frac{P}{M}$. $\square$

Example 3.11 Consider the program P_1 (example 2.2), and the model $\{p(a), \neg p(b), \neg q(a), \neg q(b)\}$ of P_1. M is a stable model of P_1, because it is a truth-minimal model of the program $\frac{P_1}{M} = \{p(a) \leftarrow \mathbf{t},\ p(b) \leftarrow \mathbf{f},\ q(b) \leftarrow q(b)\}$. $\circ$

The following lemma shows that the class of stable models coincides with a subclass of the well-supported models. This lemma is an generalization of the lemma by F. Fages [Fag91], which proves that two-valued stable models and two-valued well-supported models coincide. The proof we give, resembles the proof given by F. Fages. First, we have to introduce the notion of *(greatest) unfounded set*.

Definition 3.12 Let P be a program and let I be an interpretation of P. Let S be a subset of $\mathcal{B}_P - I^{\pm}$. S is an *unfounded set* of I, if all clauses R in P such that $concl(R) \in S$ are inapplicable in $I \cup \neg S$. The *greatest unfounded set* $U_P(\mathrm{I})$ of I is the union of all unfounded sets of I. $\square$

Note, that our definition of unfounded set differs from the definition used in [GRS91]. However, we can define their operator as follows: $\mathbf{U}_P(I) = U_P(I) \cup I^-$.

Lemma 3.13 (Equivalence) *Let P be a program and let M be an interpretation of P. M is a stable model of P iff M is a well-supported model of P such that $U_P(M) = \emptyset$.*

Proof: By definition, M is a stable model of P iff M is a truth-minimal model of $\frac{P}{M}$. By theorem 3.2 in [Prz90a] (page 451), the truth-minimal model can be characterized using the Fitting operator (see definition 7.1); M is the truth-minimal model of $\frac{P}{M}$ iff $M = \Psi_{\frac{P}{M}} \uparrow^{\omega} (\emptyset)$. We will write Ψ_α as a shorthand for $\Psi_{\frac{P}{M}} \uparrow^{\alpha} (\emptyset)$.

($\Rightarrow$) Let M be a well-supported model of P such that $U_P(M)$ is empty, and let $<_M$ be a well-founded support order on M. To prove that M is a stable model of P, it suffices to prove that $M = \Psi_\omega$.

1. We prove that $M^+ \subseteq \Psi_\omega^+$. In order to do this, we prove by induction on $<_M$ that $A \in M^+$ implies $A \in \Psi_\omega^+$. If A is a $<_M$-minimal element of M^+, then there exists a clause R in P with conclusion A that is applicable in M such that $prem(R)^+$ is empty. But then there exists a clause R' in $\frac{P}{M}$ with conclusion A that is applicable in M such that $prem(R)$ contains only propositional constants $\mathbf{t}$, and therefore by definition of Ψ, $A \in \Psi_\omega$. Assume that, for all $A' <_M A$, $A' \in M^+$ implies $A' \in \Psi_\omega$. Because $A \in M^+$, there exists a clause R in P with conclusion A that is applicable in M such that, for all $A' \in prem(R)^+$, $A' <_M A$. But then there exists a clause R' in $\frac{P}{M}$ with conclusion A that is applicable in M such that $A' \in prem(R)$ implies that A' is the propositional constant $\mathbf{t}$ or $A' <_M A$. By induction hypothesis, we have that $A' <_M A$ implies that $A' \in \Psi_\omega$. Therefore, R' is applicable in Ψ_ω and thus, by definition of Ψ_ω, $A \in \Psi_\omega$.

2. We prove by induction on α that $\Psi_\alpha^+ \subseteq M^+$. For $\alpha = 0$, the lemma holds trivially. Assume that $\Psi_\alpha^+ \subseteq M^+$. Suppose that $A \in \Psi_{\alpha+1}^+$. Then, there exists a clause R in $\frac{P}{M}$ with conclusion A that is applicable in Ψ_α. But then all elements of $prem(R)$ (excluding the propositional constant $\mathbf{t}$) are in Ψ_α and therefore in M. But then, there exists a corresponding clause R' in P with conclusion A that is applicable in M. But M is a model of P and therefore $A \in M$.

3. We prove that $M^- \subseteq \Psi_\omega^-$. Let $A in M^-$. Because M is a model of P, every clause R in P with conclusion A is inapplicable in M. But then, every clause R' in $\frac{P}{M}$ with conclusion A is inapplicable in M. But $\frac{P}{M}$ is a positive program, and therefore these clauses are also inapplicable in M^+. Also, we already have that $M^+ = \Psi_\omega^+$. So, because $\frac{P}{M}$ is positive, every clause R' in $\frac{P}{M}$ with conclusion A is inapplicable in Ψ_ω. Therefore, by definition of Ψ_ω, $A \in \Psi_\omega^-$.

4. We prove that $\Psi_\omega^- \subseteq M^-$. We already have that $M^- \subseteq \Psi_\omega^-$. Suppose that $S = \Psi_\omega^- - M^-$ is non-empty. Ψ_ω is a model of $\frac{P}{M}$. Therefore, for every $A \in S$, every clause R in $\frac{P}{M}$ with conclusion A is inapplicable in Ψ_ω. Because $M^+ = \Psi_\omega^+$, we know that $S \cap M^+ = \emptyset$. We also have that $M \cup \neg S = \Psi_\omega$. Therefore, for every $A \in S$, every clause R in $\frac{P}{M}$ with conclusion A is inapplicable in $M \cup \neg S$. But then by construction of $\frac{P}{M}$, for every $A \in S$, every clause R' in P with conclusion A is inapplicable in $M \cup \neg S$. So, S is an unfounded set of M. This is in contradiction with the fact that $U_P(M)$ is empty. Therefore S has to be empty.

($\Leftarrow$) Let $M = \Psi_\omega$. We have to prove that M is a well-supported model of P such that $U_P(M)$ is empty.

- We prove that $U_P(M)$ is empty. Suppose that $U_P(M)$ is non-empty. Consider the interpretation $M' = M \cup \neg U_P(M)$. Clearly, M' is smaller than M in the truth-ordering. But M' is also a model of P and $\frac{P}{M}$. This is in contradiction with the fact that $M = \Psi_\omega$ and that Ψ_ω is a truth-minimal model of $\frac{P}{M}$.

- We prove that there exists a well-founded support-order on M. We assign a *rank* to the elements of M^+: the rank $r(A)$ of an atom $A \in M^+$ is the least ordinal α such that $A \in \Psi_\alpha$. This rank is defined on all elements of M^+, because $M = \Psi_\omega$. We show that the partial ordering $<_r$ such that $A' <_r A$ iff $r(A') < r(A)$ is a well-founded support order on M. Clearly, $<_r$ is well-founded. Let A be an arbitrary element of M^+. We know that $A \in \Psi_{r(A)}$ iff there exists a clause R in $\frac{P}{M}$ that is applicable in $\Psi_{r(A)-1}$. But then, for all $A' \in \mathit{prem}(R)$, $r(A') < r(A)$ and therefore $A' <_r A$. By the construction of $\frac{P}{M}$ and the fact that M is a stable model of P, we have that there exists a clause R' in P with conclusion A that is applicable in M, such that, for all $A' \in \mathit{prem}(R)^+$, $A' <_r A$. Thus, $<_r$ is a well-founded support order on M.

$\square$

4 The operator $\mathcal{S}_P$

In this section, we define the operator $\mathcal{S}_P$. This operator is inspired on the operator J_P^ρ of Fages, but there are some major differences.

The idea is, to generate all *total* stable models of a program, by starting from the empty interpretation. At each step, we try to extend an interpretation I to a new interpretation I', that brings us "nearer" to a total stable model. For this, we use the following strategies:

1. If there exists a clause R that is applicable in I and $\mathit{concl}(R)$ is not an element of I, then we add $\mathit{concl}(R)$ to I (after all, we are looking for a model).

2. If there exists an atom A such that all clauses R that have A as conclusion, are inapplicable in I, and $\neg A$ is not an element of I, then we add $\neg A$ to I (after all, we are working towards a total interpretation).

3. If the previous two strategies fail, we can do little more that blindly select an atom from $\mathcal{B}_P - I^\pm$, and add it, or its negation, to I. However, in contrast with the two previous strategies, this strategy is flawed, in the sense that, even if I is a subset of some stable model, I' is not guaranteed to be a subset of a stable model. In fact, continuing the procedure with I' can lead to an inconsistent interpretation.

