Semantical Foundation of Extended Generalized Logic Programs

Heinrich Herre * Gerd Wagner †

Abstract

We present a definition of stable generated models for extended generalized logic programs (EGLP) which a) subsumes the definition of stable models for normal logic programs of [GL88], and is sound with respect to the answer set semantics for extended disjunctive programs of [GL91]. and b) does not refer to negation-as-failure by allowing for arbitrary formulas in the body and in the head of a rule (i.e. does not depend on the presence of any specific connective, nor on any specific syntax of rules). This notion was investigated for generalized logic programs in [HW97]. The intuition underlying our notion of stable generatedness is the property of groundedness which captures the concept of rule-governed generation of factual knowledge. Stable generated models of S can be understood as particular grounded models generated by a stable S-grounded chain. An important subclass of EGLP are the ampliative generalized logic programs (GLP^+) containing ampliative rules only. It turns out that every ampliative generalized logic program has a grounded model.

Key words: stable semantics, non-monotonic reasoning, practical reasoning.

1 Introduction and Background

In knowledge representation, two different notions of falsity arise in a natural way. Certain facts are *implicitly false by default* by being not verified in any intended model of the knowledge base. Others are *explicitly false* by virtue of a direct proof of their falsity, corresponding to their falsification in all intended models. These two kinds of falsity in knowledge representation are captured by the two negations, called *weak* and *strong*, of partial logic.[1] Logic programs can be viewed as knowledge bases consisting of *facts* and *rules*. Facts correspond to sentences of a suitably restricted language, and rules can be related to (Gentzen) sequents. While facts express extensional knowledge, rules express intensional knowledge. 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

*herre@informatik.uni-leipzig.de, Institut für Informatik, Univ. Leipzig, Augustusplatz 10-11, 04109 Leipzig, Germany

†gw@inf.fu-berlin.de, Institut für Informatik, Univ. Leipzig, Augustusplatz 10-11, 04109 Leipzig, Germany

[1] As was already noticed in [Wag91].

models are not adequate because they are not able to capture 'groundedness', i.e. the directedness of rules. 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 have generalized this notion in [HW97] by presenting a definition which is neither fixpoint-based nor dependent on any specific rule syntax. In the present paper we extend this definition to the class of extended generalized logic programs admitting weak and strong negation.

A *signature* $\sigma = \langle Rel, ExRel, Const, Fun\rangle$ consists of a set of relation symbols Rel, a set of *exact* relation symbols $ExRel \subseteq Rel$, a set of constant symbols $Const$, and a set of function symbols Fun. U_σ denotes the set of all ground terms of σ. For a tuple $t_1, \ldots, t_n$ we will also write $\vec{t}$ when its length is of no relevance. The logical functors are $-, \sim, \wedge, \vee, |, \supset, \forall, \exists$; where $-$ and $\sim$ are called *weak* and *strong negation*, $|$ is called *exclusive disjunction*, and $\supset$ is called *material implication*. $L(\sigma)$ 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 $\{-F, {\sim}F, F \wedge G, F \vee G, F|G, F \supset G, \exists x F, \forall x F\} \subseteq L(\sigma)$.

$L^0(\sigma)$ 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*); $\mathrm{Lit}(\sigma) = \mathrm{At}(\sigma) \cup \{{\sim}a : a \in At(\sigma)\}$, the set of all *literals*, and $\mathrm{XLit}(\sigma) = L(\sigma; -, \sim)$, the set of all *extended 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 $L(\sigma)$. $QF(\sigma) = L^0(\sigma; -, \sim, \vee, \wedge, |, \supset)$ denotes the set of all quantifier free sentences over the signature σ. If $\langle Y, <\rangle$ is a partial order, then $\mathrm{Min}(\langle Y, <\rangle)$ denotes the set of all minimal elements of Y, i.e. $\mathrm{Min}(\langle Y, <\rangle) = \{X \in Y \mid \text{ there is no } X' \in Y \text{s.th. } X' < X\}$. $Pow(X)$ or 2^X denote the power set of the set X.

