Partial Models of Extended Generalized Logic Programs

José J. Alferes[1,3], Heinrich Herre[2], Luís Moniz Pereira[3]

Departamento de Matemática, Universidade de Évora, jja@di.fct.unl.pt[1],
Centro de Inteligência, Universidade Nova de Lisboa, 2825 Monte da Caparica,
lmp@di.fct.unl.pt[3],
Institute of Informatics, University of Leipzig, Augustusplatz 10-11,
herre@informatik.uni-leipzig.de[2]

Abstract. In recent years there has been an increasing interest in extensions of the logic programming paradigm beyond the class of normal logic programs motivated by the need for a satisfactory respresentation and processing of knowledge. An important problem in this area is to find an adequate declarative semantics for logic programs. In the present paper a general preference criterion is proposed that selects the 'intended' partial models of extended generalized logic programs which is a conservative extension of the stationary semantics for normal logic programs of [13], [15] and generalizes the WFSX-semantics of [12]. The presented preference criterion defines a partial model of an extended generalized logic program as intended if it is generated by a stationary chain. The GWFSX-semantics is defined by the set-theoretical intersection of all stationary generated models, and thus generalizes the results from [9] and [1].

1 Introduction

Declarative semantics provides a mathematical precise definition of the meaning of a program in a way, which is independent of procedural considerations. Finding a suitable declarative or intended semantics is an important problem in logic programming and deductive databases. Logic programs and deductive databases should be as easy to write and comprehend and as close to natural discourse as possible.

Standard logic programs are not sufficiently expressive for comprehensible representation of large classes of knowledge bases and of informal descriptions. Formalisms admitting more complex formulas, as extended generalized logic programs, are more expressive and natural to use since they permit in many cases easier translation from natural language expressions and from informal specifications. The expressive power of generalized logic programs also simplifies the problem of translation of non-monotonic formalisms into logic programs, as shown in [4], [5], and consequently facilitates using logic programming as an inference engine for non-monotonic reasoning. We assume that a reasonable extension of logic programs should satisfy the following conditions:

1. The proposed syntax of rules in such programs resembles the syntax of logic programs but it applies to a significantly broader class of programs.
2. The proposed semantics of such programs constitute a natural extension of the semantics of normal logic programs;
3. There is a natural relationship between the proposed class of programs and their semantics and broader classes of non-monotonic formalisms.

We believe that the class of extended generalized logic programs and the stationary generated semantics, introduced in this paper, presents an extension of logic programming for which the above mentioned principles 1. and 2. can be realized. There are also results in [4] and [5] partially realizing principle 3, where relations to temporal logic and default logic are studied.

A set of facts can be viewed as a database whose semantics is determined by its minimal models. In the case of logic programs, where there are rules, minimal models are not adequate because they are not able to capture the directedness of rules. Therefore, *partial stable* models in the form of certain fixpoints have been proposed by [13] , [15], and [12]. We generalize this notion by presenting a definition which is neither fixpoint-based nor dependent on any specific rule syntax. We call our preferred models *stationary generated* because they are generated by a *stationary chain*, i.e. a stratified sequence of rule applications where all applied rules satisfy certain persistance properties. We show the partial stable models of an extended normal programs coincide with its stationary generated models. Hence, our semantics generalizes the WFSX-semantics.

The paper has the following structure. After introducing some basic notation in section 2, we recall some facts about Herbrand model theory and sequents in section 3. In section 4, we define the general concept of a stationary generated model[1] and introduce the GWFSX-semantics. In section 5 we investigate the relationship of the stationary generated models to the original fixpoint-based definitions for normal programs. In particular, we relate the stationary semantics to the WFSX-semantics for extended normal logic programs. It turns out that, for extended normal logic programs, the stationary generated models and the partial stable models coincide.

2 Preliminaries

A *signature* $\sigma = \langle Rel, ExRel, Const, Fun \rangle$ consists of a set Rel of relation symbols, a set of it exact relations symbols $ExRel \subseteq Rel$, a set $Const$ of constant symbols, and a set Fun of function symbols. U_σ denotes the set of all ground terms of σ. The logical functors are $not, \neg, \wedge, \vee, \rightarrow, \forall, \exists$, and the functors $\mathbf{t}, \mathbf{f}, \mathbf{u}$ of arity zero. $L(\sigma)$ is the smallest set containing the constants $\mathbf{t}, \mathbf{f}, \mathbf{u}$ and the atomic first order formulas of σ, and being closed with respect to the following conditions: if $F, G \in L(\sigma)$, then $\{\, not F, \neg F,\ F \wedge G,\ F \vee G,\ F \rightarrow G,\ \exists x F,\ \forall x F\} \subseteq L(\sigma)$. $L^0(\sigma)$ denotes the corresponding set of sentences (closed formulas), where the constants

[1] The term "stationary" is borrowed from [15], but the concept of a stationary generated model differs essentially from the stationary model as introduced in [15].

$\mathbf{t}$ (true) , $\mathbf{f}$ (false) , $\mathbf{u}$ (undefined) are considered as sentences. For sublanguages of $L(\sigma)$ formed by means of a subset $\mathcal{F}$ of the logical functors, we write $L(\sigma;\mathcal{F})$. Let $L_1(\sigma) = L(\sigma;\{\, not, \neg, \wedge, \vee, \mathbf{t}, \mathbf{f}, \mathbf{u}\})$, $L_2(\sigma) = L(\sigma;\{\, not, \neg, \wedge, \vee\})$, $L_p(\sigma) = L_1(\sigma) \cup \{F \to G : F \in L_1(\sigma), G \in L_2(\sigma)\}$; $L_p(\sigma)$ is the set of program formulas over σ. Program formulas are equivalently denoted by expressions of the form $G \leftarrow F$ using the left-directed arrow. With respect to a signature σ we define the following sublanguages: $\mathrm{At}(\sigma) = L(\sigma;\{\mathbf{t}, \mathbf{f}, \mathbf{u}\})$, the set of all atomic formulas (also called *atoms*). The set $\mathrm{GAt}(\sigma)$ of all ground atoms over σ is defined as $\mathrm{GAt}(\sigma) = \mathrm{At}(\sigma) \cap \mathrm{L}^0(\sigma)$. $\mathrm{Lit}(\sigma) = L(\sigma;\{\neg, \mathbf{t}, \mathbf{f}, \mathbf{u}\})$, the set of all *objective literals*; we identify the literal $\neg\neg l$ with l. The set $\mathrm{OL}(\sigma)$ of all objective ground literals over σ is defined as $\mathrm{OL}(\sigma) = \mathrm{Lit}(\sigma) \cap L^0(\sigma)$. For a set X of formulas let $notX = \{\, notF | F \in X\}$. The set $\mathrm{XLit}(\sigma)$ of all extended literals over σ is defined by $\mathrm{XLit}(\sigma) = \mathrm{Lit}(\sigma) \cup not\mathrm{Lit}(\sigma)$. Finally, $\mathrm{XG}(\sigma) = \mathrm{OL}(\sigma) \cup not\mathrm{OL}(\sigma)$ is the set of all extended grounds literals.