4. If I is inconsistent, then we should try to find a consistent interpretation I'. However, we do not want to throw away I completely. We know that the inconsistency was caused by some literal chosen by strategy 3. We will maintain "possible reasons for inconsistency" with our interpretation, in order to identify a literal in I that could be the reason for the inconsistency, and find a new consistent interpretation I' by removing from I all literals that were added to the interpretation due to the presence of this literal.

Note, that with all four strategies one could have more than one way to generate the next interpretation. For example, if there are two reasons for the inconsistency of an interpretation, there are two possibilities for resolving that inconsistency. As a result, our operator will be non-deterministic.

We have to maintain "reasons for inconsistency" with our interpretation. Moreover, we will maintain a support order with our interpretation, to help us prove various properties. This leads to the following definition of *j-interpretations*.

Definition 4.1 A *j-triple*, is a triple $\langle L, \tau, \psi \rangle$, such that L is an element of $\mathcal{L}_P$, and τ and ψ are subsets of $\mathcal{L}_P$. A *j-interpretation* J of P is a set of j-triples such that for every literal in $\mathcal{L}_P$, J contains at most one j-triple with that literal as the first element. We call τ the *support-set* of L and ψ the *culprit-set* of L. For a set S of j-triples, we will write $\overline{S}$ to denote the set of literals $\{L \mid \langle L, \tau, \psi \rangle \in S\}$. □

Note, that our support-set differs from the justification in a justified atom of Fages, because it can be infinite, and it is defined on literals instead of atoms. Moreover, our support-set is intended to contain a set of premises for a positive literal, and a set of elements of blocking-sets for negative literals, whereas the justifications of Fages contain a complete explanation for the fact that an atom is true. Using the support-sets in a j-interpretation J, we can define a partial order on the literals in $\overline{J}$.

Definition 4.2 Let J be a j-interpretation. We define $<_J$ to be the partial order such that $A' <_J A$ iff $\langle A, \tau, \psi \rangle \in J$ and $A' \in \tau^+$ (note, that A is a positive literal). □

In the interpretations on which $\mathcal{S}_P$ will operate, the culprit-set will contain the "possible reasons for inconsistency" and the partial order $<_J$ will be a support order on $\overline{J}$.

In the definition of the operator $\mathcal{S}_P$, we will use the *conflict-set*, *choice-set* and *culprit-set* of a j-interpretation J. The *conflict-set* of a j-interpretation J contains j-triples for every literal L for which there are one or more reasons for adding them to J, according to strategies 1 and 2.

Definition 4.3 Let P be a program and let J be a j-interpretation of P. The *conflict-set* $Conflict_P(J)$ of J is the set of j-triples $\langle L, \tau, \psi \rangle$ such that

- $L \notin \overline{J}$,
- if $L = A$, then there exists a clause R in P with conclusion A that is applicable in $\overline{J}$ such that $\tau = prem(R)$,
- if $L = \neg A$, then every clause R in P with conclusion A is inapplicable in $\overline{J}$, and for every clause R in P with conclusion A exists a literal L_R in the blocking-set of R in $\overline{J}$ such that $\tau = \{L_R \mid R \in P \wedge concl(R) = A\}$, and
- $\psi = \bigcup\{\psi' \mid \langle L', \tau', \psi' \rangle \in J \wedge L' \in \tau\}$.

□

For a j-triple $\langle L, \tau, \psi \rangle$ in $Conflict_P(J)$, τ contains the reason for adding L to J, and ψ contains all literals that could be the cause of L being an element of $\overline{Conflict_P(J)}$, while $\neg L$ is an element of $\overline{J}$.

The *choice-set* of J contains j-triples that could be added to J on behalf of strategy 3. The support-sets and choice-sets of these j-triples reflect the fact that there is no real support for adding these literals to J.

Definition 4.4 Let P be a program and let J be a j-interpretation of P. The *choice-set* $Choice_P(J)$ of P is the set

$$\{\langle L, \emptyset, \{L\} \rangle \mid L \in \neg(\mathcal{B}_P - \overline{J}^{\pm})\}$$

□

The *culprit-set* of an inconsistent j-interpretation J, is the set of all "possible reasons for inconsistency"; that is, the set of literal that are common to the culprit-sets of all literals L in $\overline{J}$ whose negation $\neg L$ is also an element of $\overline{J}$.

Definition 4.5 Let P be a program and let J be a j-interpretation of P. The *culprit-set* $Culprit_P(J)$ of J is the set

$$\bigcap\{\psi \cup \psi' \mid \langle A, \tau, \psi\rangle \in J \wedge \langle \neg A, \tau', \psi'\rangle \in J\}$$

□

Note, that if $\overline{J}$ is consistent then $Culprit_P(J) = \emptyset$. We are now capable of defining our operator $\mathcal{S}_P$.

Definition 4.6 For a general logic program P, we define the operator $\mathcal{S}_P$ as follows:

$$\mathcal{S}_P(J) = \begin{cases} J - \{\langle L, \tau, \psi\rangle \mid \rho_1 \in \psi\} & \text{, if } Culprit_P(J) \neq \emptyset \\ J \cup \{\rho_2\} & \text{, if } Conflict_P(J) \neq \emptyset \\ J \cup \{\rho_3\} & \text{, if } Choice_P(J) \neq \emptyset \\ J & \text{, otherwise} \end{cases}$$

$$\text{where} \quad \begin{array}{l} \rho_1 \in Culprit_P(J) \\ \rho_2 \in Conflict_P(J) \\ \rho_3 \in Choice_P(J) \end{array}$$

□

Note, that in this definition the order of the conditions is relevant (i.e. a rule is only applied if its condition is satisfied *and* the conditions of all previous rules failed).

The operator as we defined it, is non-deterministic, in the sense that it non-deterministically chooses an element (ρ_1, ρ_2 or ρ_3) from a set of candidates. Because we want to manipulate this non-deterministic behaviour, we extend the operator with a *selection strategy*, that encapsulates this non-deterministic behaviour of $\mathcal{S}_P$.

Definition 4.7 Let P be a program. A *selection strategy* ρ for P is a non-deterministic function that, for a j-interpretation J of P, chooses ρ_1 among $Culprit_P(J)$, ρ_2 among $Conflict_P(J)$ and ρ_3 among $Choice_P(J)$. □

Note, that ρ can be deterministic if we consider more information. For instance, we could use a selection strategy that bases its choices for some j-interpretation J on the way in which J was generated (i.e. previous applications of $\mathcal{S}_P$). We will use the notation $\mathcal{S}_P^\rho$ to indicate that we are using the operator on a program P with a selection strategy ρ for P.

As said before, we want to find a stable model of P by starting from the empty interpretation. In order to do this, we have to define the (ordinal) powers of $\mathcal{S}_P^\rho$.

Definition 4.8 Let P be a program and let ρ be a selection strategy for P. Let $\mathcal{S}_P^\rho$ be the operator as defined. We define the powers of $\mathcal{S}_P^\rho$ inductively:

$$\mathcal{S}_P^\rho \uparrow^\alpha = \begin{cases} \emptyset & \text{, if } \alpha = 0 \\ \mathcal{S}_P^\rho(\mathcal{S}_P^\rho \uparrow^{\alpha-1}) & \text{, if } \alpha \text{ is a successor ordinal} \\ \bigcup_{\beta<\alpha} \bigcap_{\beta \leq \gamma < \alpha} \mathcal{S}_P^\rho \uparrow^\gamma & \text{, if } \alpha \text{ is a limit ordinal} \end{cases}$$

□

The definition for zero and successor ordinals are standard. The definition for limit ordinal is the same as the one used by Fages; it states that at a limit ordinal α, we retain only the j-triples that where persistent in the preceding sequence of j-interpretations; that is, for every j-triple in $\mathcal{S}_P^\rho \uparrow^\alpha$, there exists an ordinal β smaller that α, such that, for all $\gamma \in [\beta..\alpha)$, this j-triple is an element of $\mathcal{S}_P^\rho \uparrow^\gamma$.

Using the powers of $\mathcal{S}_P^\rho$, we define the following infinite sequence of j-interpretations.

Definition 4.9 Let P be a program and let ρ be a selection strategy for P. The *sequence for P and ρ* is the infinite sequence of j-interpretations $\Gamma_P^\rho \equiv J_0, \ldots, J_\alpha, \ldots$, where $J_\alpha = \mathcal{S}_P^\rho \uparrow^\alpha$, for all ordinals α. □

We will now work towards a proof of the fact that certain fixpoints of $\mathcal{S}_P$ are stable models of P. First, we have to prove that the application of $\mathcal{S}_P$ on a j-interpretation results in a j-interpretation, and that every element of a sequence is a j-interpretation.