2 Models and Sequents

Let $\sigma = \langle Rel, ExRel, Const, Fun\rangle$ be a signature. A Herbrand σ-interpretation is a set of literals $I \subseteq \mathrm{Lit}^0(\sigma)$. Its *universe* or *domain* is equal to the set of all ground terms U_σ; its canonical interpretation of ground terms is the identity mapping. An intepretation I assigns two relations r^I and $\tilde{r}^I$, which can be viewed as the truth and falsity extension, to every relation symbol $r \in Rel$: $r^I = \{\vec{t} \in U_\sigma^{a(r)} : r(\vec{t}) \in I\}$, and $\tilde{r}^I = \{\vec{t} \in U_\sigma^{a(r)} : {\sim}r(\vec{t}) \in I\}$, where $a(r)$ denotes the arity of r, such that for all exact predicates $r \in ExRel$, $r^I \cup \tilde{r}^I = U_\sigma^{a(r)}$.

The class of all partial Herbrand σ-interpretations is denoted by $\boldsymbol{I}_4^H(\sigma) = 2^{\mathrm{Lit}^0(\sigma)}$, while the subclass of all *coherent* Herbrand σ-interpretations is defined as

$$\boldsymbol{I}_c^H(\sigma) = \{I \in 2^{\mathrm{Lit}^0(\sigma)} : \text{ if } a \in I, \text{ then } {\sim}a \notin I\}$$

In the sequel we shall simply say 'interpretation' instead of 'partial Herbrand interpretation'. An *instantiation* over an interpretation I is a function $\nu : Var \to U_\sigma$, which can be naturally extended to arbitrary terms by setting $\nu(f(t_1,\ldots,t_n)) = f(\nu(t_1),\ldots,\nu(t_n))$. Analogously, an instantiation 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 $\models \subseteq \boldsymbol{I}_4^H(\sigma) \times L^0(\sigma)$ between an interpretation and a sentence is defined inductively as follows.

Definition 1 (Model Relation) *1. For literals $l \in \mathrm{Lit}^0(\sigma)$, $I \models l$ iff $l \in I$.*

2. $I \models F \wedge G$ iff $I \models F$ and $I \models G$.

3. $I \models F \vee G$ iff $I \models F$ or $I \models G$.

4. $I \models \exists x F(x)$ iff $I \models F(t)$ for some $t \in U_\sigma$.

5. $I \models \forall x F(x)$ iff $I \models F(t)$ for all $t \in U_\sigma$.

6. $I \models -F$ iff $I \not\models F$.

In addition, we assume DeMorgan-style rewrite rules for handling the combination of $\sim$ with $-, \wedge, \vee, \forall, \exists$, and the following definitions:

$$\begin{array}{rcl} F|G & \longrightarrow & (F \wedge -G) \vee (G \wedge -F) \\ F \supset G & \longrightarrow & -F \vee G \end{array}$$

The model relation between an interpretation $I \in \boldsymbol{I}_4^H(\sigma)$ and a formula $F \in L(\sigma)$ is defined by

$$I \models F \text{ iff } I \models F\nu \text{ for every instantiation } \nu : \mathit{Var} \to U_\sigma$$

We obtain the model operators $\mathrm{Mod}_*(X) = \{I \in \boldsymbol{I}_*^H : I \models X\}$ for $* = 4, c$, and the corresponding consequence relations defined by $X \models_* F$ iff $\mathrm{Mod}_*(X) \subseteq \mathrm{Mod}_*(F)$.

Definition 2 (Extension of a Model and Persistence) *Let $I, I' \in \boldsymbol{I}_4^H$ be two interpretations. We say that I'* extends *I, resp. I' is* informationally greater *or equal than I if $I \subseteq I'$. A sentence F is said to be persistent if $I \subseteq I'$ and $I \models F$ implies $I' \models F$.*

Definition 3 (Minimal Models) *For $F \in L(\sigma) \supseteq X$, and $* = 4, c$, we define $\mathrm{Mod}_*^m(X) = \mathrm{Min}(\langle \mathrm{Mod}_*(X)\ ,\ \subseteq\rangle)$, and $X \models_*^m F$ iff $\mathrm{Mod}_*^m(X) \subseteq \mathrm{Mod}_*(F)$. Let $Lit_m(F)$ be the set of all ground literals occuring in a minimal model of F.*

