Stationary Generated Models of Generalized Logic Programs

Heinrich Herre[1,2] and Axel Hummel[1,2]

[1] Department of Computer Science, Faculty of Mathematics and Computer Science, University of Leipzig, Johannisgasse 26, 04103 Leipzig, Germany, heinrich.herre@imise.uni-leipzig.de, hummel@informatik.uni-leipzig.de
[2] Institute of Medical Informatics, Statistics and Epidemiology, University of Leipzig, Härtelstrasse 16-18, 04107 Leipzig, Germany

Abstract. The interest in extensions of the logic programming paradigm beyond the class of normal logic programs is motivated by the need of an adequate representation and processing of knowledge. One of the most difficult problems 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 generalized logic programs which is a conservative extension of the stationary semantics for normal logic programs of [Prz91]. The presented preference criterion defines a partial model of a generalized logic program as intended if it is generated by a stationary chain. It turns out that the stationary generated models coincide with the stationary models on the class of normal logic programs. The general wellfounded semantics of such a program is defined as the set-theoretical intersection of its stationary generated models. For normal logic programs the general wellfounded semantics equals the wellfounded semantics.

Keywords: Extensions of logic programs, semantics, knowledge representation

1 Introduction

Declarative semantics provides a mathematical precise definition of the meaning of a program in a manner, which is independent of procedural considerations, context-free, and easy to manipulate, exchange and reason about. Finding a suitable declarative or intended semantics is an important and difficult problem in logic programming and deductive data bases. Logic programs and deductive data bases must be as easy to write and comprehend as possible and as close to natural discourse as possible. Research in the area of logic programming and non-monotonic reasoning made a significant contribution towards the better understanding of relations existing between various formalizations of non-monotonic reasoning and the discovery of deeper underlying principles of non-monotonic reasoning and logic programming. Standard logic programs are not sufficiently expressive for the representation of large classes of knowledge bases. In particular, the inability of logic programs to deal with arbitrary open formulas is an obstacle to use logic programming as a declarative specification language for software engineering and knowledge representation. Formalisms admitting more complex formulas are more expressive and natural to use since they permit in many cases easier translation from natural language expressions and from informal specifications. The additional expressive power of generalized logic programs significantly simplifies the problem of translation of non-monotonic formalisms into logic programs, and, consequently facilitates using logic programming as an inference engine for non-monotonic reasoning.

A set of facts can be viewed as a database whose semantics is determined by its minimal models. In the case of logic programs, minimal models are not adequate because they are not able to capture the directedness of rules, i.e. they do not satisfy the *groundedness* requirement. Therefore, *stable* models in the form of certain fixpoints have been proposed by Gelfond and Lifschitz [GL88] as the intended models of normal logic programs. 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 remain (in a certain sense) applicable throughout the model computation. The notion of a stationary model of a normal logic program was introduced in [Prz91] and further elaborated in [Prz94]. Stationary generated models - as expounded in the current paper - are defined in a different way. This notion can be easily extended to generalized logic programs which include several types of programs as special cases, among them disjunctive programs [Prz91] and super-logic programs [Prz96]. Lifschitz, Tang and Turner propose in [LTT99] a semantics for logic programs allowing for nested expressions in the heads and the body of the rules. The syntax is similar to our generalized logic programs, but the semantics differs.

In [AHP00] the notion of stationary generated AP-models was introduced. This notion differs from the stationary generated models as defined in the present paper. Stationary generated AP-models and stationary generated models are based on different truth-relations for three-valued partial models. Hence, the current paper closes a gap that remained open in [AHP00].

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, and then, in section 5 we investigate the relationship of this general concept to the original fixpoint-based definitions for normal programs as in [Prz91]. It turns out that for normal programs the stationary generated models coincide with the stationary models in the sense of [Prz91]. This fact motivates the introduction of the notion of a general well-founded semantics for a generalized logic program which is defined as the set-theoretical intersection of its stationary generated models. We believe that the notion of general well-founded semantic is the most natural generalization of well-founded semantics to generalized logic programs.

2 Preliminaries

A *signature* $\sigma = \langle Rel, Const, Fun \rangle$ consists of a set *Rel* of relation symbols, a set *Const* of constant symbols, and a set *Fun* of function symbols. U_σ denotes the set of all ground terms of σ. For a tuple $t_1, \ldots, t_n$ we will also write $\bar{t}$ when its length is of no relevance. The logical functors are $not, \wedge, \vee, \rightarrow, \forall, \exists$. $L(\sigma)$ is the smallest set containing the atomic formulas of σ, and being closed with respect to the following conditions: if $F, G \in L(\sigma)$, then $\{not F,\ F \wedge G,\ F \vee G,\ F \rightarrow G,\ \exists x F,\ \forall x F\} \subseteq L(\sigma)$.