We introduce the following conventions. When $L \subseteq L(\sigma)$ is some sublanguage, L^0 denotes the corresponding set of sentences. If the signature σ does not matter, we omit it and write, e.g., L instead of $L(\sigma)$. If Y is a set and $\leq$ a partial ordering on Y then $\mathrm{Min}_{\leq}(Y)$ denotes the set of all minimal elements of $(Y, \leq)$. $Pow(X) = \{Y \mid Y \subseteq X\}$ denotes the power set of X.

Definition 1 (Partial Interpretation) *Let $\sigma = \langle Rel, Const, Fun\rangle$ be a signature. A partial interpretation I of signature σ is defined by a function $I : \mathrm{OL}(\sigma) \longrightarrow \{0, \frac{1}{2}, 1\}$ satisfying the conditions $I(\mathbf{t}) = 1, I(\neg\mathbf{f}) = 1$, $I(\mathbf{u}) = I(\neg\mathbf{u}) = \frac{1}{2}$. I is said to be coherent if the following coherence principles are satisfied: $I(\neg a) = 1$ implies $I(a) = 0$, and $I(a) = 1$ implies $I(\neg a) = 0$ for every ground atom a.*

A partial σ-interpretation I can equivalently be represented by a set of extended ground literals $I^* \subseteq \mathrm{OL}(\sigma) \cup not\mathrm{OL}(\sigma)$ by the following stipulation: $I^* = \{l \mid I(l) = 1\} \cup \{\, notl \mid I(l) = 0\}$. Then, I^* satisfies the following conditions:

1. *$\{\mathbf{t}, \neg\mathbf{f}\} \subseteq I^*$, $\{\mathbf{u}, \neg\mathbf{u}\} \cap I^* = \emptyset$.*
2. *There is no objective ground literal $l \in \mathrm{OL}$ such that $\{l,\, notl\} \subseteq I$ (consistency).*
   If I is coherent then I^* satisfies the additional conditions:
3. *$\neg a \in I^*$ implies $not a \in I^*$ for every ground atom a.*
4. *$a \in I^*$ implies $not\neg a \in I^*$ for every ground atom a.*

Conversely, every set J of extended ground literals satisfying the conditions 1.and 2. defines a function $I : \mathrm{OL}(\sigma) \longrightarrow \{0, \frac{1}{2}, 1\}$ being an interpretation by the following conditions: $I(l) = 1$ iff $l \in J$, $I(l) = 0$ iff $notl \in J$, $I(l) = \frac{1}{2}$ iff $\{l,\, notl\} \cap J = \emptyset$. If J satisfies conditions 3. and 4, then I is coherent.

Remark: In the sequel we use both descriptions of an interpretation (the functional or the literal version), and it should be clear from the context which kind of representation is meant.

For a partial interpretation I let $Pos(I) = I \cap \mathrm{OL}$ and $Neg(I) = I \cap \, not\, OL$. A partial interpretation I is two-valued (or total) if for every $l \in OL$ the condition $\{l,\, not\, l\} \cap I \neq \emptyset$ is satisfied. A *generalized partial interpretation* is a (arbitrary) set $I \subseteq \mathrm{OL}(\sigma) \cup not\,\mathrm{OL}(\sigma)$ of extended ground literals. A generalized partial interpretation is said to be *consistent* if conditions 1) and 2) in definition 1 are satisfied, and it is called *coherent* if the conditions 3) and 4) from definition 1 are fulfilled. The class of all generalized partial σ-interpretations is denoted by $\boldsymbol{I}_{gen}(\sigma)$, and the class of all consistent partial σ-interpretations is denoted by $\boldsymbol{I}(\sigma)$, and the class of all consistent coherent interpretations by $\boldsymbol{I}_{coh}(\sigma)$. In the sequel we shall also simply say 'interpretation' instead of 'consistent partial interpretation'. A *valuation* over an interpretation I is a function ν from the set of all variables *Var* into the Herbrand universe U_σ, which can be naturally extended to arbitrary terms by $\nu(f(t_1, \ldots, t_n)) = f(\nu(t_1), \ldots, \nu(t_n))$. Analogously, a valuation ν can be canonically extended to arbitrary formulas F, where we write $F\nu$ instead of $\nu(F)$. The model relation $\models \subseteq \boldsymbol{I}(\sigma) \times L^0(\sigma)$ between an interpretation and a sentence from $L_p(\sigma)$ is defined inductively as follows; it generalizes the definition 4.1.4 in [1] to arbitrary formulas from L_p.

Definition 2 (Model Relation) *Let I be an interpretation of signature σ. Then I can be naturally expanded to a function $\overline{I}$ from the set of all sentences of $L_p(\sigma)$ into the set $\{0, \frac{1}{2}, 1\}$.*

1. $\overline{I}(l) = I(l)$ *for every* $l \in OL$.
2. $\overline{I}(\, not\, F) = 1 - \overline{I}(F)$;
3. $\overline{I}(F \wedge G) = min\{\overline{I}(F), \overline{I}(G)\}$.
4. $\overline{I}(F \vee G) = max\{\overline{I}(F), \overline{I}(G)\}$
5. $\overline{I}(\neg\neg F) = \overline{I}(F)$;
6. $\overline{I}(\neg\, not\, F) = \overline{I}(F)$;
7. $\overline{I}(\neg(F \wedge G)) = \overline{I}(\neg F \vee \neg G)$;
8. $\overline{I}(\neg(F \vee G)) = \overline{I}(\neg F \wedge \neg G)$;
9. $\overline{I}(F \to G) = 1$ *if* $\overline{I}(F) \leq \overline{I}(G)$ *or* $\overline{I}(\neg G) = 1$ *and* $\overline{I}(F) = \frac{1}{2}$.
10. $\overline{I}(F \to G) = 0$ *if condition 9. is not satisfied.* [2]
11. $\overline{I}(\exists x F(x)) = sup\{\overline{I}(F(x/t)) | t \in U_\sigma\}$
12. $\overline{I}(\forall x F(x)) = inf\{\overline{I}(F(x/t)) | t \in U_\sigma\}$.

To simplify the notation we don't distinguish between I and $\overline{I}$. A sentence F is true in I, denoted by $I \models F$, iff $I(F) = 1$. For arbitrary formulas F we write $I \models F \iff I \models F\nu$ for all $\nu : Var \to U_\sigma$. For a set X of formulas we write $I \models X$ iff for all $F \in X$ it holds $I \models F$. I is said to be model of a set X of formulas if $I \models X$, and we use the notation $\mathrm{Mod}(X) = \{I \mid I \models X\}$.

Remark: If the conditions 9. and 10. of definition 2 are replaced by 9a. $\overline{I}(F \to G) = 1$ if $\overline{I}(F) \leq \overline{I}(G)$ and 10a. $\overline{I}(F \to G) = 0$ if $\overline{I}(F) \not\leq \overline{I}(G)$, then we get the truth-relation considered in [13] which we denote by $I \models_{pr} F$.