Traditionally, Gentzen sequents are used in a schematic way in sequent calculi in order to express valid transitions from one argument schema to another. In other words, a sequent in a sequential inference rule stands for a whole class of propositional substitution instances. Here, we propose to use sequents in a non- schematic way for the purpose of representing rule knowledge. A sequent, then, is not a schematic but 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 $F_i, G_j \in L(\sigma)$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. The body *of s, denoted by Bs, is given by $\{F_1, \ldots, F_m\}$, and the* head *of s, denoted by Hs, is given by $\{G_1, \ldots, G_n\}$. $\mathrm{Seq}(\sigma)$ denotes the class of all sequents s such that $Hs, Bs \subseteq L(\sigma)$, and for a given set $S \subseteq \mathrm{Seq}(\sigma)$, $[S]$ denotes the set of all ground instances of sequents from S.*

For a sequent $\Rightarrow F$ with empty body we also write more simply F.

Definition 5 (Satisfaction Set and logical equivalence)

1. *Let $I \in \boldsymbol{I}_4^H(\sigma)$, and $F \in L(\sigma)$.*
   Then $\mathrm{Sat}_I(F) = \{\nu \in {U_\sigma}^{Var} : I \models F\nu\}$

2. *Two formulas F and G are logically equivalent, denoted by $F \equiv G$ iff $Sat_I(F) = Sat_I(G)$.*

Definition 6 (Model of a Sequent) *Let $I \in \boldsymbol{I}_4^H$. Then,*

$$I \models F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n \quad \textit{iff} \quad \bigcap_{i \leq m} Sat_I(F_i) \subseteq \bigcup_{j \leq n} Sat_I(G_j)$$

We define the following classes of sequents:

1. $\mathrm{EDLP}^+(\sigma) = \{s \in \mathrm{Seq}(\sigma) \mid Hs, Bs \subseteq \mathrm{Lit}(\sigma), Hs \neq \emptyset\}$.

2. $\mathrm{EDLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) \mid Hs \subseteq \mathrm{Lit}(\sigma), Hs \neq \emptyset, Bs \subseteq \mathrm{XLit}(\sigma)\}$.

3. $\mathrm{EGLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) \mid Hs, Bs \subseteq L(\sigma; -, \sim, \wedge, \vee, |, \supset)\}$.

4. $\mathrm{EGLP}^+(\sigma) = \{s \in \mathrm{Seq}(\sigma) \mid Bs \subseteq L(\sigma; -, \sim, \wedge, \vee, |, \supset), Hs \subseteq L(\sigma; \sim, \wedge, \vee),$
   $Hs \neq \emptyset\}$

5. $\mathrm{GLP}^+(\sigma) = \{s \in \mathrm{Seq}(\sigma) \mid Bs \subseteq L(\sigma; -, \wedge, \vee, |, \supset), Hs \subseteq L(\sigma; \wedge, \vee),$
   $Hs \neq \emptyset\}$

6. $\mathrm{GLP}(\sigma) = \{s \in \mathrm{Seq}(\sigma) \mid Bs, Hs \subseteq L(\sigma; -, \wedge, \vee, |, \supset)\}$

A set $P \subseteq \mathrm{EDLP}$ is called an extended disjunctive logic program, a set $P \subseteq \mathrm{EGLP}$ is said to be an extended generalized logic program. A rule whose head does not contain weak negation, nor exclusive disjunction, nor material implication, is called *ampliative*. An ampliative rule is is said to be positive if its head does not contain strong negation.

For $S \subseteq \mathrm{EGLP}(\sigma)$, we define the model operator

$$\mathrm{Mod}_c(S) \quad = \quad \{I \in \boldsymbol{I}_c^H(\sigma) : I \models s, \text{ for all } s \in S\}$$

and the associated entailment relation

$$S \models_c F \quad \text{iff} \quad \mathrm{Mod}_c(S) \subseteq \mathrm{Mod}_c(F)$$

where $F \in L(\sigma)$. Instead of sequent notation $B \Rightarrow H$ we shall also use sometimes the rule notation $F \leftarrow G$, where $F = \bigvee H$ and $G = \bigwedge B$.