$L^0(\sigma)$ denotes the corresponding set of sentences (closed formulas). For sublanguages of $L(\sigma)$ formed by means of a subset $\mathcal{F}$ of the logical functors, we write $L(\sigma; \mathcal{F})$. With respect to a signature σ we define the following sublanguages: $\mathrm{At}(\sigma) = L(\sigma; \emptyset)$, the set of all atomic formulas (also called *atoms*). The set $\mathrm{GAt}(\sigma)$ of all ground atoms over σ is defined as $\mathrm{GAt}(\sigma) = \mathrm{At}(\sigma) \cap \mathrm{L}^0(\sigma)$. $\mathrm{Lit}(\sigma) = L(\sigma;\ not)$, the set of all *literals*; for a set X of formulas let $\overline{X} = \{not F \mid F \in X\}$. Then, the set of all ground literals over σ is defined as $\mathrm{GAt}(\sigma) \cup \overline{\mathrm{GAt}(\sigma)}$. $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ exhibit particular ground atoms with the meaning true (value 1), false (value 0), undetermined (value $\frac{1}{2}$).

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 $L(\sigma)$. Let $L^*(\sigma) = L(\sigma;\ not, \wedge, \vee, \forall, \exists)$, and $PL(\sigma) = L^*(\sigma) \cup \{F \rightarrow G | F, G \in L^*(\sigma)\}$. $PL(\sigma)$ is called the set of program formulas of signature σ. If Y is a partially ordered set, then $\mathrm{Min}(Y)$ denotes the set of all minimal elements of Y, i.e. $\mathrm{Min}(Y) = \{a \in Y \mid \neg\exists a' \in Y : a' < a\}$. A *Herbrand interpretation* of the language $L(\sigma)$ is one for which the universe equals U_σ, and the function symbols are interpreted canonically.

Definition 1 (Partial Herbrand Interpretation) *Let be $\sigma = \langle Rel, Const, Fun\rangle$ a signature. A partial Herbrand interpretation I of σ is defined as follows $I = (U(\sigma), (f^I)_{f\in Fun}, (r^I)_{r\in Rel})$. Its* universe $U(\sigma)$ *is equal to the set of all ground terms U_σ; its canonical interpretation of ground terms is the identity mapping. The relation symbols $r \in Rel(\sigma)$ are interpreted by functions r^I defined by $r^I : U^{a(r)} \longrightarrow \{0, \frac{1}{2}, 1\}$ for every relation symbol $r \in Rel$, where $a(r)$ denotes the arity of r. Obviously, every Herbrand interpretation is determined by a function $i_I : \mathrm{At}(\sigma) \longrightarrow \{0, \frac{1}{2}, 1\}$.*

A partial Herbrand σ-interpretation I can be represented as a set of ground literals $I \subseteq \mathrm{GAt}(\sigma) \cup \overline{\mathrm{GAt}(\sigma)}$ such that there is no ground atom $a \in \mathrm{GAt}$ satisfying $\{a, not\, a\} \subseteq I$. For a partial Herbrand σ-interpretation I let $Pos(I) = I \cap \mathrm{GAt}$ and $Neg(I) = I \cap \overline{\mathrm{GAt}}$. A partial Herbrand interpretation I is two-valued (or total) if for every $a \in GAt$ holds $\{a, not\, a\} \cap I \neq \emptyset$. I is a partial interpretation over $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ if $\{\mathbf{u}, not\,\mathbf{u}\} \cap I = \emptyset$, and $\{\mathbf{t}, not\,\mathbf{f}\} \subseteq I$.

The class of all partial Herbrand σ-interpretations is denoted by $\boldsymbol{I}_H(\sigma)$. In the sequel we shall also simply say 'interpretation' instead of 'partial Herbrand 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 $\nu(F)$. Note that for a constant c, being a 0-ary function, we have $\nu(c) = c$. The model relation $\models \subseteq \boldsymbol{I}_H(\sigma) \times L^0(\sigma)$ between an interpretation and a sentence is defined inductively as follows.

Definition 2 (Model Relation) *Let $I \in \boldsymbol{I}_H(\sigma)$. Then the mapping i_I can be extended to a function $\tilde{I}$ from the set of all sentences from $PL(\sigma)$ into $\{0, \frac{1}{2}, 1\}$.*

1. $\tilde{I}(a) = i_I(a)$ *for atomic sentences a.*
2. $\tilde{I}(not\, F) = 1 - \tilde{I}(F)$.
3. $\tilde{I}(F \wedge G) = min\{\tilde{I}(F), \tilde{I}(G)\}$.
4. $\tilde{I}(F \vee G) = max\{\tilde{I}(F), \tilde{I}(G)\}$.

5. $\widetilde{I}(F \to G) = 1$ *if* $\widetilde{I}(F) \leq \widetilde{I}(G)$.
6. $\widetilde{I}(F \to G) = 0$ *if* $\widetilde{I}(F) \not\leq \widetilde{I}(G)$.
7. $\widetilde{I}(\exists x F(x)) = sup\{\widetilde{I}(F(x/t)) \mid t \in U(\sigma)\}$.
8. $\widetilde{I}(\forall x F(x)) = inf\{\widetilde{I}(F(x/t)) \mid t \in U(\sigma)\}$.