Lemma 4.10 *Let P be a program and let ρ be selection strategy for P. If J is a j-interpretation, then $\mathcal{S}_P^\rho(J)$ is a j-interpretation.*

Proof: Suppose J is a j-interpretation. Then, we can obtain $\mathcal{S}_P^\rho(J)$ from J in two different ways:

- By adding a j-triple $\langle L, \tau, \psi\rangle$ to J. By definition of $\mathcal{S}_P$ (conflict-set and choice-set), we know that $L \notin \overline{J}$. From this it follows that $\mathcal{S}_P^\rho(J) = J \cup \{\langle L, \tau, \psi\rangle\}$ is a j-interpretation of P.

- By removing elements from J. Because any subset of a j-interpretation is itself a j-interpretation, we have that $\mathcal{S}_P^\rho(J)$ is a j-interpretation.

□

Lemma 4.11 *Let Γ_P^ρ be a sequence for a program P. Every element J_α of Γ_P^ρ is a j-interpretation of P.*

Proof: For $J_0 = \emptyset$, the lemma is trivially true. Assume that for all $\beta < \alpha$, J_β is a j-interpretation of P.

If α is a successor ordinal, $J_{\alpha-1}$ is a j-interpretation by induction hypothesis, and therefore, by lemma 4.10, J_α is a j-interpretation.

If α is a limit ordinal, we know that it is a set of j-triples, because it is a subset of a union of j-interpretations. Furthermore, we have that if $\langle L, \tau, \psi\rangle \in J_\alpha$, then for some β such that $\beta < \alpha$ we have that, for all $\gamma \in [\beta..\alpha)$, $\langle L, \tau, \psi\rangle \in J_\gamma$. By induction hypothesis, for all $\gamma \in [\beta..\alpha)$, J_γ is a j-interpretation and therefore there is no j-triple other than $\langle L, \tau, \psi\rangle$ in J_γ with L on the first position. But then we have that there is no j-triple, other than $\langle L, \tau, \psi\rangle$, in J_α with L on the first position. Therefore, J_α is a j-interpretation. □

We will now prove that for every j-interpretation J_α in a sequence Γ_P^ρ, the partial order $<_{J_\alpha}$ is a support order and a well-founded order. First, we have to prove the following auxiliary lemma.

Lemma 4.12 *Let Γ_P^ρ be a sequence. For all J_α in Γ_P^ρ, for all $\langle L, \tau, \psi\rangle \in J_\alpha$ and for all $L' \in \tau$, there exist a τ' and a $\psi' \subseteq \psi$ such that $\langle L', \tau', \psi'\rangle \in J_\alpha$.*

Proof: For $J_0 = \emptyset$, the lemma is trivially true. Assume that, for all β smaller than α, we have that $\langle L, \tau, \psi \rangle \in J_\beta$ implies that, for all $L' \in \tau$, there exist a τ' and a $\psi' \subseteq \psi$ such that $\langle L', \tau', \psi' \rangle \in J_\beta$.

If α is a successor ordinal, J_α can be obtained from $J_{\alpha-1}$ in two ways:

- By adding a j-triple $\langle L, \tau, \psi \rangle$ to $J_{\alpha-1}$. Here, the lemma follows directly from the definition of $\mathcal{S}_P$ (conflict-set and choice-set).

- By removing elements from $J_{\alpha-1}$. Suppose $\langle L, \tau, \psi \rangle \in J_{\alpha-1}$ and $L' \in \tau$. Then, we have by induction hypothesis that there exist a τ' and a $\psi' \subseteq \psi$ such that $\langle L', \tau', \psi' \rangle \in J_{\alpha-1}$. Because $\psi' \subseteq \psi$, we have by the definition of $\mathcal{S}_P$ that $L' \notin J_\alpha$ implies that $L \notin J_\alpha$.

If α is a limit ordinal, we have that if $\langle L, \tau, \psi \rangle \in J_\alpha$, then for some β such that $\beta < \alpha$ we have that, for all $\gamma \in [\beta..\alpha)$, $\langle L, \tau, \psi \rangle \in J_\gamma$. By induction hypothesis, we have that, for all $\gamma \in [\beta..\alpha)$ and for all $L' \in \tau$, there exist a τ' and a $\psi' \subseteq \psi$ such that $\langle L', \tau', \psi' \rangle \in J_\gamma$. Also, we have that if $\langle L', \tau', \psi' \rangle \in J_\gamma$ and $\langle L', \tau'', \psi'' \rangle \in J_{\gamma+1}$, then $\tau' = \tau''$ and $\psi' = \psi''$. Therefore, $\langle L', \tau', \psi' \rangle \in J_\alpha$. $\square$

Theorem 4.13 (Supportedness) *Let Γ_P^ρ be a sequence for a program P. For every J_α in Γ_P^ρ, the partial order $<_{J_\alpha}$ is a support order on $\overline{J}_\alpha$.*

Proof: We have to prove that for all $A \in \overline{J}_\alpha^+$ there exists a applicable clause R in P with conclusion A such that for all $A' \in \mathit{prem}(R)^+$, $A' <_{J_\alpha} A$.

We will proceed by induction on α. For $J_0 = \emptyset$, the claim holds trivially. Assume that for all β smaller than α and for all $A \in \overline{J}_\beta^+$ there exists a applicable clause R in P with conclusion A such that for all $A' \in \mathit{prem}(R)^+$, $A' <_{J_\beta} A$.

If α is a successor ordinal, then J_α can be obtained from $J_{\alpha-1}$ in two ways:

1. By adding a j-triple $\langle L, \tau, \psi \rangle$ to $J_{\alpha-1}$. If L is a negative literal, then $<_{J_\alpha} \equiv <_{J_{\alpha-1}}$ and the claim follows from the induction hypothesis and the fact that $J_\alpha \supseteq J_{\alpha-1}$. If L is positive, we have by the definition of $\mathcal{S}_P$ (conflict-set) that there exists a applicable clause R in P with conclusion L such that $\tau = \mathit{prem}(R)$. Therefore, $A \in \mathit{prem}(R)^+$ implies that $A \in \tau^+$, which, by definition of $<_{J_\alpha}$, implies that $A <_{J_\alpha} L$.

2. By removing a set of j-triples from $J_{\alpha-1}$. The claim follows from lemma 4.12 and the fact that $<_{J_{\alpha-1}}$ is a support order on $J_{\alpha-1}$.

If α is a limit ordinal, then $A \in \overline{J}_\alpha^+$ implies that there exists an β such that $\beta < \alpha$ and for some τ and ψ, for all $\gamma \in [\beta..\alpha)$, $\langle A, \tau, \psi \rangle \in J_\gamma$. By induction hypothesis we have that there exists a applicable clause $R \in P$ with conclusion A such that for all $A' \in \mathit{prem}(R)^+$, $A' <_{J_\gamma} A$ and therefore that $A' \in \tau$. By lemma 4.12 we have that there exist τ' and ψ' such that $\langle A', \tau', \psi' \rangle \in J_\gamma$ and therefore $\langle A', \tau', \psi' \rangle \in J_\alpha$. From this we can conclude that if $A' \in \mathit{prem}(R)^+$ then $A' <_{J_\alpha} A$. $\square$

Theorem 4.14 (Well-Foundedness) *Let Γ_P^ρ be a sequence for a program P. For every J_α in Γ_P^ρ, the partial order $<_{J_\alpha}$ is well-founded.*

Proof: Suppose that $<_{J_\alpha}$ is not well-founded. Then, there exists an infinite decreasing chain $\ldots <_{J_\alpha} A_2 <_{J_\alpha} A_1 <_{J_\alpha} A_0$. Because $A_i \in \overline{J}_\alpha^+$, there exists a least ordinal β_i such that $\beta_i \leq \alpha$ and for some τ_i and ψ_i, for all $\gamma \in [\beta_i..\alpha]$, $\langle A_i, \tau_i, \psi_i \rangle \in J_\gamma$. Also, because $A_{i-1} \in \overline{J}_\alpha^+$, there exists a least ordinal β_{i-1} such that $\beta_{i-1} \leq \alpha$ and for some τ_{i-1} and ψ_{i-1}, for all $\gamma \in [\beta_{i-1}..\alpha]$, we have that $\langle A_{i-1}, \tau_{i-1}, \psi_{i-1} \rangle \in J_\gamma$. Furthermore, we have that $A_i <_{J_\alpha} A_{i-1}$, which implies that $A_i \in \tau_{i-1}$, and therefore $\beta_i < \beta_{i-1}$. As a result, we have that $\ldots < \beta_2 < \beta_1 < \beta_0$ is an infinite decreasing chain. But the $<$ order on ordinals is well-founded. Thus, the assumption that $<_{J_\alpha}$ is not well-founded is in contradiction with the fact that the $<$ order on ordinals is well-founded. Therefore, we can conclude that $<_{J_\alpha}$ is well-founded. $\square$

We will now show that all fixpoints of $\mathcal{S}_P$ that appear in sequences are consistent. In order to prove this, we need a few auxiliary lemmas.