With respect to a class of interpretations $\boldsymbol{K}$, we write $\boldsymbol{K} \models F$ iff $I \models F$ for all $I \in \boldsymbol{K}$. We denote the set $S_{\mathbf{K}}$ of all sequents from a sequent set S which are applicable in $\boldsymbol{K}$ by

$$S_{\mathbf{K}} = \{s \in [S] : \boldsymbol{K} \models Bs\}$$

If $\boldsymbol{K}$ is a singleton, we omit brackets.

Observation 1 *Let $S \subseteq \mathrm{EGLP}$ be an extended generalized logic program. Then,*

$$\mathrm{Mod}_H(S) = \mathrm{Mod}_H([S])$$

where $[S]$ is the Herbrand instantiation of S.

A semantics for sequents is given by a preferred model operator $\Phi : Seq \to 2^{I_H}$, satisfying the condition $\Phi(S) \subseteq \mathrm{Mod}_H(S)$, and defining the preferential entailment relation

$$S \models_\Phi F \text{ iff } \Phi(S) \subseteq \mathrm{Mod}_H(F)$$

The main question concerning the semantics of logic programs is: which preferred model operator captures the intended semantics for (generalized) extended logic programs ?

If we assume Mod_H as the model operator for the semantics of sequents then there is no difference between sequents and material implications, since for $F, G \in L$, it holds that

$$\mathrm{Mod}_H(F \Rightarrow G) = \mathrm{Mod}_H(F \supset G).$$

3 Grounded and Supported Models

A sequent s may have several meanings. The traditional meaning of a sequent, based on classical model theory, is given by the formula $\bigwedge Bs \supset \bigvee Hs$ and the usual model relation. Our intuitive understanding of rules suggests another meaning which interprets a sequent as a rule for generating factual (extensional) knowledge. We consider a model of a set S of sequents as intended if it can be generated bottom-up starting from zero information by an iterated application of the sequents $s \in [S]$. Such models are called *grounded* and we assume that an intended model of S has to be grounded.

Definition 7 (Grounded Model) *Let $S \subseteq \mathrm{EGLP}(\sigma)$. $M \in \mathrm{Mod}_H(S)$ is called a* grounded model *of S, symbolically $M \in \mathrm{Mod}_g(S)$, if there is a chain of Herbrand interpretations $I_0 \subseteq \ldots \subseteq I_\kappa$ such that $M = I_\kappa$, and*

1. *$I_0 = \emptyset$.*

2. *For successor ordinals α with $0 < \alpha \leq \kappa$, I_α is a minimal extension of $I_{\alpha-1}$ satisfying the heads of certain sequents whose bodies hold in $I_{\alpha-1}$, i.e. there exists $S_\alpha \subseteq S_{I_{\alpha-1}}$ such that*

$$I_\alpha \in \mathrm{Min}\{I \in \boldsymbol{I}_H(\sigma) \mid I \supseteq I_{\alpha-1},\ s.th.\ for\ all\ s \in S_\alpha : I \models \bigvee Hs\}$$

3. For limit ordinals $\lambda \leq \kappa$, $I_\lambda = \bigcup_{\alpha<\lambda} I_\alpha$.

We also say that M is generated *by the S*-grounded chain *$I_0 \subseteq \ldots \subseteq I_\kappa$. A S*-grounded chain *is said to be* stable *if for every $s \in S_\alpha$, and $\alpha \leq \beta \leq \kappa$, it holds that $I_\beta \models Bs$; in this case I is said to be* stable grounded*. A special case of stable grounded models are the* end-regulated *models where the condition on S_α is defined as follows $S_\alpha = \{s : s \in S_{I_\alpha}$ such that $I_\beta \models Bs$ for every $\alpha \leq \beta \leq \kappa\}$.*

Let $\mathrm{Mod}_c^g(S)$ be the set of all grounded coherent models of S. If a knowledge base contains sequents with non-persistent body formulas, it may have minimal models which are not grounded. Consequently, minimal model semantics is, in general, not able to guarantee groundedness. This is illustrated by the following example.