We write $I \models F \iff \widetilde{I}(F) = 1$ *for sentences* F *and for arbitrary formulas* F*:*
$I \models F \iff I \models F\nu$ *for all* $\nu : Var \to U_\sigma$. I *is called a model of* F*, and for sets* X *of formulas* $I \models X$ *if and only if for all* $F \in X$ *it holds* $I \models F$. *To simplify the notation we don't distinguish between* I *and* $\widetilde{I}$ *in the following. Two formulas* $F, G \in L(\sigma)$ *are said to be logical equivalent iff for every instantiation* ν *and every partial interpretation* I *the condition* $I(F\nu) = I(G\nu)$ *is satisfied.*

$\mathrm{Mod}_H(X) = \{I \in \boldsymbol{I}_H : I \models X\}$ denotes the Herbrand model operator, and $\models_H$ denotes the corresponding consequence relation, i.e. $X \models_H F$ iff $\mathrm{Mod}_H(X) \subseteq \mathrm{Mod}_H(F)$. In the following we omit the subscript H.

Let $L^{\wedge,\vee}(\mathrm{Lit}(\sigma))$ the smallest subset of $L(\sigma)$ containing the set $\mathrm{Lit}(\sigma)$ and closed with respect to the connectives $\wedge, \vee$.

Proposition 1 *[HJW95] For every formula* $F \in L_1(\sigma)$ *there is a formula* $G \in L^{\wedge,\vee}(\mathrm{Lit}(\sigma))$ *such that* F *and* G *are logical equivalent.*

Definition 3 (Partial Orderings between Interpretations) *Let be* $I, I_1 \in \boldsymbol{I}_H$ *two interpretations. We define the following orderings between* I *and* I_1.

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 *iff* $I \subseteq I_1$. *The partial ordering* $\subseteq$ *between Herbrand interpretations is called information-ordering.* I_1 *is said to be an information-extension (briefly i-extension) of* I.
3. *Let* I, I_1 *be two-valued Herbrand interpretations. Define* $I \leq I_1$ *if and only if* $Pos(I) \subseteq Pos(I_1)$.

Obviously, if I, I_1 *are two-valued models then* $I \preceq I_1$ *iff* $Pos(I) \subseteq Pos(I_1)$.

Proposition 2 *The system* $\mathcal{C} = (\boldsymbol{I}_H, \preceq)$ *of consistent partial interpretations is a complete lattice.*

Proof: Let $\Omega \subseteq \boldsymbol{I}_H$ be an arbitrary non-empty subset. Define $I =_{df} \bigcup\{Pos(K) \mid K \in \Omega\} \cup \bigcap\{Neg(K) \mid K \in \Omega\}$, then I is the least upper bound of Ω, i.e. $I = sup\Omega$. Analogously, the infimum of Ω, denoted by $inf\Omega$ is defined by $inf\Omega = \bigcap\{Pos(K) \mid K \in \Omega\} \cup \bigcup\{Neg(K) \mid K \in \Omega\}$. □

3 Sequents and Programs

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

Definition 4 (Sequent) *A* sequent $s = F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$ *is an expression where* $F_i, G_j \in L(\sigma, \{\wedge, \vee, not\})$ *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* $\{G_1, \ldots, G_n\}$. $\mathrm{Seq}(\sigma)$ *denotes the class of all sequents* s *such that* $H(s), B(s) \subseteq L(\sigma)$*, and for a given set* $S \subseteq \mathrm{Seq}(\sigma)$*,* $[S]$ *denotes the set of all ground instances of sequences from* S*.*

Definition 5 (Model of a Sequent) *Let* $I \in \boldsymbol{I}_H$*. Then we define,* $I \models F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$ *iff for all ground substitutions the following condition is satisfied:* $I \models \bigwedge_{i \leq m} F_i\nu \rightarrow \bigvee_{j \leq n} G_j\nu$*.* I *is said to be a model of* $F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$*.*

We define the following classes of sequents corresponding to non-negative, positive disjunctive, normal, normal disjunctive, and generalized logic programs, respectively.

1. $\mathrm{PLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) : H(s) \in \mathrm{At}(\sigma),\ B(s) \subseteq \mathrm{At}(\sigma) \cup \{\mathbf{u}, \mathbf{t}, \mathbf{f}\}$.
2. $\mathrm{PDLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) : B(s), H(s) \subseteq \mathrm{At}(\sigma), H(s) \neq \emptyset\}$.
3. $\mathrm{NLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) : H(s) \in \mathrm{At}(\sigma),\ B(s) \subseteq \mathrm{Lit}(\sigma)\}$.
4. $\mathrm{NDLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) : H(s) \subseteq \mathrm{At}(\sigma),\ B(s) \subseteq \mathrm{Lit}(\sigma), H(s) \neq \emptyset\}$.
5. $\mathrm{GLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) : H(s), B(s) \subseteq L(\sigma;\ not, \wedge, \vee)\}$.

Subsets of PLP are called *non-negative* logic programs, programs associated to PDLP are called *positive disjunctive* logic programs. NLP relates to *normal* logic programs, NDLP to *normal disjunctive* logic programs, and GLP to *generalized* logic programs.