[2] The condition "not 9." is equivalent to $\overline{I}(F) \not\leq \overline{I}(G)$ and ($\overline{I}(\neg G) < 1$ or $\overline{I}(F) = 1$).

A coherent consistent interpretation I is called a *AP-model* of a set X of formulas if and only if for all $F \in X$ it holds $I \models F$. An interpretation I is called a *Pr-model* of X iff $I \models_{pr} X$.

Example 1 (AP-models and Pr-models) *Let P be the following program:* $\{\neg b;\ c \leftarrow\ not \neg c;\ a \leftarrow\ not\, a,\ not\, c;\ \neg a \leftarrow\ not\, c;\ b \leftarrow a\}$.
P has following AP-models:

$M_1 = \{\neg b,\ not\, b\}$
$M_2 = \{\neg b,\ not\, b, c,\ not \neg c\}$
$M_3 = \{\neg b,\ not\, b, c,\ not \neg c,\ not\, a\}$
$M_4 = \{\neg b,\ not\, b,\ not\, c, \neg c\}$
$M_5 = \{\neg b,\ not\, b, \neg a,\ not\, a\}$
$M_6 = \{\neg b,\ not\, b, \neg a,\ not\, a, c,\ not \neg c\}$
$M_7 = \{\neg b,\ not\, b,\ not \neg a\}$
$M_8 = \{\neg b,\ not\, b, c,\ not \neg c,\ not \neg a\}$
$M_9 = \{\neg b,\ not\, b, c,\ not \neg c,\ not\, a,\ not \neg a\}$
$M_{10} = \{\neg b,\ not\, b,\ not\, c, \neg c,\ not \neg a\}$

Only the models M_3, M_6, M_9 are Pr-models.

Remark: The relevance of AP-models wrt. to Pr-models is that the former impose coherence on interpretations while the latter not. So, take for instance the set $P = \{a \leftarrow \neg a, b;\ b \leftarrow\ not\, b\}$. Now $\{\neg a,\ not\, a,\ not \neg b\}$ is an AP-model but not an Pr-model of P. While $\{\neg a,\ not \neg b\}$ is a Pr-model of P but not an AP-model because it is not an coherent interpretation. In this case, coherence imposes "$not\, a$" true on the basis of the truth of $\neg a$, if "$not\, a$" is otherwise undefined as in P. That is, explicit negation $\neg$ overrides undefinedness. Pr-models do not impose coherence they allow the unnatural result that though "a" is explicitly false "$not\, a$" remains undefined. In our opinion, the truth relation $\models$ of definition 2 is better suited to treat coherent interpretations.

Definition 3 (Partial Orderings between Interpretations) *Let $I, I_1 \in \boldsymbol{I}_{gen}$ be two generalized interpretations.*

1. *Let $I \preceq I_1$ if and only if $Pos(I) \subseteq Pos(I_1)$ and $Neg(I_1) \subseteq Neg(I)$. $\preceq$ is called the truth-ordering between interpretations, and I_1 is said to be a truth-extension (briefly t-extension) of I.*
2. *I_1 is informationally greater or equal to I if and only if $I \subseteq I_1$. The partial ordering $\subseteq$ between interpretations is called information-ordering. I_1 is said to be an information-extension (briefly i-extension) of I.*

Proposition 1 *The system $\mathcal{C} = (\boldsymbol{I}_{coh}, \preceq)$ of coherent and consistent generalized partial interpretations is a complete lower semi-lattice.*

Proof: Let $\Omega \subseteq \boldsymbol{I}$ an arbitrary subset. We show that there exists a greatest lower bound for Ω with respect to $\preceq$. Let I be defined by $Pos(I) = \bigcap\{Pos(J) : J \in \Omega\}$, and $Neg(I) = \bigcup\{Neg(J) : J \in \Omega\}$. Obviously, $I \preceq J$ for every $J \in J \in \Omega$, and I is the greatest lower bound within $(\boldsymbol{I}_{gen}, \preceq)$. I is consistent. Assume there are $l \in \mathrm{OL}$ such that $\{l, \mathit{not} l\} \subseteq I$. Then $l \in I$. From $\mathit{not} l \in I$ there follows the existence of a $J \in \Omega$ such that $\mathit{not} l \in J$, a contradiction. The coherency of I is immediate. Hence $I \in \boldsymbol{I}_{coh}$. □

Remark: $\mathcal{C}$ is not a lattice, there are elements $I, J \in \boldsymbol{I}$ having no least upper bound. Take, for example: $I = \{\neg a, \mathit{not} a\}, J = \{\neg b, \mathit{not} b\}$. Every upper bound K of I, J has to contain $\{\neg a, \neg b\}$, and because of coherence also $\{\mathit{not} a, \mathit{not} b\}$. But this violates the condition $I \preceq K$ because $Neg(K) \not\subseteq Neg(I)$. The set of (consistent) interpretations $(\boldsymbol{I}, \preceq)$ is a complete lattice.

Let $I_1 \preceq I_2 \preceq \ldots \preceq I_\alpha \preceq \ldots$ be a t-increasing sequence of interpretations. The supremum $J = sup\{I_\alpha : \alpha < \kappa\}$ of this sequence is defined by $Pos(J) = \bigcup_\alpha < \kappa : Pos(I_\alpha)$ and $Neg(J) = \bigcap_\alpha < \kappa : Neg(I_\alpha$. For a t-decreasing sequence of interpretations $I_1 \succeq I_2 \succeq \ldots \succeq I_\alpha \succeq \ldots$ its infimum $J = inf\{I_\alpha : \alpha < \kappa\}$ is defined by the following conditions $Neg(J) = \bigcup_{\alpha<\kappa} Neg(I_\alpha)$, and $Pos(J) = \bigcap_{\alpha<\kappa} Pos(I_\alpha)$. An interpretation I is called a *t-minimal* model of X if $I \in \mathrm{Min}_\preceq(\mathrm{Mod}(X))$, it is called *i-minimal* if $I \in \mathrm{Min}_\subseteq(\mathrm{Mod}(X))$. The following version of proposition 1 is true: for any t-increasing sequence $\{I_\alpha \mid \alpha < \kappa\}$ of coherent interpretations the interpretation $sup_{\alpha<\kappa} I_\alpha$ is itself coherent.

3 Sequents and Programs

Here, we propose to use sequents for the purpose of representing rule knowledge.[3] A sequent, then, is a concrete expression representing some piece of knowledge.