Example 1 $S_1 = \{\Rightarrow r(c);\ \Rightarrow p(a), p(b);\ -p(x) \Rightarrow q(x)\}$.
Every intended model of S_1 should contain $q(c)$. Assume $I \models S_1$, but $q(c) \notin I$. Then $p(c) \in I$. But $p(c)$ cannot be generated from below starting with $\emptyset$ by applying the sequents from $[S_1]$ because $p(c)$ does not appear in the head of any rule $s \in [S_1]$. But $\{r(c), p(a), p(b), p(c)\}$ is a minimal model of S_1. Hence, the model operator Mod_m *does not capture the intended semantics of 'groundedness'.*

Example 2 *In grounded model semantics, sequents differ from material implication:*

$$\mathrm{Mod}_c^g(\neg p \supset q) = \{\{p\}, \{q\}\} \neq \mathrm{Mod}_c^g(\neg p \Rightarrow q) = \{\{q\}\}$$

Another important property of a model I is *supportedness*, introduced in [ABW88] for normal logic programs: I is supported if every atom $a \in I$ is underpinned by some rule $s \in [S]$, i.e. l occurs in the head of s and Bs is true in I. In the following definition several generalizations of this notion for the class of generalized logic programs are introduced.

Definition 8 *1. Let F be a quantifier free sentence and $Lit(F)$ the set of literals occurring in F. A ground literal l occurs positively in F if there is a conjunctive normal form $G = C_1 \wedge \ldots \wedge C_m$ of F such that*
- $Lit(G) \subseteq Lit(F)$;
- none of the conjuncts C_j is a tautology;
- there is a $j \leq m$ such that $l \in C_j$.
Let $Pos(F)$ be the set of all ground literals occurring positively in F.

2. For a set X of quantifier free sentences let be $Lit_m(X) = \{l : l$ is a ground literal occuring in a minimal model of $X\}$.

Lemma 1 *Let X be a set of quantifier free sentences, l ground literal and $l \notin Pos(F)$ for every $F \in X$. Then no minimal model of X contains l.*

Proof: We may assume that every $F \in X$ is in conjunctive normal form according to definition 8, i.e. $F := C_1 \wedge \ldots \wedge C_m$, C_j are clauses, $j \leq m$. Let $I \models X$ and $l \in I$. By assumption $l \notin Pos(C_j)$ for every $j \leq m$, that means $l \notin C_j$. Obviously, $I - \{l\} \models F$. Hence, I is not minimal for X. □

Definition 9 (Supported Model) *1. Let $P \subseteq EGLP$, $I \models_c P$. I is supported iff for every $l \in I$ there is a sequent $s \in [P]$ such that $l \in Pos(\bigvee Hr)$ and $I \models \bigwedge Br$.*

2. *I is minimally supported if for every $l \in I$ there is a finite subset $X \subseteq [P]$ such that $I \models Bs$ for every $s \in X$ and $l \in Lit_m(\{\bigvee Hs : s \in X\})$. For $n \geq 1$, I is said to be minimally n-supported if the set X can be chosen to have cardinality at most n.*

3. *I is atomically supported if for every $l \in I$ there is a finite set $X \subseteq [P]$ of rules whose bodies are true in I such that $\{\bigvee Hs : s \in X\} \models l$.*

4. *I is said to be strongly supported if for every quantifier free sentence F satisfying $I \models F$ there is a finite set $X \subseteq [P]$ of rules whose bodies are true in I such that $\{\bigvee Hs : s \in X\} \models F$.* [2]

5. *I is weakly supported if for every $l \in I$ there are finite subset $J \subseteq I$ and a $s \in [P]$ such that $J \models Bs$ and $l \in Pos(\bigvee Hs)$.*

The intended semantics for a logic program should satisfy at least both groundedness and supportedness.