Lemma 3 *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*

$F \in L(\wedge, \vee, not) \cup \{G \rightarrow H \mid F, G \in L(\wedge, \vee, not)\}$. *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* $F \in L(\wedge, \vee, not) \cup \{G \rightarrow H \mid F, G \in L(\wedge, \vee, not)\}$. *Then there exists a number* k *such that for all* $s > k$ *the condition* $J(F) = J_s(F)$ *is satisfied.*

Let X be an interpretation and $P \subseteq \mathrm{GLP}$. X is said to be upward-consistent with respect to P if there is a model $I \models P$ such that $X \preceq I$.

Proposition 4 *Let* $P \subseteq \mathrm{GLP}$ *and* K *an interpretation being upward-consistent with respect to* P. *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* $J_1 \in \boldsymbol{I}_H$ *the conditions* $K \preceq J_1 \preceq J$ *and* $J_1 \models P$ *imply* $J = J_1$.

Corollary 5 *Let* $P \subseteq \mathrm{GLP}$. *Every partial model of* P *is an t-extension of a t-minimal partial model and can be t-extended to a t-maximal partial model of* P.

Proposition 6 *Every non-negative logic program has a t-least partial model.*

4 Stationary Generated Models

Definition 6 (Truth Interval of Interpretations) *Let* $I_1, I_2 \in \boldsymbol{I}_H$. *Then,* $[I_1, I_2] = \{I \in \boldsymbol{I}_H : I_1 \preceq I \preceq I_2\}$. *Let* $P \subseteq \mathrm{GLP}$ *and let* F *be a sentence. We introduce the following notions.*

- $[I, J](F) = inf\{K(F) | K \in [I, J]\}$
- $\underline{P}_{[I,J]} = \{r \mid r \in [P]$ *and* $[I, J](B(r)) \geq \frac{1}{2}\}$
- $\overline{P}_{[I,J]} = \{r \mid r \in [P]$ *and* $[I, J](B(r)) = 1\}$

The following notion of a *stationary generated* or *stable generated partial model* is a refinement of the notion of a stable generated (two-valued) model which was introduced in [HW97].

Definition 7 (Stationary Generated Model) *Let be* $P \subseteq \mathrm{GLP}$. *A model* I *of* P *is called* stationary generated *or partial stable generated if there is a sequence* $\{I_\alpha : \alpha < \kappa\}$ *of interpretations satisfying the following conditions:*

1. $I_0 = \overline{GAt}$ *(is the t-least interpretation)*
2. $\alpha < \beta < \kappa$ *implies* $I_\alpha \preceq I_\beta$
3. $sup_{\alpha<\kappa} I_\alpha = I$
4. *For all* $\alpha < \kappa$: $I_{\alpha+1} \in Min_{tm}\{J \mid I_\alpha \preceq J \preceq I$ *and (a) for all* $r \in \overline{P}_{[I_\alpha,I]}$ *it holds* $I_{\alpha+1}(H(r)) = 1$ *and (b) for all* $r \in P_{[I_\alpha,I]} : I_{\alpha+1}(H(r)) \geq \frac{1}{2}\}$.
5. $I_\lambda = sup_{\beta<\lambda} I_\beta$ *for every limit ordinal* $\lambda < \kappa$.

We also say that I *is* generated *by the* P-stationary chain $\{I_\alpha \mid \alpha < \kappa\}$.

The set of all stationary generated models of P is denoted by $\mathrm{Mod}_{statg}(P)$. The resp. stationary generated entailment relations are defined as follows: $P \models_{statg} F$ iff $\mathrm{Mod}_{statg}(P) \subseteq \mathrm{Mod}(F)$.

Notice that our definition of stationary generated models also accommodates negation in the head of a rule and 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.

Lemma 7 *Let* $\{I_n \mid n < \omega\}$ *a t-increasing sequence of partial interpretations, i.e.* $I_n \preceq I_{n+1}$ *for all* $n < \omega$, *and let be* $sup\{I_n \mid n < \omega\} = I_\omega$, *and* $I_\omega \preceq I$. *Let* F *be a quantifier free sentence.*

1. *If* $[I_\omega, I](F) \geq \frac{1}{2}$, *then there is a number* $n < \omega$ *such that* $[I_n, I](F) \geq \frac{1}{2}$.
2. *If* $[I_\omega, I](F) = 1$, *then there is a number* $n < \omega$ *such that* $[I_n, I](F) = 1$.

Proposition 8 *Let* $P \subseteq$ GLP *and let* $I \in \mathrm{Mod}_{statg}(P)$ *which is generated by the sequence* $\{I_\alpha : \alpha < \kappa\}$. *Then there is an ordinal* $\beta \leq \omega$ *such that* $I_\beta = I$.

Corollary 9 *If* $P \subseteq$ GLP *and* $I \in \mathrm{Mod}_{statg}(P)$, *then there is either a finite* P*-stationary chain, or a* P*-stationary chain of length* ω, *generating* I.