Definition 4 (Sequent) *A* sequent *s is an expression of the form*

$$F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$$

where $F_i, G_j \in L(\sigma, \{\wedge, \vee, \neg, \mathit{not}\})$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. The body *of s, denoted by $B(s)$, is given by $\{F_1, \ldots, F_m\}$, and the* head *of s, denoted by $H(s)$, is given by $\{G_1, \ldots, G_n\}$.* Seq(σ) *denotes the class of all sequents s such that $Hs, Bs \subseteq L(\sigma; \wedge, \vee, \neg, \mathit{not})$, and for a given set $S \subseteq$* Seq(σ), *$[S]$ denotes the set of all ground instances of sequences from S. Sometimes, we write a sequent in the following rule-form: $G_1 \vee \ldots \vee G_n \leftarrow F_1, \ldots, F_m$.*

Definition 5 (Model of a Sequent) *Let $I \in \boldsymbol{I}$. Then,*

$$I \models F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$$

if and only if for all ground substitutions the condition

$$I \models \bigwedge_{i \le m} F_i \nu \rightarrow \bigvee_{j \le n} G_j \nu$$

is satisfied. In this case, I is said to be model of $F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$.

---

[3] The motivation for choosing sequents is to get a connection to Gentzen-like proof systems. Furthermore, the sequent-arrow $\Rightarrow$ and the material implication $\rightarrow$ have different properties.

We define the following classes of sequents corresponding to different types of logic programs.

- $\mathrm{EPLP}^*(\sigma) = \{s \in \mathrm{Seq}(\sigma) : H(s) \in \mathrm{Lit}(\sigma),\ B(s) \subseteq \mathrm{Lit}(\sigma) \cup \{\mathbf{t}, \mathbf{u}, \mathbf{f}\}\}$.
- $\mathrm{EPLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) : H(s) \in \mathrm{Lit}(\sigma),\ B(s) \subseteq \mathrm{Lit}(\sigma)\}$.
- $\mathrm{ENLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) : H(s) \in \mathrm{Lit}(\sigma),\ B(s) \subseteq \mathrm{XLit}(\sigma)\}$.
- $\mathrm{EDLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) : H(s) \subseteq \mathrm{Lit}(\sigma),\ B(s) \subseteq \mathrm{XLit}(\sigma), H(s) \neq \emptyset\}$.
- $\mathrm{EGLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) : H(s), B(s) \subseteq L(\sigma;\ not, \neg, \wedge, \vee)\}$.

Subsets of EPLP* are called *non-negative extended* logic programs, programs associated to EPLP are called *extended positive* programs, ENLP relates to *extended normal* programs, EDLP to *extended disjunctive programs*, and EGLP to *extended generalized* logic programs. The following lemma is an important tool for analyzing the structure of the partial models of a generalized logic program.

Lemma 1. [4]

1. *Let $J_0 \succeq J_1 \succeq \ldots J_n \succeq \ldots$ be an infinite t-decreasing sequence of partial interpretations and $J = inf\{J_n \mid n < \omega\}$. Let $F \in L^0_p(\sigma)$. Then there exists a number k such that for all $s > k$ the condition $J(F) = J_s(F)$ is satisfied.*
2. *Let $J_0 \preceq J_1 \preceq \ldots J_n \preceq \ldots$ be an infinite t-increasing sequence of partial interpretations and $J = sup\{J_n \mid n < \omega\}$. Let $F \in L^0_p(\sigma)$. Then there exists a number k such that for all $s > k$ the condition $J(F) = J_s(F)$ is satisfied.*

Proposition 2 *Let P be a set of formulas from L_p and K an interpretation. Let I be a model of P such that $K \preceq I$. Then there exists a model $J \models P$ satisfying the following conditions:*

1. *$K \preceq J \preceq I$;*
2. *for every interpretation J_1 the conditions $K \preceq J_1 \preceq J$ and $J_1 \models P$ imply $J = J_1$.*

Proof: Let be $I \models P$ and $\Omega(K, I) = \{K | K \preceq M \preceq I$ and $M \models P\}$. Assume $I = J_0 \succeq J_1 \succeq \ldots \succeq J_n \ldots$, and $J_n \in \Omega(K, I)$. Let be $J = inf_{n<\omega} J_n$. We show that $J \models P$; then, by Zorn's lemma, the set $\Omega(K, I)$ contains a $\preceq$-minimal element satisfying the conditions 1. and 2. Let $r : B(r) \Rightarrow H(r) \in [P]$, and $F := \bigwedge B(r)$, $G := \bigvee H(r)$. It is sufficient to show $J \models F \to G$. Assume this is not the case, then $J \not\models F \to G$, and by definition $J(F \to G) = 0$. By lemma 1 there is a number $k < \omega$ such that for all $s \geq k$ we have $J_s(F \to G) = 0$, this is a contradiction, because $J_s \models P$. □

Corollary 3 *Let P be an extended generalized logic program. Every partial model of P is an t-extension of a t-minimal partial model of P and can be t-extended to a t-maximal partial model of P.*

Proposition 4 *Every non-negative extended logic program P having a partial Pr-model has a t-least partial Pr-model.*

[4] Complete proofs will be published in the full paper [2].

Remark: Proposition 4 is not true for AP-models as the following example shows.

Example 2 *Let* $P = \{c \leftarrow \mathbf{f};\ a \leftarrow b;\ b \leftarrow \mathbf{u};\ \neg a;\ d \leftarrow b\}$.
Let $I = \{\neg a,\ not\, a,\ not\, c,\ not\neg b,\ not\neg c,\ not\neg d\}$. *Obviously,* I *is a t-minimal* AP*-model of* P. *Take* $I_1 = (I - \{\ not\neg d\}) \cup \{\neg d,\ not\, d\}$. *Then* $I_1 \not\preceq I$ *and* $I \not\preceq I_1$. *It is easy to see that* I_1 *is a t-minimal model of* P.

Example 3 *The following program* P *has* AP*-models but no coherent* Pr*-model.*
$P = \{a \leftarrow b;\ \neg a;\ b \leftarrow \mathbf{u}\}$.
Then $\{\neg a,\ not\, a,\ not\neg b\}$ *is an* AP*-model.*

4 Stationary Generated Models and GWFX-semantics

A preferential semantics is given by a preferred model operator $\Phi : Pow(Seq) \rightarrow Pow(\boldsymbol{I})$ satisfying the condition $\Phi(P) \subseteq \mathrm{Mod}(P)$ for $P \subseteq Seq$, and determining the associated preferential entailment relation defined by $P \models_{\Phi} F$ iff $\Phi(P) \subseteq \mathrm{Mod}(F)$. The definition of a stationary chain uses certain persistance properties of formulas which are based on the following notion of truth intervals.

Definition 6 (Truth Interval of Interpretations) *Let* $I_1, I_2 \in \boldsymbol{I}$. *Then,* $[I_1, I_2] = \{I \in \boldsymbol{I} : I_1 \preceq I \preceq I_2\}$. *Let* P *be an extended generalized logic program, for* $r \in P$ *let* $H(r) :=$ *head of* r, $B(r) :=$ *body if* r, $[P] :=$ *set of all ground instantiations of rules of* P. *We introduce the following notions and sets:*

- $[I, J](F) \geq \frac{1}{2} =_{df}$ *for all* $K \in [I, J] : K(F) \geq \frac{1}{2}$
- $[I, J](F) = 1 =_{df}$ *for all* $K \in [I, J] : K(F) = 1$
- $P_{[I,J]} = \{r | r \in [P]$ *and* $[I, J](\wedge B(r)) \geq \frac{1}{2}\}$
- $\overline{P}_{[I,J]} = \{r | r \in [P]$ *and* $[I, J](\wedge B(r)) = 1\}$.