Lemma 4.15 *Let Γ_P^ρ be a sequence for a program P. Let α be the least ordinal such that $Conflict_P(J_\alpha) = \emptyset$. Then, for all $\beta \in [0..\alpha]$, $\overline{J}_\beta$ is consistent.*

Proof: We prove the lemma with induction on β. For $\beta = 0$, we have that $J_\beta = \emptyset$, which is consistent. Assume that for all γ smaller than β, $\overline{J}_\gamma$ is consistent.

Suppose that β is a successor ordinal. If $\beta \notin [0..\alpha]$ or $\overline{J}_\beta$ is consistent, then the claim holds trivially. So, assume that $\beta \in [0..\alpha]$ and that $\overline{J}_\beta$ is inconsistent. Then, we have that $\overline{J}_\beta = \overline{J}_{\beta-1} \cup \{\neg L\}$, where $L \in \overline{J}_{\beta-1}$. First, note that by induction hypothesis, for all γ smaller than β, $\overline{J}_\gamma$ is consistent, and therefore $J_\gamma \subseteq J_{\gamma+1}$. As a result, every clause that is applicable (resp. inapplicable) in $\overline{J}_\gamma$, is applicable (resp. inapplicable) in $\overline{J}_{\gamma+1}$. There are two cases:

1. L is positive. Because $L \in \overline{J}_{\beta-1}$, there has to be at least one clause with conclusion L that is applicable in $J_{\beta-1}$. Also, by induction hypothesis, $J_{\beta-1}$ is consistent. Therefore, there exists at least one clause with conclusion L that is not inapplicable in $J_{\beta-1}$. But then, $\neg L \notin Conflict_P(J_{\beta-1})$. This is in contradiction with the fact that $\beta \in [0..\alpha]$ and $\overline{J}_\beta = \overline{J}_{\beta-1} \cup \{\neg L\}$.

2. L is negative. Because $L \in \overline{J}_{\beta-1}$, all clauses with conclusion $\neg L$ have to be inapplicable in $J_{\beta-1}$. Also, by induction hypothesis, $J_{\beta-1}$ is consistent. Therefore, there does not exists a clause with conclusion $\neg L$ that is applicable in $J_{\beta-1}$. But then, $\neg L \notin Conflict_P(J_{\beta-1})$. This is in contradiction with the fact that $\beta \in [0..\alpha]$ and $\overline{J}_\beta = \overline{J}_{\beta-1} \cup \{\neg L\}$.

Suppose that β is a limit ordinal. Then J_β is consistent, because it is the union of a monotone increasing chain of consistent interpretations. $\square$

Lemma 4.16 *Let Γ_P^ρ be a sequence for a program P. Let α be the least ordinal such that $Conflict_P(J_\alpha) = \emptyset$. For all β greater than α and for all $\langle L, \tau, \psi \rangle \in J_\beta - J_\alpha$, the culprit-set ψ is non-empty.*

Proof: Suppose that for some γ greater than α and some $\langle L, \tau, \psi \rangle \in J_\gamma - J_\alpha$, the culprit-set ψ is empty. Let β be the least ordinal greater than α such that for some $\langle L, \tau, \psi \rangle \in J_\beta - J_\alpha$, ψ is empty. Because ψ is empty, the j-triple can only have been added on behalf of $Conflict_P(J_{\beta-1})$. There are two cases:

1. If L is a positive literal, then ψ is the union of the culprit-sets of the literals in $prem(R)$, where R is a applicable clause with conclusion L. Clearly, $prem(R)$ is non-empty, because otherwise $L \in \overline{J}_\alpha$. But if $prem(R)$ is non-empty and ψ is empty, then the culprit-sets of all the literals in $prem(R)$ have to be empty But then all these literals are elements of J_α, and therefore $L \in \overline{Conflict_P(J_\alpha)}$. This contradicts the fact that $Conflict_P(J_\alpha) = \emptyset$.

2. If L is a negative literal, then ψ is the union of the culprit-sets of a set of literals that block all clauses with conclusion $\neg L$. This set is non-empty, because otherwise $L \in \overline{J}_\alpha$. But if this set is non-empty and ψ is empty, then the culprit-sets of all these literals have to be empty. But then, all these literals are elements of J_α, and therefore $L \in \overline{Conflict_P(J_\alpha)}$. This contradicts the fact that $Conflict_P(J_\alpha) = \emptyset$.

From these contradictions, we have that there cannot exist a least β greater than α such that for some $\langle L, \tau, \psi\rangle \in J_\beta - J_\alpha$, the culprit-set ψ is empty. $\square$

Lemma 4.17 *Let Γ_P^ρ be a sequence for a program P. Let J_α be an element of Γ_P^ρ. If $\overline{J}_\alpha$ is inconsistent, then $\overline{J}_{\alpha+1}$ is consistent.*

Proof: We will prove the lemma by induction on α. The induction base holds trivially: $\overline{J}_0 = \emptyset$ is consistent. Assume that, for all ordinals β smaller than α, $\overline{J}_{\beta+1}$ is consistent if $\overline{J}_\beta$ is inconsistent.

Suppose that α is a successor ordinal and $\overline{J}_\alpha$ is inconsistent. By lemma 4.15 this means that α is greater than γ, where γ is the least ordinal such that $Conflict_P(J_\gamma) = \emptyset$. It is sufficient to prove that $Culprit_P(J_\alpha) \neq \emptyset$, because then is follows from the definition of $\mathcal{S}_P$ that $J_{\alpha+1}$ is consistent. First, observe that there is exactly one atom A such that both $\langle A, \tau, \psi\rangle$ and $\langle \neg A, \tau'\psi'\rangle$ are elements of J_α; at least one, because $\overline{J}_\alpha$ is inconsistent and at most one because by induction hypothesis $\overline{J}_{\alpha-1}$ is consistent. As a result, we have that $Culprit_P(J_\alpha) = \psi \cup \psi'$. We also know that at least on of these two j-triples is not an element of J_γ, because $\overline{J}_\gamma$ is consistent. Therefore, by lemma 4.16 we have that at least one of ψ and ψ' is non-empty, and thus $\psi \cup \psi'$ is non-empty.

If α is a limit ordinal we have by induction hypothesis that, for all β smaller than α such that $\overline{J}_\beta$ is inconsistent, $\overline{J}_{\beta+1}$ is consistent. Therefore, for all β smaller than α such that $\overline{J}_\beta$ is inconsistent, $\bigcap_{\beta \leq \delta < \alpha} J_\delta \subseteq J_{\beta+1} \subset J_\beta$. From this we can conclude that J_α is consistent. $\square$

Theorem 4.18 (Fixpoint Consistency) *Let Γ_P^ρ be a sequence for a program P. Let J_α be an element of Γ_P^ρ. If J_α is a fixpoint of $\mathcal{S}_P$, then $\overline{J}_\alpha$ is consistent.*

Proof: Suppose $\overline{J}_\alpha$ is inconsistent. Then, by lemma 4.17, $J_{\alpha+1}$ is consistent. But then $J_\alpha \neq J_{\alpha+1}$. This is in contradiction with the fact that J_α is a fixpoint of $\mathcal{S}_P$. $\square$

5 Total stable models as limit fixpoint of $\mathcal{S}_P$

We will now take a look at the fixpoints of $\mathcal{S}_P$ that appear in the sequence of P (we will call them *limit fixpoints*), and prove that they are the total stable models of P. First, we have to define the class of sequences that will contain a fixpoint: *stabilizing sequences*.