We now relate the stationary generated models to the stable generated two-valued models as introduced in [HW97]. We recall the definition of [HW97].

Definition 8 (Stable Generated Model) *[HW97] Let* $P \subseteq$ GLP. *A two-valued model* M *of* P *is called* stable generated, *symbolically* $M \in \mathrm{Mod}_{sg}(P)$, *if there is a chain* $\{I_\alpha : \alpha < \omega\}$ *of two-valued Herbrand interpretations such that*

1. *$m \leq n$ implies $I_m \subseteq I_n$ and $I_0 = \emptyset$.*
2. *I_{n+1} is a minimal two-valued extension of I_n which is contained in M and which satisfies all sequents whose body is true in every two-valued interpretation from the set $\{J \mid I_n \subseteq M\}$.*
3. *$M = \bigcup\{I_n \mid n < \omega\}$.*

We also say that M is generated *by the P-*stable chain *$\{I_n \mid n < \omega\}$.*

Proposition 10 *Let $P \subseteq$ GLP. A two-valued model I of P is stable generated if and only if it is a stationary generated model of P.*

Corollary 11 *Let $P \subseteq$ GLP. Then* $\mathrm{Mod}_{sg}(P) \subseteq \mathrm{Mod}_{statg}(P)$.

Example 1 *Let $S = \{\Rightarrow a, b;\ a \Rightarrow b\}$. Then $M = \{a, b\}$ is not minimal since $\{b\}$ is a model of S. But $\{a, b\}$ is stable: $I_0 = \emptyset$, $S_{[\emptyset,\{a,b\}]} = \{\Rightarrow a, b\}$; and since $\{a\} \in \mathrm{Min}\{I \mid \emptyset \leq I \leq M, I \models a \vee b\}$, we obtain $S_{[\{a\},\{a,b\}]} = \{\Rightarrow a, b; a \Rightarrow b\}$. Obviously, $\{a, b\}$ is a minimal extension of $\{a\}$ satisfying $a \vee b$ and b.*

5 Stationary Generated Models of Normal Logic Programs

The aim of this section is to prove that for normal logic programs the stationary models introduced in [Prz91] coincides with our stationary generated models. To make the paper self-contained we recall the main notions. Let P be 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 atomic. Let $I \subseteq B \cup \overline{B}$ be a (consistent) partial interpretation. The transformation $tr_I(r)$ is defined as follows.

- $tr_I(B^+(r)) = B^+(r)$ (positive literals are not changed);
- $tr_I(\ not\, b_i) = \mathbf{f}$, if $b_i \in I$; $tr_I(\ not\, b_i) = \mathbf{t}$, if $not\, b_i \in I$; $tr_I(\ not\, b_i) = \mathbf{u}$, if $\{b_i,\ not\, b_i\} \cap I = \emptyset$. Then $tr_I(B^-(r)) = tr_I(\ not\, b_1), \ldots, tr_I(\ not\, b_n)$;
- $tr_I(r) = B^+(r), tr_I(B^-(r)) \Rightarrow H(r)$.

The resulting program P/I which is called the $I-$*reduction* of P is defined by $P/I := \{tr_I(r) \mid r \in [P]\}$. P/I is an example of a so-called non-negative program [Prz91]. A normal logic program P is said to be non-negative, symbolically $P \subseteq$ PLP, if for every rule $r \in P$ the body $B(r)$ of r satisfies the condition $B(r) \subseteq At(\sigma) \cup \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. Every interpretation I contains $\mathbf{t}$ and $not\,\mathbf{f}$, and satisfied $\{\mathbf{u},\ not\,\mathbf{u}\} \cap I = \emptyset$. Every non-negative logic program has a t-least partial model that can be constructed as follows [Prz91].

Definition 9 *Let $P \subseteq$ PLP. The operator $T_P : 2^{\boldsymbol{I}} \rightarrow 2^{\boldsymbol{I}}$ is defined as follows.*
$T_P(I) = \{a \mid$ *there is a rule* $B(r) \Rightarrow a \in [P]$ *such that* $I(\bigwedge B(r)) = \mathbf{t}\} \cup$
$\{\, not\, a \mid$ *for every rule* $r \in [P]$ *satisfying* $H(r) = a$ *it is* $I(\bigwedge B(r)) = \mathbf{f}\}$.

The operator T_P is monotonic with respect to the truth-ordering $\preceq$. Since $(2^{\boldsymbol{I}}, \preceq)$ is a complete partial ordering the operator T_P has a least fixpoint I being a model of P. I is defined as follows. Let $I_0 = \{\, not\, a \mid a \in GAt\}$, i.e. I_0 is the t-least interpretation. We define an t-increasing sequence of partial interpretations: $I_0 \preceq I_1 \preceq \ldots \preceq I_n \preceq \ldots$ by $I_{n+1} = T_P(I_n)$. Obviously, $I_n \preceq I_{n+1}$, for $n < \omega$. Then $sup\{I_n : n < \omega\}$ is the least partial model of P; we denote it by $lpm(P)$.