The following notion of a *stationary generated* is an essential refinement of the notion of a stable generated (two-valued) model which was introduced in [9].

Definition 7 (Stationary Generated Model) *Let* $P \subseteq \mathrm{EGLP}(\sigma)$ *an extended generalized logic program. Let* I *be an* AP*-model of* P. I *is a* stationary generated *model of* P*, symbolically* $I \in \mathrm{Mod}_{statg}(S)$*, if there is a sequence* $\{I_\alpha : \alpha < \kappa\}$ *of coherent interpretations satisfying following conditions:*

1. $I_0 = not\,\mathrm{OL}$ *(is the t-least interpretation).*
2. $I_\alpha \preceq I_{\alpha+1}$ *and* $I_\alpha \preceq I$ *for all* $\alpha < \kappa$.
3. $sup_{\alpha<\kappa} I_\alpha = I$.
4. $I_{\alpha+1} \in Min_{\preceq}\{J | I_\alpha \preceq J \preceq I$ *and*
   (a) for all $r \in \overline{P}_{[I_\alpha, I]}$ *it holds* $J(\vee H(r)) = 1$ *and*
   (b) for all $r \in P_{[I_\alpha, I]} : J(\vee H(r)) \geq \frac{1}{2}$ *or* $I(\neg \vee H(r)) = 1\}$.
5. $I_\lambda = sup_{\alpha<\lambda} I_\alpha$, λ *a limit ordinal.*

The sequence $\{I_\alpha : \alpha < \kappa\}$ is called a stationary AP-chain (or briefly a stationary chain) generating I. I is called a stationary generated Pr-model if I is a Pr-model, I_α are consistent interpretations, and in condition 4. the Pr-truth-relation $\models_{Pr}$ is used and if the condition (b) is replaced by $\forall r \in P_{[I_\alpha, I]} : I_{\alpha+1}(\vee H(r)) \geq \frac{1}{2}$. In this case $\{I_\alpha : \alpha < \kappa\}$ is called a stationary Pr-chain generating I.

By using proposition 2 one may prove that the set $\text{Min}_{\preceq}\{J \mid I_\alpha \preceq J \preceq I \;\&\; (a) \;\&\; (b)\}$ in condition 4. of definition 7 is non-empty. Hence, for every model $I \models P$, we may construct, according to definition 7, chains of interpretations satisfying the conditions 1.,2.,4. (but not neccessarily 3.). Such chains are called stationary chains in I. A stationary chain $\{I_\alpha \mid \alpha < \kappa\}$ in I is said to be maximal if for $K = sup\{I_\alpha\}$ we have $\text{Min}_{\preceq}\{J \mid K \preceq J \preceq I \;\&\; (a) \;\&\; (b)\} = \{K\}$, i.e. if K cannot be further extended. The set of stationary generated models of S is denoted by $\text{Mod}_{statg}(S)$. The associated entailment relation is defined as follows:

$$S \models_{statg} F \quad \text{iff} \quad \text{Mod}_{statg}(S) \subseteq \text{Mod}(F)$$

Notice that our definition of stationary generated models also accommodates

1. Negation in the head of a rule, such as in
   $$\Rightarrow\ not(nationality(x, German) \wedge nationality(x, US))$$
   expressing the integrity constraint that it is not possible to have both the German and the US nationality.
2. Nested negations, such as in $p(x) \wedge not(q(x) \wedge not\, r(x)) \Rightarrow s(x)$ which would be the result of folding $p(x) \wedge not\, ab(x) \Rightarrow s(x)$ and $q(x) \wedge not\, r(x) \Rightarrow ab(x)$.

We continue this section with the investigations of some fundamental properties of the introduced concepts. What can be said about the length of the stationary chains?

Proposition 5 *Let $P \subseteq EGLP$ and let I be a stationary generated model of P generated by the sequence $\{I_\alpha : \alpha < \kappa\}$, $\kappa \geq \omega$. Then there is an ordinal $\beta \leq \omega$ such that $I_\beta = I$ and $I_\beta = I_{\beta+1}$. We say that this sequence stabilizes at β.*

Definition 8 (Rank of a stationary generated model) *Given a program P and let I be a stationary generated AP- model of P. Let St(I)=$\{\alpha \mid$ there is a stationary chain for I stabilizing at $\alpha\}$. Then Rk (I) = infimum St(I) is called the rank of I.*

Corollary 6 *If M is a stationary generated AP-model of $P \subseteq$ EGLP, then there is either a finite P-stationary chain, or a P-stationary chain of length ω, generating M.*

Example 4 *Consider the following program $P = \{a \leftarrow not\, b;\ b \leftarrow not\, a;\ \neg a\}$. The following interpretation $\{\neg a,\ not\, a,\ b,\ not \neg b\} = K$ is a stationary generated AP-model of P. It is $I_0 = \{not\, a,\ not\, b,\ not \neg a,\ not \neg b\}$, and $P_{[I_0,K]} = \{b \leftarrow not\, a;\ \neg a\}$, and $\overline{P}_{[I_0,K]} = P_{[I_0,K]}$. Then K is a minimal extension of I_0 such that for all $r \in \overline{P}_{[I_0,K]}$ it holds that $K(H(r)) = 1$. Hence, $\{I_0, K\}$ is a stationary sequence generating K. Obviously, $Rk(K) = 1$.*

Example 5 *Consider the program* $P = \{a \vee b;\ \neg b \leftarrow a;\ a \leftarrow \ not\, a\}$*. The following interpretation is a stationary generated model of* P*:* $K = \{a,\ \neg b,\ \ not\, b,\ \ not\, \neg a\}$*.* $I_0 = \{\ not\, a,\ \ not\, \neg a,\ \ not\, b,\ \ not\, \neg b\}$*. Then* $P_{[I_0,K]} = \{a \vee b\} = \overline{P}_{[I_0,K]}$*. Then* $I_1 = \{a,\ \ not\, \neg a,\ \ not\, b,\ not\, \neg b\}$ *is a t-minimal extension of* I_0*. It is* $P_{[I_1,K]} = \{a \vee b,\ \neg b \leftarrow a\} = \overline{P}_{[I_1,K]}$*. Then* K *is a minimal t-extension of* I_1*, hence* $\{I_0, I_1, K\}$ *is a stationary sequence generating* K*. Obviously,* $Rk(K) = 2$*.*

The GWFSX-semantics can be introduced as follows.