Definition 5.1 A sequence Γ_P^ρ is *stabilizing*, if there exists an ordinal α, such that, for all ordinals β greater than α, $J_\alpha = J_\beta$. The *closure ordinal* of Γ_P^ρ is the least ordinal α, such that, for all ordinals β greater than α, $J_\alpha = J_\beta$. $\square$

Definition 5.2 Let P be a program. A j-interpretation J is a *limit fixpoint* of $\mathcal{S}_P$, if there exists a selection strategy ρ for P, such that the sequence Γ_P^ρ is stabilizing and $J = J_\alpha$, where α is the closure ordinal of Γ_P^ρ. $\square$

Theorem 5.3 *Let P be a program. If J is a limit fixpoint of $\mathcal{S}_P$, then $\overline{J}$ is a total stable model of P.*

Proof: J is a limit fixpoint of $\mathcal{S}_P$. Therefore, there exists a selection strategy ρ such that Γ_P^ρ is stabilizing and $J = J_\alpha$, where α is the limit ordinal of Γ_P^ρ. By the Fixpoint Consistency Theorem (4.18), $\overline{J}_\alpha$ is consistent. By the construction of $\mathcal{S}_P$ and the fact that $J_\alpha = J_{\alpha+1}$, $\overline{J}_\alpha$ is a total model of P. Also, by the Supportedness Theorem (4.13) and the Well-Foundedness Theorem (4.14), $<_{J_\alpha}$ is a well-founded support order for $\overline{J}_\alpha$. Therefore, $\overline{J}$ is a total well-supported model of P. Because $\overline{J}$ is total, $U_P(\overline{J})$ is empty. From the Equivalence Lemma (3.13), we conclude that $\overline{J}$ is a total stable model of P. $\square$

So, the limit fixpoints of $\mathcal{S}_P$ are total stable models of P. We will now show the converse: every total stable model is a limit fixpoint of $\mathcal{S}_P$. We define, for every stable model M of P, a class of selection strategies ρ such that M is contained in Γ_P^ρ.

Definition 5.4 Let P be a program and let M be a stable model of P. A *selection strategy for* M is a selection strategy that, for all J such that $\overline{J} \subset M$, selects a j-triple $\langle L, \tau, \psi \rangle$ from $Conflict_P(J)$ or $Choice_P(J)$ such that $L \in M$. □

Lemma 5.5 *Let P be a program, let M be a stable model of P and let J be a j-interpretation such that $\overline{J} \subset M$. Then $\overline{Conflict_P(J)} \subseteq M$*

Proof: Suppose $A \in \overline{Conflict_P(J)}^+$. Then, there exists a clause with conclusion A that is applicable in $\overline{J}$. By construction of $\overline{J}$, this clause is also applicable in M, and therefore A has to be an element of M.

Suppose $A \in \overline{Conflict_P(J)}^-$. Then, all clauses with conclusion A are inapplicable in $\overline{J}$. By construction of $\overline{J}$, these clauses are also inapplicable in M. As a result, we have that every clause in $\frac{P}{M}$ with conclusion A is inapplicable. Because M is the truth-minimal model of $\frac{P}{M}$, we can conclude that $\neg A$ is an element of M. □

Lemma 5.6 *Let P be a program and let M be a stable model of P. Then, there exists a selection strategy ρ for M and for some J_α in Γ_P^ρ, $M = \overline{J}_\alpha$.*

Proof: First, we have to prove that there exists a selection strategy for M. Suppose that J is a j-interpretation such that $\overline{J} \subseteq M$.

1. If ρ has to select from $Conflict_P(J)$ then, by lemma 5.5, any element select by a selection strategy is an element of M.

2. Suppose $Conflict_P(J) = \emptyset$ and $\overline{Choice_P(J)} \cap M \neq \emptyset$. Then we can select an element of M from $\overline{Choice_P(J)}$. Therefore, there exists a selection strategy that selects an element of M from $\overline{Choice_P(J)}$.

3. Suppose $Conflict_P(J) = \emptyset$ and $\overline{Choice_P(J)} \cap M = \emptyset$. Because $\overline{Choice_P(J)} = \neg(\mathcal{B}_P - \overline{J}^{\pm})$ and $\overline{J} \subseteq M$, it follows that $\overline{J}^- = M^-$. Because $Conflict_P(J) = \emptyset$ and M is a supported model of P, we have that $\overline{J}^+ = M^+$. This is in contradiction with the fact that $\overline{J} \subset M$.

So, there exists a selection strategy ρ for M. Consider the sequence Γ_P^ρ. Let α be the least ordinal such that $\overline{J}_\alpha \not\subset M$.

1. If $\alpha = 0$, then $\overline{J}_\alpha = \emptyset \subseteq M$. Because $J_\alpha \not\subset M$, it follows that $J_\alpha = M$.

2. If α is a successor ordinal, then, by definition of ρ, $\overline{J}_{\alpha-1} \subset M$. Also, by definition of ρ, $\overline{J}_\alpha = \overline{J}_{\alpha-1} \cup \{L\}$, where $L \in M$. Because $\overline{J}_\alpha \not\subset M$, we have that $\overline{J}_\alpha = M$.

3. If α is a limit ordinal, then we have that for all β smaller than α, $\overline{J}_\beta \subset M$. By definition of ρ, the prefix of Γ_P^ρ up to (not including) $\overline{J}_\alpha$ is a monotone increasing chain. Therefore, $\overline{J}_\alpha = \bigcup_{\beta<\alpha} \overline{J}_\beta \subseteq M$. Because $\overline{J}_\alpha \not\subset M$, we have that $\overline{J}_\alpha = M$.

So, there exists an J in Γ_P^ρ such that $\overline{J} = M$. □

Theorem 5.7 (Characterization) *Let P be a program. The limit fixpoints of $\mathcal{S}_P$, coincide with the total stable models of P.*

Proof: We have from theorem 5.3 that all limit fixpoints of $\mathcal{S}_P$ contain stable models of P. Also, by lemma 5.6, there exists for every (total) stable model M of P a selection strategy ρ such that M is contained in an element of Γ_P^ρ. Because M is total, it follows that M is a limit fixpoint of $\mathcal{S}_P$. □

6 A characterization of stable models, using $\mathcal{S}_P$

In this section, we characterize the stable models of a program P, using our operator $\mathcal{S}_P$. As we have seen, the total stable models coincide with the limit fixpoints of $\mathcal{S}_P$. This means that we cannot characterize the set of all three-valued stable models as a set of fixpoints of $\mathcal{S}_P$. Instead, we identify the set of stable models of a program with some set of j-interpretations appearing in the sequences for that program.

Lemma 6.1 *Let P be a program and let M be an interpretation of P. M is a stable model of P iff there exists a j-interpretation J in a sequence for P, such that $M = \overline{J}$, $\overline{J}$ is consistent, $Conflict_P(J) = \emptyset$ and $U_P(\overline{J}) = \emptyset$.*

Proof:
($\Leftarrow$) Let J be an element of a sequence for P such that $\overline{J}$ is consistent, $\overline{Conflict_P(J)} = \emptyset$ and $U_P(\overline{J}) = \emptyset$. By the Supportedness Theorem (4.13) and the Well-Foundedness Theorem (4.14), $\overline{J}$ is a well-supported interpretation of P. Also, we know that $\overline{J}$ is consistent and that $U_P(\overline{J}) = \emptyset$. Because $Conflict_P(J) = \emptyset$, we know that for every clause R that is applicable in $\overline{J}$, $concl(R) \in \overline{J}$. Therefore, $\overline{J}$ is a model of P. Finally, by the Equivalence Lemma (3.13), $\overline{J}$ is a stable model of P.
($\Rightarrow$) Let M be a stable model of P. By lemma 5.6, there exists a strategy ρ such that there exists an element J of Γ_P^ρ where $M = \overline{J}$. Clearly, M is consistent. So, we only have to prove that $Conflict_P(J) = \emptyset$ and that $U_P(\overline{J}) = \emptyset$.

- Suppose that $\langle L, \tau, \psi \rangle \in Conflict_P(J)$. If L is positive, then there exists a clause with conclusion L that is applicable in $\overline{J}$. But $\overline{J} = M$ and M is a model of P and therefore $L \in \overline{J}$. If L is negative, then all clauses with conclusion $\neg L$ are inapplicable in $\overline{J}$. The corresponding clauses in $\frac{P}{J}$ will also be inapplicable. Because $\overline{J} = M$ and M is a stable model of P, M is a truth-minimal model of $\frac{P}{M}$ and therefore $L \in \overline{J}$. But the fact that $L \in \overline{J}$ is, by definition of $Conflict_P$, in contradiction with the fact that $\langle L, \tau, \psi \rangle \in Conflict_P(J)$.
- Suppose that $U_P(\overline{J}) \neq \emptyset$. Let $M' = M \cup \neg U_P(\overline{J})$. Clearly, M' is smaller than M in the truth-ordering. But M' is also a model of $\frac{P}{M}$. This is in contradiction with the fact that M is a stable model of P.