Definition 10 *Let $P \subseteq$ NLP and I a partial interpretation. I is said to be a stationary model of P if and only if $lpm(P/I) = I$.*

Lemma 12 *Let $P \subseteq$ NLP and let I be a stationary model of P with the generating sequence $\{I_n \mid n < \omega\}$, $sup\{I_n : n < \omega\} = I$. Then for every $r \in [P]$, and every $n < \omega$, the following conditions are equivalent:*
1) $I_n(\bigwedge B(tr_I(r))) \geq \mathbf{u}$;
2) for all J satisfying the condition $I_n \preceq J \preceq I$ it holds $J(\bigwedge B(r)) \geq \mathbf{u}$.

Proof: 1) $\rightarrow$ 2). Let $r := a_1, \ldots, a_m,\ not\, b_1, \ldots,\ not\, b_n \Rightarrow c$, and $tr_I(r) = a_1, \ldots, a_m, v_1, \ldots, v_m \Rightarrow c$, where $v_i \in \{f, u, t\}$. Now we assume, that $I_n(\bigwedge B(tr_I(r))) \geq \mathbf{u}$, then $I_n(a_1 \wedge \ldots \wedge a_m \wedge v_1 \wedge \ldots \wedge v_n) \geq \mathbf{u}$. Then, for every $J : I_n \preceq J \preceq I$ we have $\{a_1, \ldots, a_m\} \cap J = \emptyset$. Assume this is not the case. Then there is a $not\, a_j \in J$, and since $Neg(J) \subseteq Neg(I_n)$ this implies $not\, a_j \in I_n$, hence $I_n(a_j) = f$ which yields $I_n(\bigwedge B(tr_I(r)) = \mathbf{f}$, which is a contradiction. This implies $J(a_i) \geq \mathbf{u}$ for every J: $I_n \preceq J \preceq I$. Furthermore, it holds $J(\,not\, b_j) \geq \mathbf{u}$ for every J: $I_n \preceq J \preceq I$. Assume, there is such an interpretation J satisfying $J(\,not\, b_j) = \mathbf{f}$. Then $b_j \in Pos(J)$ and this implies by $J \preceq I$ the condition $b_j \in Pos(I)$. By definition of the translation tr_I this would imply $tr_I(\,not\, b_j) = \mathbf{f}$, a contradiction to $I_n(\bigwedge B(tr_I(r))) \geq \mathbf{u}$.
2) $\rightarrow$ 1). Now we assume, that for all J: $I_n \preceq J \preceq I$: $J(\bigwedge B(r)) \geq \mathbf{u}$ We show that then $I_n(\bigwedge B(tr_I(r))) \geq \mathbf{u}$. Obviously, $I_n(a_1 \wedge \ldots \wedge a_n) \geq u$. It remains to show that $I_n(v_1 \wedge \ldots \wedge v_n) \geq u$. Assume this is not the case, then there is a number $j \leq n$ such that $I_n(v_j) = \mathbf{f}$. This implies $b_j \in I$. But then there is an extension J: $I_n \preceq J \preceq I$ such that $b_j \in Pos(J)$, hence $J(\,not\, b_j) = \mathbf{f}$, and this yields $J(\bigwedge B(r)) = \mathbf{f}$, which is a contradiction. □

We shall show below that the stationary generated models of a normal logic program S agree with the fixpoints of Γ_S, i.e. with stationary models as defined in [Prz91]. Since the definition of the extended Gelfond-Lifschitz transformation requires a specific rule syntax, the definition of stationary models based on it is not very general; as a consequence, Gelfond and Lifschitz are not able to treat negation-as-failure as a standard connective, and to allow for arbitrary formulas in the body of a rule. The interpretation of negation-as-failure according the stationary (generated) semantics seems to be the first general standard logical treatment of non-monotonic logic programs.

Proposition 13 *Let P be a normal logic program and I a stationary model of P. Then I is a stationary generated model of P.*