Definition 9 (Generalized Wellfounded Semantics) *Let* P *be an extended generalized logic program and* $GWFSX(P) = \bigcap \mathrm{Mod}_{statg}(P) = \{l \mid l \in XG\ \&\ P \models_{statg} l\}$*. GWFSX(P) is said to be the generalized well-founded semantics of P.*

Let $QF = L^0(\ not, \neg, \wedge, \vee)$ be denote the set of quantifier-free sentences (not containing $\rightarrow$) and $C_{statg}(P) = \{F \mid P \models_{statg} F\}$. An inference operation $C : Pow(EGLP) \rightarrow Pow(QF)$ is said to be cumulative iff for every $X \subseteq C(P), X \subseteq QF$ holds $C(P \cup X) = C(P)$. Unfortunately, the operation C_{statg} is not cumulative on the set of all generalized logic programs, thus the important task remains to find natural cumulative approximations of C_{statg} (cf section 6).

5 Extended Normal Logic Programs and WFSX-semantics

In this section we present the result that for extended normal logic programs the $WFSX$-semantics models introduced in [12] coincides with the partial stationary generated semantics. To make the paper self-contained we recall the main notions. Let $P \subseteq$ ENLP a normal logic program, i.e. the rules r have the form: $r := a_1, \ldots, a_m,\ not\, b_1, \ldots,\ not\, b_n \Rightarrow c$, where a_i, b_j, c are objective literals. Such a rule is denoted in the following also by $c \leftarrow a_1, \ldots, a_m,\ not\, b_1, \ldots,\ not\, b_n$. Let $I \subseteq \mathrm{OL} \cup\ not\,\mathrm{OL}$ be a (consistent) partial interpretation.

Definition 10 *Let* I *be an interpretation and* P *a (instantiated) program,* $r \in P$*. The* I*-transformation of* r*, denoted by* $tr_I(r)$*, is defined as follows:*

- *if* $not\, l \in B(r)$ *and* $l \in I$ *then* $not\, l$ *is replaced by* $tr_I(\ not\, l) = \mathbf{f}$*;*
- *if* $l \in B(r)$*,* l *objective and* $\neg l \in I$*, then* l *is replaced by* $tr_I(l) = \mathbf{f}$*;*
- *if* $l \in B(r)$*,* l *objectiv and* $\neg l \notin I$ *then* $tr_I(l) = l$*;*
- *if* $not\, l \in I$ *then* $tr_I(\ not\, l) = \mathbf{t}$*;*
- *the remaining default literals* $not\, l$ *are replaced by* $tr_I(\ not\, l) = \mathbf{u}$*.*

Let be $tr_I(B(r)) = \{tr_I(l) | l \in B(r)\}$*,* $tr_I(r) = H(r) \leftarrow tr_I(B(r))$*, and* $tr_I(P) = \{tr_I(r) | r \in P\}$*. Obviously,* $tr_I(P)$ *is a non-negative logic program, also denoted by* P/I*.*

We now investigate a semi-constructive description of a special AP-model of a non-negative logic program; note that there are consistent non-negative logic programs without a t-least AP-model.

Definition 11 *Let P be a instantiated non-negative normal program. The operator $\Gamma_P : \boldsymbol{I} \to \boldsymbol{I}$ is defined as follows:*
$\Gamma_P(I) = \{l \mid$ *there is a rule* $l \leftarrow B(r) \in [P]$ *such that* $I(\wedge B(r)) = 1\} \cup \{\, not l \mid$ *for every rule* $r \in [P]$ *satisfying* $H(r) = l$ *it is* $I(\wedge B(r)) = 0\}$.

The operator Γ_P is monotonic with respect to the truth-ordering $\preceq$. We construct a sequence $\{I_n\}_{n<\omega}$ of interpretations as follows. Let $I_0 = \{\, not a \mid a \in \mathrm{OL}\}$, i.e. I_0 is the t-least interpretation (it is the least element in the semi-lattice $\mathcal{C} = (\boldsymbol{I}, \preceq)$), and $I_{n+1} = \Gamma_P(I_n)$. Obviously, $I_n \preceq I_{n+1}$, for $n < \omega$. Let be $I_\omega = \Gamma_P^\omega = sup\{I_n : n < \omega\}$. Define $Coh(I) = \{\, not \neg l | l \in Pos(I)\} \cup I$, $Coh(I)$ is called the coherency-closure of I.

Proposition 7 *Let P be a non-negative normal logic program and I_ω defined as above. If $Coh(I_\omega) = K$ is a consistent interpretation then K is a AP-model of P.*

Proof: Obviously, K is coherent. Let $l \leftarrow B(r) \in [P]$. We have to show that $K(l \leftarrow B(r)) = 1$, by definition: $K(B(r)) \leq K(l)$ or ($K(\neg l) = 1$ and $K(B(r)) = \frac{1}{2}$). We may assume that $K(B(r)) \geq \frac{1}{2}$ (the case $K(B(r)) = 0$ is trivial.)
1) $K(B(r)) = \frac{1}{2}$. If $K(H(r)) \geq \frac{1}{2}$ we are ready. Assume that for $l = H(r)$ we have $K(l) = 0$, i.e. $not l \in K$. By the definition of the sequence $\{I_n\}_{n<\omega}$ follows that $not l \notin I_\omega$. This implies $\neg l \in I_\omega$ and this verifies $K(r) = 1$.
2) $K(B(r)) = 1$, $B(r) = l_1, \ldots, l_m$. We may assume that $\{l_1, \ldots, l_m\} \subseteq \mathrm{OL}$. By lemma 1 there is a k such that for all $n > k$ we have $\{l_1, \ldots, l_m\} \subseteq I_n$, hence $l \in I_\omega$ and this implies $K(r) = 1$. $\Box$

Remark: If $Coh(I_\omega) = K$ is consistent then K is a t-minimal AP-model of P. In general, K is no Pr-model.

Example 6 *Let $P = \{a \leftarrow b;\ \neg a;\ b \leftarrow \mathbf{u}\}$. Then $Coh(I_\omega) = \{\neg a,\ not a,\ not \neg b\}$ is a AP-model, but P has no coherent Pr-model.*

The interpretations $I_1, I_2, \ldots, I_n, \ldots$ are (in general) not coherent. We define a new sequence $J_n = I_n \cup Ch(I_\omega)$, where $Ch(I_\omega) = \{\, not \neg l | l \in Pos(I)\}$. Obviously, the following conditions are satisfied

- $J_1 \preceq J_2 \preceq \ldots \preceq J_n \preceq \ldots$
- every J_n is coherent
- $I_\omega = sup_{n<\omega} J_n$.

$\{I_n\}_{n<\omega}$ is said to be the *standard sequence with respect to* P, and $\{J_n\}_{n<\omega}$ is called the *coherent standard sequence with respect to P*. We show now that the stationary generated models coincides with the partial stable models in the sense of [1].

Definition 12 *[1] Let be $P \subseteq ENLP$. A coherent interpretation I is a partial stable model of P if $I = Coh(\Gamma_{P/I}^\omega(I_0))$.*

We use the following technical lemma to prove the main result in this section.