□

7 Relating the fixpoint of the Fitting operator to the sequences for P

In the operator $\mathcal{S}_P$, we have a preference for using elements of $Conflict_P$ to extend an interpretation. The definition of $Conflict_P$ bares resemblance to the sets T_P and F_P used by the

Fitting operator [Fit85]. We can identify the least fixpoint of the Fitting operator Φ_P with a special j-interpretation that appears in every sequence for P (in fact, it is the last element of the maximal prefix shared by all sequences for P). First, we give a definition of the Fitting operator.

Definition 7.1 Let P be a program. The Fitting operator Φ_P is defined as follows:

$$\begin{array}{ll} \Phi_P(I) = T_P(I) \cup F_P(I) & \\ \text{where} \quad T_P(I) = \{A \mid \exists_{R \in P} concl(R) = A \wedge prem(R) \subseteq I\} & \\ \qquad\qquad F_P(I) = \{\neg A \mid \forall_{R \in P} concl(R) = A \rightarrow \neg prem(R) \cap I \neq \emptyset\} & \end{array}$$

□

The powers of the Fitting operator can be defined in the same way as we did for $\mathcal{S}_P$. Although the definition of Fitting differs in the case of limit ordinals, we can safely use our definition, because Φ_P is monotone, and for monotone operators both definitions coincide.

Lemma 7.2 *Let Γ_P^ρ be a sequence for a program P. Let α be the least ordinal such that $Conflict_P(J_\alpha) = \emptyset$. Then, $\overline{J}_\alpha$ is the least fixpoint of the Fitting operator Φ_P.*

Proof: Let M be the least fixpoint of Φ_P. We have that $M = \Phi \uparrow^\phi (\emptyset)$, where ϕ is the closure ordinal of Φ_P. We will prove that $\overline{J}_\alpha \subseteq M$ and $\overline{J}_\alpha \supseteq M$.

1. We will prove by induction on β that if $\beta \leq \alpha$ then $\overline{J}_\beta \subseteq M$. For $\overline{J}_0 = \emptyset$, the lemma holds trivially. Assume that for all $\gamma < \beta \leq \alpha$, $\overline{J}_\gamma \subseteq M$.

   If β is a successor ordinal, we have that $J_\beta = J_{\beta-1} \cup \{\langle L, \tau, \psi \rangle\}$. By induction hypothesis, we have that $\overline{J}_{\beta-1} \subseteq M$. Also, by the definition of $Conflict_P(J)$ and Φ_P, we have that $\overline{Conflict_P(J_{\beta-1})} \subseteq M$. Therefore, $\overline{J}_\beta \subseteq M$.

   If β is a limit ordinal, we have, because $\beta \leq \alpha$, that $J_\beta = \bigcup_{\gamma < \beta} J_\gamma$. By induction hypothesis, we have that $\overline{J}_\gamma \subseteq M$, for all $\gamma < \beta$. Therefore, $\overline{J}_\beta \subseteq M$.

2. We have to prove that $\overline{J}_\alpha \supseteq M$. It is enough to prove that $L \notin \overline{J}_\alpha$ implies that $L \notin M$. Suppose $L \notin \overline{J}_\alpha$. There are two cases:

   - L is positive.

     By definition of $\mathcal{S}_P$ and the fact that $Conflict_P(J_\alpha) = \emptyset$, we know that all clauses with conclusion L are not applicable in $\overline{J}_\alpha$. Therefore, by the definition of Φ_P, $L \notin T_P(M)$. As a result, we have that $L \notin M$, because $M^+ = \Phi_P(M)^+ = T_P(M)$.

   - L is negative.

     By definition of $\mathcal{S}_P$ and the fact that $Conflict_P(J_\alpha) = \emptyset$, we know that there exists a clause R in P with conclusion $\neg L$ such that $\neg prem(R) \cap \overline{J}_\alpha = \emptyset$. By this and the definition of Φ_P we have that $L \notin F_P(M)$, and therefore $L \notin M$.

□

8 Finding the Well-Founded Model using $\mathcal{S}_P$

Although the well-founded model, as introduced in [GRS91], is a stable model, and therefore can be found using the results in section 6, we want to give special consideration to this model, because it is one of the most interesting stable models (together with the total stable models).

In this section, we will show that the well-founded model of a program can be found using a special class of selection strategies, the *well-founded strategies*. First, we will give a definition of the well-founded model (for a proper definition, we refer to [GRS91]).

Definition 8.1 Let P be a program. The *well-founded model* of P is the smallest stable model of P (with respect to the knowledge ordering). □

Now, we introduce the class of *well-founded strategies*.

Definition 8.2 Let P be a program. A selection strategy ρ for P is a *well-founded strategy*, if, for all J such that ρ has to select from $Choice_P(J)$ and $U_P(\overline{J})$ is non-empty, ρ selects a j-triple that contains a literal $\neg A$ such that $A \in U_P(\overline{J})$. □

Lemma 8.3 *Let P be a program and let M be a stable model of P. There exists a well-founded selection strategy for M.*

Proof: Let M be a stable model of P. By lemma 5.6, there exist selection strategies for M. Therefore, it suffices to prove that, for a j-interpretation J such that $\overline{J} \subset M$, $Conflict_P(J)$ is empty, $Choice_P(J)$ is non-empty and $U_P(\overline{J})$ is non-empty, $U_P(\overline{J}) \cap M^-$ is non-empty. This follows from the stronger claim that, for $I \subseteq M$, $U_P(I) \subseteq M^-$. By lemma 3.3 in [GRS91], the operator $\mathbf{U}_P$ is monotone. We also have that $U_P(M) = \emptyset$. From these two facts we have that, for $I \subseteq M$,

$$U_P(I) \subseteq U_P(I) \cup I^- = \mathbf{U}_P(I) \subseteq \mathbf{U}_P(M) = U_P(M) \cup M^- = M^-$$

□

Lemma 8.4 *Let P be a program. Every well-founded selection strategy for P is a selection strategy for the well-founded model of P.*

Proof: Let M be the well-founded model of P and let ρ be a well-founded selection strategy for P. Let J be a j-interpretation such that $\overline{J} \subset M$. By lemma 5.5, we know that $\overline{Conflict_P(J)} \subseteq M$. Therefore, we only have to consider the case in which we have to select from $Choice_P(J)$. There are two cases:

- Suppose that $U_P(\overline{J})$ is non-empty. Then, ρ will select a j-triple from $Choice_P(J)$ that contains a literal $\neg A$ such that $A \in U_P(\overline{J})$. Because $\overline{J} \subseteq M$, we have that $U_P(\overline{J}) \subseteq M^-$, and therefore that $A \in M^-$.
- Suppose that $U_P(\overline{J})$ is empty. Then, by lemma 6.1, $\overline{J}$ is a stable model of P. But then, because $\overline{J} \subset M$, $\overline{J}$ is smaller than M in the knowledge-ordering, which is in contradiction with the fact that M is the well-founded model of P.

□

Lemma 8.5 *Let P be a program. M is the well-founded model of P iff M is the first stable model in Γ_P^ρ, where ρ is a well-founded selection strategy for P.*

Proof: Let M be the well-founded model of P and let ρ be a well-founded selection strategy for P. By lemma 8.4, ρ is a selection strategy for M. Therefore, there exists a least ordinal α, such that $\overline{J}_\alpha = M$ (for $J_\alpha \in SeqP\rho$). Moreover, the prefix of Γ_P^ρ ending at J_α is monotone increasing (in the knowledge order). Because M is the knowledge-minimal stable model of P, there does not exist an ordinal β smaller that α such that $\overline{J}_\beta$ is a stable model of P. □

9 On the complexity of $\mathcal{S}_P$

The fact that we can generate all stable models as limits of sequences of interpretations, does not mean that we are in general capable of finding them in finite time. M. Fitting has already shown in [Fit85] that the closure ordinal of his operator Φ_P could be as high as Church-Kleene ω_1, the first nonrecursive ordinal. Because our operator in some sense 'encapsulates' the Fitting operator, we cannot hope to do better with our operator. It would be interesting to define classes of programs whose stable models can be generated in an "acceptable" amount of time.