Proof: By assumption we have $I = lpm(P/I)$, and let $I_0 \preceq I_1 \preceq \ldots \preceq I_n$ the defining t-increasing sequence for I. Then $I = sup\{I_n \mid n < \omega\}$. We show that $\{I_n : n < \omega\}$ is a stationary chain generating I. By definition is $I_0 = \overline{GAt(\sigma)}$. We show that for every $n < \omega$ the interpretation I_{n+1} is a t-minimal extension of I_n satisfying the set $P_{[I_n,I]} = \{r \in [P] \mid$ for all $J : I_n \preceq J \preceq I$ it is $J(\bigwedge B(r)) \geq u\}$. Firstly, we prove for all $r \in P_{[I_n,I]}$ the condition $I_{n+1}(r) = t$. Then we show: if K is a partial interpretation satisfying the condition $I_n \preceq K \preceq I_{n+1}$, and if $K(r) = t$ for all $r \in P_{[I_n,I]}$, then $K = I_{n+1}$.
By definition it is $I_{n+1} = \{a \mid B(r) \Rightarrow a \in tr_I([P])$ and $I_n(\bigwedge B(r)) = t\} \cup \{\, not\, a$: for all $B(r) \Rightarrow a \in tr_I([P])$ it is $I_n(\bigwedge B(r)) = f\}$. Let $r \in P_{[I_n,I]}$, we show that $I_{n+1}(\bigwedge B(r)) \leq I_{n+1}(H(r))$. By lemma 12 it is $I_n(\bigwedge B(r)) \geq u$. If $I_n(\bigwedge B(r) = t$, then $I_{n+1}(H(r)) = t$ (by definition of I_{n+1} and we are ready. Now assume $I_n(\bigwedge B(tr_I(r))) = u$. It is sufficient to show that $I_{n+1}(H(r)) \geq u$. Assume this is not the case, then $I_{n+1}(H(r)) = I_{n+1}(a) = f$, hence $not\, a \in I_{n+1}$. But then $not\, a \in I_n$, hence $I_n(a) = f$. By definition of I_{n+1} for all $B(s) \Rightarrow a \in tr_I([P])$ is $I_n(\bigwedge B(s)) = f$, in particular $I_n(\bigwedge B(tr_I(r))) = f$, this is a contradiction. Hence $I_{n+1} \models P_{[I_n,I]}$. Now let K be satisfy the condition $I_n \preceq K \preceq I_{n+1}$. Obviously, if $K \models P_{[I_n,I]}$, then $Pos(K) \subseteq Pos(I_{n+1})$. It remains to show that $Neg(K) = Neg(I_{n+1})$. Assume this is not the case, then there is an element $not\, a \in Neg(K) - Neg(I_{n+1})$. Then a does not satisfy the condition for $Neg(I_{n+1})$, i.e. there is a rule $B(s) \Rightarrow a \in tr_I([P])$ such that $I_n(\bigwedge B(s)) \geq u$ (o.w. $not\, a \in Neg(I_{n+1})$. Let be $B(s) = tr(B(r))$. Then, by lemma 12 for all $J : I_n \preceq J \preceq I$ we have $J(\bigwedge B(r)) \geq u$, in particular $K(\bigwedge B(r)) \geq u$. Since $K(a) = f$ it follows $K \not\models B(r) \Rightarrow a$. From this follows that $Neg(K) - Neg(I_{n+1}) = \emptyset$, hence $Neg(K) = Neg(I_{n+1})$,

then I_{n+1} satisfies the conditions according to the definition of stationary generated model. □

Proposition 14 *Let I be a stationary generated model of the normal logic program P. Then I is a stationary model of P.*

Proof: Let $\{I_n : n < \omega\}$ be a stationary chain generating I. We show that this sequence coincides with the sequence associated to the least model of $tr_I(P) = P/I$. Let $Q = P/I$. We show that $T_Q(I_n) = I_{n+1}$ for every $n < \omega$, and we have to prove the following conditions:
a) $Pos(T_Q(I_n)) = Pos(I_{n+1})$, and b) $Neg(T_Q(I_n)) = Neg(I_{n+1})$.
a) To show: $Pos(T_Q(I_n)) \subseteq Pos(I_{n+1})$. Let be $a \in Pos(T_Q(I_n))$, then there is a rule $B(r) \Rightarrow a \in tr_I([P])$ such that $I_n(\bigwedge B(r)) = t$. Let $s \in [P]$ the rule satisfying $tr_I(s) = r$, and $B(s) = a_1, \ldots, a_m$, $not\, b_1, \ldots,$ $not\, b_n$. Then $\{a_1, \ldots, a_m\} \subseteq I_n$. Furthermore, $tr_I(\, not\, b_j) = t$ for all $j \leq n$. That means $not\, b_j \in I$, and this implies the condition $s \in P_{[I_n,I]}$. Since $I_{n+1} \models s$ and $I_{n+1}(\bigwedge B(s)) = t$ this yields $a \in I_{n+1}$, hence finally $Pos(T_Q(I_n)) \subseteq I_{n+1}$. By induction hypothesis we assume $T_Q(I_{n-1}) = I_n$. Let $a \in Pos(I_{n+1}) - Pos(I_n)$, then there is a rule $B(r) \Rightarrow a \in P_{[I_n,I]}$, i.e. $[I_n, I](\bigwedge B(r)) \geq u$. But then there must be a rule of this kind satisfying $I_n(\bigwedge B(r)) = t$ (otherwise $I_{n+1} - \{a\}$ would be a model $P_{[I_n,I]}$). This shows that $a \in Pos(T_Q(I_n))$.
b) This condition follows immediately from the following claim:
$(*)$: $not\, a \in Neg(I_{n+1})$ iff for all $B(r) \Rightarrow a \in [P]$ it holds $I_n(\bigwedge B(tr_I(r))) =$ **f**. To prove $(*)$, let $not\, a \in Neg(I_{n+1})$, and assume there is a rule $B(r) \Rightarrow a \in [P]$ such that $I_n(\bigwedge B(tr_I(r))) \geq u$. Then by lemma 12 we have $K(\bigwedge B(r)) \geq u$ for every $K : I_n \preceq K \preceq I$, hence $B(r) \Rightarrow a \in P_{[I_n,I]}$. But then $I_{n+1} \not\models B(r) \Rightarrow a$, because $I_{n+1}(\bigwedge B(r)) \geq u$ and $I_{n+1}(a) = f$. Hence $I_{n+1} \not\models P[I_n, I]$, a contradiction.