Lemma 2. *Let P be a normal extended logic program, I an interpretation. Let be $\{I_n\}_{n<\omega}$ resp. $\{J_n\}_{n<\omega}$ be the standard resp. coherent standard sequence with respect to P/I, and $l \leftarrow B(r) \in [P]$. Then the following conditions are equivalent:*
(1) $I_n(\wedge tr_I(B(r)) \geq \frac{1}{2}$;
(2) $[J_n, I](\wedge B(r)) \geq \frac{1}{2}$ (i.e. $r \in P_{[J_n,I]}$).

Remark: Lemma 2 remains true if in condition 1) and 2) the relation "$\geq \frac{1}{2}$" is replaced by "=1".

Proposition 8 *Let P be an extended normal logic program and I a model of P. Then I is a stationary generated AP-model if and only if I is a partial stable model of P.*[5]

Sketch of the Proof: We sketch only one direction. Let I be a partial stable model. By definition we have $I = Coh(\Gamma^{\omega}_{P/I}(I_0))$, I_0 is the t-least interpretation. Let $I_0 \preceq I_1 \preceq \ldots \preceq I_n \preceq$ be the standard sequence associated to $\Gamma_{P/I}$ and $I_\omega = \Gamma^{\omega}_{P/I}(I_0)$. Let $J_0 \preceq J_1 \preceq \ldots J_n \preceq$ be the coherent standard sequence defined by $J_n = I_n \cup Coh(Pos(I_\omega))$. We show that $\{J_n | n < \omega\}$ is a stationary chain. One has to show that for every $n < \omega$ the interpretation J_{n+1} is a minimal t-extension of J_n satisfying the following two condition of stationary generatedness:

1. $\forall r(r \in \overline{P}_{[J_n,I]} \rightarrow J_{n+1}(H(r)) = 1)$ and
2. $\forall r(r \in P_{[J_n,I]} \rightarrow J_{n+1}(H(r)) \geq \frac{1}{2}$ or $I(\neg H(r)) = 1)$.

By definition is
$(*) : J_{n+1} = \{l | l \leftarrow B(r) \in [P/I] \wedge I_n(\wedge tr(B(r)) = 1\} \cup$
$\{ notl | \forall(l \leftarrow B(r)) \in [tr_I(P)] : I_n(\wedge B(r)) = 0\} \cup Coh(Pos(I_\omega))$.

1. Let $l \leftarrow B(r) \in \overline{P}_{[J_n,I]}$, then $[J_n, I](B(r)) = 1$. Let $B(r) = l_1, \ldots, l_s,\ notm_1, \ldots,\ notm_t$. By 1. it is $I(\wedge B(r)) = 1$, hence $tr_I(\ notm_j) = 1$ which yields $I_n(tr_I(B(r))) = 1$, by definition $(*)$ it is $l \in I_{n+1}$ and this implies $l \in J_{n+1}$, hence $J_{n+1}(l) = 1$, $l = H(r)$.

2. Let $l \leftarrow B(r) \in P_{[J_n,I]}$. We have to show that the condition
$(**)$: $(J_{n+1}(l) \geq \frac{1}{2} \vee I(\neg l) = 1)$
is satisfied. Since $[J_n, I](B(r)) \geq \frac{1}{2}$ this implies $I_n(tr(B(r))) \geq \frac{1}{2}$ for all $r \in P_{[J_n,I]}$. Assume there is a rule $l \leftarrow B(r) \in P_{[J_n,I]}$ such that $(**)$ is not satisfied. Then $J_{n+1} = 0$ and $I(\neg L) < 1$. Then $notl \in Neg(\Gamma_P(I_n))$ (if $notl \notin Neg(\Gamma_P(I_n))$ this would imply $notl \in Coh(Pos(I_\omega))$, which is impossible since $I(\neg l) < 1$). $notl \in Neg(\Gamma_P(I_n))$ implies for all $l \leftarrow B(s) \in [P/I]$ the condition $I_n(B(s)) = 0$. But, there is one rule $l \leftarrow B(r) \in P_{[J_n,I]}$. i.e. $[J_n, I](B(r)) \geq \frac{1}{2}$, $B(r) = l_1, \ldots, l_s,\ notm_1, \ldots,\ notm_t$. It is sufficient to show that $I_n(tr(B(r))) \geq \frac{1}{2}$ which gives a contradiction. The condition $I_n(tr(B(r))) \geq \frac{1}{2}$ follows from lemma 2.

We finally show that for all J such that $J_n \preceq J \preceq J_{n+1}$ satisfying the conditions 1. and 2. it follows $J = J_{n+1}$ (i.e. J_{n+1} is a minimal extension satisfying 1. and

[5] A similar proposition may be proved for Pr-models of P.

2.). Assume that J satisfies 1. and 2. From 1. follows that $Pos(J_{n+1}) \subseteq Pos(J)$, hence $Pos(J_{n+1}) = Pos(J)$. It remains to show that $Neg(J_{n+1}) = Neg(J)$ and since $Neg(J_{n+1} \subseteq Neg(J)$ it is sufficient to show that $Neg(J) \subseteq Neg(J_{n+1})$. From 2. follows for all $r \in P_{[J_n,I]}$ the condition $(J(H(r)) \geq \frac{1}{2} \vee I(\neg H(r)) = 1)$. Assume, by contradiction $Neg(J) \not\subseteq Neg(J_{n+1})$. Then there is a default literal $not l \in Neg(J)$ such that $not l \notin Neg(J_{n+1})$. Obviously, $not l \notin Coh(I_\omega)$, otherwise we would have $not l \in Neg(J_{n+1})$ since $Coh(I_\omega)$ is contained in any $Neg(J_n)$. The condition $not l \notin Coh(I_\omega)$ implies $I(\neg l) \neq 1$. By assumption $J(l) = 0$; also $not l \notin Neg(I_{n+1})$, and by definition of I_{n+1} there is a rule $r \in tr_I(P)$ such that $I_n(B(r)) \geq \frac{1}{2}$ and $H(r) := l$. Let $tr_I(B(s)) = B(r)$. Then, by lemma 2 we have $[J_n, I](B(s)) \geq \frac{1}{2}$; hence $s \in P_{[J_n,I]]}$. Since J satisfies the condition 1. and $I(\neg l) \neq 1$ this implies $J(l) \geq \frac{1}{2}$, which yields a contradiction.□

Let be $P \subseteq ENLP$ and I be a coherent and consistent set of extended literals. We may test whether I is partial stable AP-Model or a partial stable Pr-model by using proposition 8. Let be $\kappa \leq \omega$ and $\{I_n : n < \kappa\}$ a sequence of interpretations satisfying the conditions 1., 2. , and 4. of definition 7 with respect to I. Such a sequence is said to be a successful AP-sequence for (P, I) if $I = sup_{n<\kappa} I_n$ and one of the following conditions is satisfied:

(1) $k = \omega$ and $\{I_n : n < \kappa\}$ does not stabilize at any $m < \kappa$;

(2) $\kappa < \omega$ and $\{I_n : n < \kappa\}$ stabilizes at a number $m < \kappa$.