The first class of programs that comes to mind, is the class of programs P whose Herbrand Base $\mathcal{B}_P$ is finite. The following result is similar to the results obtained in [Fag91] and [SZ90]. First, we have to define a class of selection strategies whose sequences are guaranteed to be stabilizing.

Definition 9.1 Let P be a program and let ρ be a selection strategy for P. We call ρ *fair* if, for all ordinals α and all ordinals β smaller than α, $J_\alpha = J_\beta$ implies that the selection made by ρ for J_α differs from the selection made by ρ for J_β. □

Lemma 9.2 *Let P be a program. If ρ is a fair strategy for P, then the sequence Γ_P^ρ is stabilizing.*

Proof: Suppose there exists a fair strategy ρ such that Γ_P^ρ is not stabilizing. Then, we have that, for all ordinals α, $J_\alpha \neq J_{\alpha+1}$. Because J_α is defined for all ordinals α, there exists at least one j-interpretation J, such that for any ordinal α, there exists an ordinal β such that $\beta > \alpha$ and $J_\beta = J$. This j-interpretation J has a set C associated with it, from which ρ makes a selection (C is one of $Culprit_P(J)$, $Conflict_P(J)$ and $Choice_P(J)$). This set C is non-empty, because otherwise we would have that $J = \mathcal{S}_P^\rho(J)$, and is countable (but possibly infinite), because $\mathcal{B}_P$ is countable. Because ρ is fair, we have that for any two j-interpretations J_α and J_β in Γ_P^ρ such that $J_\alpha = J_\beta$ and $\alpha \neq \beta$, the element selected by ρ for J_α differs from the element selected by ρ for J_β. Therefore, there exists an ordinal γ after which every element of C has been selected once for J. But we know that there exists an ordinal δ such that $\delta > \gamma$ and $J = J_\delta$. At that point, ρ cannot make a fair selection. This is in contradiction with the fact that ρ is a fair selection rule. Therefore, if ρ is fair then Γ_P^ρ is stabilizing. □

Lemma 9.3 *Let P be a program with a finite Herbrand base $\mathcal{B}_P$. Let ρ be a fair strategy for P. The closure ordinal of the sequence Γ_P^ρ is finite.*

Proof: First, note that by lemma 9.2 Γ_P^ρ is stabilizing, and that therefore it has a closure ordinal. Because $\mathcal{B}_P$ is finite, the number of j-interpretations is finite. Furthermore, for any j-interpretation J, the sets $Conflict_P(J)$, $Choice_P(J)$ and $Culprit_P(J)$ are finite. Because of this and the fact that ρ is fair, any j-interpretation J that is not the limit fixpoint of Γ_P^ρ will occur only finitely many times in Γ_P^ρ. As a result, we have that the closure ordinal of Γ_P^ρ is finite. □

Note, that this result is not very surprising. If $\mathcal{B}_P$ is finite, the set of interpretations of P is finite, which means that one can simply enumerate the set of all interpretations of P and test which of them are stable models of P. Thus, any operator should be capable of finding a solution in finite time in this case.

There remains the question of what is the best method for finding stable models of programs in the case of finite Herbrand Bases; generate and testing all consistent interpretations of a program or using $\mathcal{S}_P$ with some carefully chosen family of selection strategies. We have good hope, that the second option will, in general, perform better than the first option. First of all, by inducing some order on the atoms in the Herbrand Base of a program, like Saccà and Zaniolo did with their backtracking operator in [SZ90], we can restrict ourselves to a family of 'ordered' selection strategies, in which the redundancy in partial interpretations being considered is greatly reduced (though not eliminated completely). Moreover, although in general the number of well-supported partial interpretations of a program can be greater than the number of consistent total interpretations of a program, we think that in the typical case the number of well-founded interpretations taken into consideration by $\mathcal{S}_P$ when using a family of ordered selection strategies will be much smaller.

In the remainder of this section, we will formalize the idea of 'using $\mathcal{S}_P$ to find stable models' and present classes of families of strategies that reduce redundancy. First, we introduce the notion of a search-tree for a family of strategies.

Definition 9.4 Let P be a program and let $\mathcal{F}$ be a family of selection strategies for P. $T_{\mathcal{F}}$ is a tree, with j-interpretations as nodes, such that the branches of $T_{\mathcal{F}}$ are exactly the maximal prefixes of sequences Γ_P^ρ such that $\rho \in \mathcal{F}$ and, for any two j-interpretations J and J' in a branch, $J \neq J'$. □

The idea is that –in order to find stable models– we have to traverse the tree $T_{\mathcal{F}}$ for some family $\mathcal{F}$ of strategies. Moreover, we think that building and traversing this tree should account for the exponential part in the costs of finding a stable model; the strategies in $\mathcal{F}$ should be relatively easy to find (i.e. we don't want to define $\mathcal{F}$ as the family of selection strategies that, for every stable model M of P, contains exactly one selection strategy for M). We now have to find some condition that allows us to conclude that the tree for some family of strategies contains stable models. The following lemma will give us such a condition.

Lemma 9.5 *Let P be a program and let $\mathcal{F}$ be a family of selection strategies for P. If, for some stable model M for P, $\mathcal{F}$ contains a selection strategy for M, then $T_{\mathcal{F}}$ has a node n containing a j-interpretation J such that $M = \overline{J}$. Moreover, if M is total, then n is a leaf.*

Proof: Suppose ρ is an element of $\mathcal{F}$ and suppose that ρ is a selection strategy for some stable model M for P. Let α be the least ordinal such that $\overline{J}_\alpha = M$ ($J_\alpha \in \Gamma_P^\rho$).

The prefix of Γ_P^ρ up to J_α increases strictly monotone (inclusion order). Therefore this prefix is contained in a branch in $T_{\mathcal{F}}$. Moreover, if M is total, α is the closure ordinal of Γ_P^ρ, and therefore $J_\alpha = J_{\alpha+1}$. So, if M is total, the prefix of Γ_P^ρ up to J_α is the maximal prefix of Γ_P^ρ that does not contain twice the same j-interpretation, and therefore it coincides exactly with a branch in $T_{\mathcal{F}}$.

The last j-interpretation of the prefix of Γ_P^ρ, contains M. Therefore, there exists a branch in $T_{\mathcal{F}}$ with a node that contains M. Moreover, if M is total, there exists a branch that coincides exactly with this prefix, and therefore the leaf of this branch contains M. □

So, we have to find a family $\mathcal{F}$ of selection strategies such that $\mathcal{F}$ contains a selection strategy for every stable model in M (later on, we will turn our attention to *total* stable models).

We present a number of restrictions on selection strategies, that define a class of so-called *families of $<$-order unfounded-set selection strategies*. Every family in this class will, for every stable model M, contain at least one selection strategy for M, but the size of the search-tree for these families (w.r.t. the search-tree for the family of all selection strategies) will be relatively small. We start by introducing $<$-ordered strategies.

Definition 9.6 Let P be a program and let $<$ be a total order on $\mathcal{L}_P$. We call a strategy ρ for P *$<$-ordered*, if, for all j-interpretations J of P such that ρ has to select from $Conflict_P(J)$, ρ selects a j-triple from $Conflict_P(J)$ containing a literal that is a $<$-minimal element of $\overline{Conflict_P(J)}$. $\square$

The idea of restricting ourselves to $<$-ordered strategies (for some order $<$) is, that we can define an equivalence relation on the selection strategies for P, in a way that every $<$-ordered strategy is a representative of an equivalence class.

Example 9.7 Consider program P_5 consisting of the clauses $p \leftarrow$, $q \leftarrow$, $r \leftarrow p$ and $r \leftarrow q$. We have that $Conflict_{P_5}(\emptyset)$ consists of the j-triples $\langle p, \emptyset, \emptyset \rangle$ and $\langle q, \emptyset, \emptyset \rangle$. There exist two kinds of selection strategies for P_5: the ones that in a given situation select first p, then q or r and then the remaining one, and the ones that –in that given situation– select first q, then p or r and then the remaining one. But any two selection strategies of P that differ in this aspect only, are essentially equivalent, because they both will end up with a j-interpretation containing the interpretation $\{p, q, r\}$ (note however, that the j-interpretations themselves may differ). $\circ$

Lemma 9.8 *Let P be a program and let $<$ be a total order on $\mathcal{L}_P$. Then, for every stable model M of P, the family of $<$-ordered selection strategies contains a selection strategy for M.*

Proof: Let M be a stable model of P. By lemma 5.6, there exist selection strategies for M. Therefore, it suffices to prove that, for a j-interpretation J such that $\overline{J} \subset M$ and $Conflict_P(J)$ is non-empty, $D \cap M \neq \emptyset$, where D is the set of $<$-minimal elements of $\overline{Conflict_P(J)}$. But this follows from the fact that, by definition of D and lemma 5.5, $D \subseteq \overline{Conflict_P(J)} \subseteq M$. $\square$

We can strengthen this result by combining it with the result on well-founded strategies.