Assume for all $r \in tr_I([P])$ with $B(r) \Rightarrow a$ the condition $I_n(\bigwedge B(r)) = f$. We have to show that $not\, a \in Neg(I_{n+1})$. Assume, this is not the case, then $not\, a \notin Neg(I_{n+1})$, then $not\, a \in Neg(I_n) - Neg(I_{n+1})$. From this follows that $I_{n+1} \cup \{not\, a\} \models P_{[I_n,I]}$, which gives a contradiction, because I_{n+1} is a minimal extension of I_n satisfying $P_{[I_n,I]}$. Let $I' = I_{n+1} \cup \{a\}$. We show: for all $r \in P_{[I_n,I]}$ the condition $I'(\bigwedge B(r)) \leq I'(H(r))$. If $H(r) \neq a$, then this is clear. Now let be $B(r) \Rightarrow a \in [P]$ and $tr_I(r) = B(s) \Rightarrow a$. By assumption is $I_n(\bigwedge B(s)) = f$. We show that $I_n(\bigwedge B(r)) = f$ (this is indeed sufficient). Let be $B(s) = a_1, \ldots, a_m, v_1, \ldots, v_n$, $tr_I(\, not\, b_j) = v_j$. If $I_n(a_i) = f$, then $I_n(\bigwedge B(r)) = f$ and we are ready. Assume $I_n(a_i) \geq u$ for every a_i, $i \leq m$. Then there is a $not\, b_j$ such that $tr_I(\, not\, b_j) = f$, which means $b_j \in I$. Then there is an $J : I_n \preceq J \preceq I$ such that $b_j \in J$, hence $J(\bigwedge B(r)) = f$, and this means $r \in P_{[I_n,I]}$. $\square$

6 Conclusion and Future Research

By introducing a new general definition of stationary generated models, we have established the foundation of a theory of *partial models* for generalized logic programs. As a special case we get a model-theoretic interpretation of the well-founded semantics for normal logic programs. The consequence operator for generalized logic programs P - based on stationary generated models - exhibits a form of non-monotonic reasoning which is determined by the following definition: $P \models_{statg} \phi$ iff $Mod_{statg}(P) \subseteq Mod_H(\phi)$, ϕ a quantifier-free sentence. The corresponding closure operator of $\models_{statg}$ is defined by: $C_{statg}(P) = \{\phi | \phi$ quantifier-free and $P \models_{statg} \phi\}$. We believe that only cumulative consequence relations allow the development of a reasonable proof theory. Hence, it is an interesting task to find natural cumulative approximations of C_{statg}. In [HL07] non-monotonic reasoning was successfully applied to the integration problem for ontologies. We will explore the expressive power of generalized logic programs with stationary generated semantics for the representation and processing of knowledge in the field of clinical medicine.

Acknowledgment
Thanks due to the anonymous referees for their criticism and useful comments.

References

[AHP00] J. Alferes, H. Herre, L. M. Pereira. Partial Models of Extended Generalized Logic Programs. Int. Conference Computational Logic 2000, Springer LNAI 1861, 2000, pages 149-163

[AP1996] J. Alferes, L. M. Pereira. Reasoning with Logic Programming. Springer LNAI vol. 111, 1996

[EH99] J. Engelfriet, H. Herre. Stable Generated Models, Partial Temporal Logic and Disjunctive Defaults, *Journal of Logic Programming* 41 (1): 1-25, 1999

[GL88] 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

[HJW95] H. Herre, J. Jaspars and G. Wagner. Partial logics with two kinds of negation as a foundation for knowledge-based reasoning, in D. Gabbay and H. Wansing (Eds.), *What is negation ?*, pages 121-159. Kluwer Academic Publishers, 1999

[HW97] H. Herre and G. Wagner. Stable Models Are Generated by a Stable Chain, *Journal of Logic Programming*, 30 (2): 166-177, 1997

[HL07] R. Hoehndorf, F. Loebe, J. Kelso and H. Herre. Representing default knowledge in biomedical ontologies: Application to the integration of anatomy and phenotype ontologies. *BMC Bioinformatics* Vol. 8, pp. 377.1-377.12, 2007

[LTT99] V. Lifschitz, L. R. Tang, H. Turner. Nested Expressions in Logic Programs *Annals of Mathematics and Artificial Intelligence* 1999

[Prz90] T.C. Przymusinski. Well-founded semantics coincides with three-valued stable-semantics. *Fundamenta Informaticae 13* (1990), 445-463

[Prz91] T.C. Przymusinski. Stable semantics for disjunctive programs. *New Generation Computing*, 9:401-424, 1991

[Prz94] T.C. Przymusinsksi. Well-founded and stationary models of logic programs. *Annals of Mathematics and Artificial Intelligence* 12 (1994) 141-187

[Prz96] 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