From proposition 8 and corollary 6 follows:

Proposition 9 *Let P be an extended normal logic program and I be a coherent and consistent interpretation. I is a stationary generated AP-model of P if and only if there exists a successful AP-sequence for (P,I).*

An analogous proposition holds for stationary generated Pr-models.

Example 7 *Let be* $P = \{c \leftarrow a;\ a \leftarrow b;\ b \leftarrow\ not b;\ \neg a\}$ *and*
$I = \{\neg a,\ not a,\ not c,\ not \neg b,\ not \neg c\}$.

Then there exists a successful AP-sequence for (P,I):
Take $I_0 = \{ not a,\ not \neg a,\ not c,\ not \neg c,\ not b,\ not \neg b\}$.
Then $\overline{P}_{[I_0,I]} = \{\neg a\}$ *and* $P_{[I_0,I]} = \{\neg a,\ b \leftarrow\ not b\}$. *Take* $I_1 := I$, *then* I_1 *is a minimal t-extension of* I_0 *such that* $I_1(\neg a) = 1$ *and* $I_1(b) \geq \frac{1}{2}$.

Now consider $\overline{P}_{[I_1,I]} = \{\neg a\}$, *and* $P_{[I_1,I]} = \{\neg a,\ a \leftarrow b,\ b \leftarrow\ not b\}$. *There is a minimal t-extension* I_2 *of* I_1, *namely* $I_2 = I_1$, *such that* $I_2(\neg a) = 1$, $I_2(b) \geq \frac{1}{2}$ *and* $(I_2(a) \geq \frac{1}{2}\ \vee\ I(\neg a) = 1)$. *Hence, the sequence* $\{I_0, I_1, I_2\}$ *is a successful AP-sequence for* (P, I) *of length 2 which stabilizes at 1; it is* $I_1 = I_2 = I$.

It is easy to see that there is no successful Pr-sequence for (P, I).

6 Conclusion

By introducing a new general definition of stationary generated models, we have sketched the idea of a *stationary model theory* for logic programs. The conceptual tools presented may be useful for a systematic study of partial models of extended logic programs. One interesting invariant of a model I of P is the set $StatC(I)$ of its stationary chains constructed according to definition 7. Evidently, I is stationary generated if there is a chain in $StatC(I)$ reaching I. Furthermore, it seems to be possible to analyze further extensions of normal logic programs, such as quantifiers in bodies and heads of rules. An interesting task is to find natural cumulative approximations of the inference operation C_{statg}. An inference operation $C : Pow(EGLP) \rightarrow Pow(QF)$ is a cumulative approximation of C_{stat} iff the following conditions are fulfilled:

1. For all $P \subseteq EGLP$ it is $C(P) \subseteq C_{statg}(P)$;
2. for extended normal logic programs P it is $C(P) = C_{statg}(P)$;
3. C is cumulative for arbitrary extended generalized logic programs P, i.e. for for all $X \subseteq C(P) \cap QF$ holds $C(P) = C(P \cup X)$.

Recently, several kinds of semantics were studied beyond the class of normal logic programs. Gelfond and Lifschitz [8], and Przymusinski [13] expand the stable model semantics to the class of disjunctive logic programs admitting two kinds of negation, classical negation $\neg$ and default negation *not*. Their semantics do not assume coherency, a condition that we consider as very natural.

Super logic programs were introduced in [16], they present a proper subclass of generalized logic programs. The semantics is based on the notion of minimal belief operator which is a part of the *Autoepistemic Logic of Beliefs*, [15]. This approach does not include strong negation and the coherency principle.

Brass and Dix investigate in [3] the so-called D-WFS-semantics for normal disjunctive logic programs. This semantics is defined by an abstract inference operation satisfying certain structural properties of proof-theoretical type. The D-WFS semantics does not determine a proper model theory, furthermore it is not clear how to expand it to generalized logic programs.

Lifschitz, Tang and Turner propose in [10] a semantics for logic programs allowing nested expressions in the heads and the body of the rules. The syntax of these programs is similar to ours, but the semantics differs.

Pearce presents in [11] an elegant characterization of the non-monotonic inference relation associated with the stable model semantics by using intuitionistic logic.

7 Acknowledgements

The support of the German INIDA and the Portuguese ICCTI organizations is thankfully acknowledged, as well as the Praxis programme MENTAL project 2/2.1/TIT/1593/95. Thanks due to the anonymous referees for their criticism and useful comments.

References

1. J.J.Alferes, L.M.Pereira. Reasoning with Logic Programming, LNAI vol. 1111, 1996, Springer Lecture Notes in Artificial Intelligence

2. J. J. Alferes, H. Herre, L.M. Pereira. Partial Models of Extended Generalized Logic Programs, Report Nr. (2000), University of Leipzig
3. S. Brass and J. Dix. Characterizing D-WFS. Confluence and Iterated GCWA. *Logics in Artificial Intelligence, Jelia'96*, p.268–283,1996, Srpinger LNCS 1121
4. J. Engelfriet and H. Herre. Generated Preferred Models and Extensions of Nonmonotonic Systems; *Logic Programming*, p. 85-99, Proc. of the 1997 International Symposium on LP, J. Maluszinski (ed.), The MIT Press, 1997
5. J. Engelfriet and H. Herre. Stable Generated Models, Temporal Logic and Disjunctive Defaults, *Journal of Logic Programming* 41: 1-25 (1999)
6. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R. A. Kowalski and K. A. Bowen, editors, *Proc. of ICLP*, pages 1070–1080. MIT Press, 1988.
7. M. Gelfond and V. Lifschitz. Logic programs with classical negation. *Proc. of ICLP'90*, pages 579–597. MIT Press, 1990.
8. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
9. H. Herre and G. Wagner. Stable Models are Generated by a Stable Chain. *Journal of Logic Programming*, 30 (2): 165-177, 1997
10. V. Lifschitz, L.R. Tang and H. Turner. Nested Expressions in Logic Programs, *Annals of Mathematics and A.I.*, to appear (1999)
11. D. Pearce. Stable inference as intuitionistic validity, *Journal of Logic Programming* 38, 79-91 (1999)
12. L.M.Pereira, J.J.Alferes. Wellfounded semantics for logic programs with explicit negation. In B. Neumann:(ed.) European Conference on AI, 102-106, Wien, 1992
13. T.C. Przymusinski. Stable semantics for disjunctive programs. *New Generation Computing*, 9:401–424, 1991.
14. T.C. Przymusinski. Autoepsitemic Logic of Knolwedge and Belief. *Proc. of the 12th Nat. Conf. on A.I.*, AAAI'94, 952-959
15. T.C. Przymusinski. Well-founded and Stationary Models of Logic Programs; *Annals of Mathematics and Artificial Intelligence*, 12 (1994), 141-187
16. T.C. Przymusinski. Super Logic Programs and Negation as Belief. In: R. Dyckhoff, H. Herre, P. Schroeder-Heister, editors, *Proc. of the 5th Int. Workshop on Extensions of Logic Programming*, Springer LNAI 1050, 229–236