Lemma 9.9 *Let P be a program and let $<$ be a total order on $\mathcal{L}_P$. Let $\mathcal{F}$ be the family of strategies that are both well-founded and $<$-ordered. Then, for every stable model M of P, $\mathcal{F}$ contains a selection strategy for M.*

Proof: The proof follows directly from lemma's 8.3 and 9.8, because the condition for $<$-orderedness is only relevant if an element of $Conflict_P$ is selected, while the condition for well-foundedness is only relevant if an element of $Choice_P$ is selected. $\square$

A further strengthening is possible by using the order on $\mathcal{L}_P$ when selecting an element of U_P.

Definition 9.10 Let P be a program and let $<$ be a total order on $\mathcal{L}_P$. We call a strategy ρ for P an *$<$-order unfounded-set strategy*, if, for all j-interpretations J of P:

- if ρ has to select from $Conflict_P(J)$, it selects j-triple that contains a $<$-minimal literal of $\overline{Conflict_P(J)}$, and
- if ρ has to select from $Choice_P(J)$ and $U_P(\overline{J})$ is non-empty, it selects a j-triple that contains a $<$-minimal literal of $U_P(\overline{J})$.

$\square$

Lemma 9.11 *Let P be a program and let $<$ be a total order on $\mathcal{L}_P$. Let $\mathcal{F}$ be the family of $<$-order unfounded-set strategies. Then, for every stable model M of P, $\mathcal{F}$ contains a selection strategy for M.*

Proof: By definition, $\mathcal{F}$ is contained in the family of selection strategies that are both $<$-ordered and well-founded. Let M be a stable model of P and let J be a j-interpretation of P such that $\overline{J} \subset M$, $Conflict_P(J)$ is empty, $Choice_P(J)$ is non-empty and $U_P(\overline{J})$ is non-empty. We know that $U_P(\overline{J}) \subseteq M^-$ (see lemma 8.3). But the $<$-minimal element of $U_P(\overline{J})$ is clearly an element of $U_P(\overline{J})$, and therefore an element of M^-. Therefore there exist $<$-order unfounded set strategies for M. $\square$

We will conclude this section by defining a class of families of selection strategies such that, for any family in this class and any *total* stable model M of P, the family contains a selection strategy for M. For this, we need to define a special dependency relation on the unknown atoms of an interpretation.

Definition 9.12 Let P be a program and let I be an interpretation for P. We define the *dependency relation* $<_{\mathcal{D}_I}$ *on* $\mathcal{B}_P - I^{\pm}$ as the transitive closure of the relation $\mathcal{D}_I$, which is defined as follows: $A' \, \mathcal{D}_I \, A$ iff there exists a rule R in P with conclusion A that is neither applicable nor inapplicable in I such that $A' \in prem(R)^{\pm}$. An element A of $\mathcal{B}_P - I^{\pm}$ is called $<_{\mathcal{D}_I}$*-minimal* if, for all A' such that $A' <_{\mathcal{D}_I} A$, $A <_{\mathcal{D}_I} A'$. $\square$

Example 9.13 Consider program P_6 consisting of the clauses $p \leftarrow \neg p$, $p \leftarrow q$, $r \leftarrow s$, $s \leftarrow r$, $t \leftarrow s$, $u \leftarrow \neg v$ and $v \leftarrow \neg u$. Let $I = \{\neg q\}$ be an interpretation of P_6. Then we have that $\mathcal{D}_I = \{\langle p, p\rangle, \langle r, s\rangle, \langle s, r\rangle, \langle s, t\rangle, \langle u, v\rangle, \langle v, u\rangle\}$. So, $\{p, s, r, u, v\}$ is the set of $<_{\mathcal{D}_I}$-minimal elements. $\circ$

Definition 9.14 Let P be a program. We call a strategy ρ for P $\mathcal{D}$*-ordered*, if, for all j-interpretations J of P such that ρ has to select from $Choice_P(J)$, ρ selects a j-triple containing a literal $\neg A$ such that A is $<_{\mathcal{D}_{\overline{J}}}$-minimal. $\square$

Lemma 9.15 *Let P be a program and let $<$ be a total order on $\mathcal{L}_P$. Let $\mathcal{F}$ be the family of selection strategies that are both $<$-ordered and $\mathcal{D}$-ordered. For every total stable model of P, $\mathcal{F}$ contains a selection strategy for M.*

Proof: Let M be a total model of P. Because the conditions for $<$-orderedness and $\mathcal{D}$-orderedness do not interfere with eachother and because, by lemma 9.8, there exist $<$-ordered selection strategies for M, we only have to show that the $\mathcal{D}$-orderedness condition does not interfere with the condition for strategies for M. Let J be a j-interpretation sucht that $\overline{J} \subseteq M$, $Conflict_P(J)$ is empty and $Choice_P(J)$ is non-empty. Let D be the set of $<_{\mathcal{D}_{\overline{J}}}$-minimal elements of $\overline{J}$. It suffices to prove the $D \cap M^-$ is non-empty. First, note that D is non-empty because $\overline{J} \subset M$.

Suppose that $D \cap M^-$ is empty. Then, because M is total, $D \subseteq M^+ \subseteq M$. Now, let A be an element of D and let R be a clause with conclusion A that is applicable in M (there has to exist at least one such clause). Because $Conflict_P(J)$ is empty, $prem(R)^{\pm} \cap I^{\pm}$ is non-empty. Moreover, because A is $<_{\mathcal{D}_{\overline{J}}}$-minimal, $prem(R)^{\pm} \cap D$ is non-empty. Finally, because R is consistent in M and $D \subseteq M$, we know that $prem(R)^- \cap D$ is empty. So, for all clauses R with conclusion in D that are applicable in M, $D \cap prem(R)^+$ is non-empty. But then, D is an unfounded set of M, and thus $D \subseteq U_P(M) \subseteq \mathbf{U}_P(M) = M^-$, which contradicts the assumption that $D \cap M^-$ is empty. $\square$

10 Conclusion

In this paper, we have presented an operator that generates sequences of interpretations. We have shown that the limits of these sequences are exactly all total stable models of a general logic program. Moreover, the set of all stable models can be identified as a subset of the interpretations generated by the operator. Furthermore, we have shown that the least fixpoint of the Fitting operator appears in all sequences generated by our operator, and that we can find the well-founded model, using a special family of selection strategies.

It would be interesting to find classes of selection strategies that can be implemented efficiently, are complete (i.e. are capable of finding all (total) stable models), and have small closure ordinals. The families of selection strategies we presented here seems to be good candidates, and it might be possible that we are capable of restricting these classes further.

Acknowledgements

This paper was partially supported by a grant from SION, a department of NWO, the Netherlands Organization for Scientific Research. I would like to thank Krzysztof Apt for his support, proof reading and giving valuable suggestions for improving the paper. Also, I would like to thank Elena Marchiori for proof reading the paper.

References

[Fag91] Francois Fages. A new fixpoint semantics for general logic programs compared with the well-founded and the stable model semantics. *New Generation Computing*, 9:425–443, 1991.

[Fit85] Melvin Fitting. A Kripke-Kleene semantics for logic programs. *Journal of Logic Programming*, 2(4):295–312, 1985.

[GL88] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proceedings of the International Conference on Logic Programming*, pages 1070–1080, 1988.

[GRS91] Allen van Gelder, Kenneth A. Ross, and John S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, July 1991.

[Prz90a] Teodor Przymusinski. Well-founded semantics coincides with three-valued stable semantics. *Fundamenta Informaticae*, 13(4):445–463, 1990.

[Prz90b] Teodor C. Przymusinski. Extended stable semantics for normal and disjunctive programs. In *Proceedings of the International Conference on Logic Programming*, pages 459–477, 1990.

[SZ90] Domenico Saccà and Carlo Zaniolo. Stable models and non-determinism in logic programs with negation. In *Proceedings of the ACM Symposium on Principles of Database Systems*, page 16, 1990.

[Teu93] Frank Teusink. A characterization of stable models using a non-monotonic operator. In *Proceedings of the workshop on Logic Programming and Non-Monotonic Reasoning*, 1993.