Claim 2 *Let $P \subseteq EGLP$, and I a model of P.*

1. *If I is supported then I is weakly supported.*

2. *If I is minimally supported then I is supported.*

3. *If I is atomically supported then I is minimally supported.*

4. *If I is strongly supported then I is atomically supported.*

Proof: 1. Let I be supported, and $l \in I$. Then there is a sequent $s \in [P]$ such that $I \models Bs$ and $l \in Pos(\bigvee Hs)$. By lemma 3 there is a finite subset $I_f \subseteq I$ satisfying $I_f \models Bs$, hence I is weakly supported.
2. Assume I is minimally supported, but I not supported. Then there is a literal $L \in I$ such that for every rule $r \in [P]$ the condition $l \notin Pos(\bigvee Hs)$ holds. Let X be a finite subset of $[P]$, by lemma 1 holds $l \in \bigcup \mathrm{Mod}_c^m(\{\bigvee Hs : s\ X\})$. This contradicts minimal supportedness of I.
3. Let I atomically supported and $l \in I$. Then there is a finite subset $X \subseteq [P]$ such that for every $s \in X$ holds $I \models Bs$, and $\{\bigvee Hs : s \in X\} \models l$. This implies $l \in \bigcap Mod_m^c(\{\bigvee Hs\})$, hence I is minimally supported.
4. is immediate. □

Remark: None of the converses of the implications in claim 2 is true.

Example 3 *1. Let be $P_1 = \{\neg a \Rightarrow b, \quad , \neg a \Rightarrow a\}$, and $I = \{a, b\}$. I is weakly supported, but not supported.*

[2] Strongly supported models are equivalent to the causal models in [MC96].

2. $P_2 = \{\Rightarrow a \vee \neg b, \quad \Rightarrow c \vee \neg d, \quad \Rightarrow b\}$. *The model* $\{a, b, c\}$ *is supported, but not minimally supported.*

3. $P_3 = \{\Rightarrow a \vee b\}$; $\{a, b\}$ *is minimally supported, but not atomically supported.*

4. $P_4 = \{\Rightarrow a, \quad b \Rightarrow b\}$, $\{a\}$ *is atomically supported, but not strongly supported, because of* $\{a\} \models \neg b$.

In the sequel, we need the notion of an 'interval' of interpretations. Let be $[M_1, M_2] = \{M \in \boldsymbol{I}_c^H : M_1 \subseteq M_2\}$.

Lemma 3 *Let F be a quantifier free sentence, and I a Herbrand interpretation. If $I \models F$ then there is finite subset $I_f \subseteq I$ such that $[I_f, I] \models F$.*

Proof: We may assume that F is a conjunction of non-tautological clauses $C_j : j = 1, 2 \ldots, m$, $C_j = l_1^j \vee \ldots \vee l_{k(j)}^j$, l_i^j are extended ground literals. By assumption $I \models C_j$; C_j is said to be positively satisfied in I if $I \cap Pos(C_j) \neq \emptyset$. Let $\{D_1, \ldots, D_t\}$ be the set of all positively satisfied clauses from F, and let $a_i \in D_i \cap I$ an arbitrary element, $1 \leq i \leq t$. It is easy to show that $I_0 = \{a_1, \ldots, a_t\}$ satisfies the desired condition. $\Box$.

Claim 4 *Every grounded model is weakly supported.*

Proof: Let be $P \subseteq EGLP$, and I a grounded model of P. Then $I = \bigcup_{\alpha<\kappa} I_\alpha$, and $(I_\alpha)_{\alpha<\kappa}$ is a grounded chain generating I. Let $l \in I$, then there is a α such that $l \in (I_{\alpha+1} - I_\alpha)$, and this implies that $l \in \bigcup Mod_c^m(I_\alpha \cup \{\bigvee Hs : s \in [P] \quad \text{and} \quad I_\alpha\})$. By lemma 1 this implies the existence of a rule $r \in [P]$ such that $I_\alpha \models Br$, and $l \in Pos(\bigvee Hs)$. By lemma 3 there is a finite subset $I_f \subseteq I_\alpha$ such that $I_f \models Br$. Hence, I is weakly supported. $\Box$

Claim 5 *Every ampliative generalized logic program has a grounded model.*

Proof: Let $S \subseteq GLP^+$; we construct a sequence $I_0, I_1, \ldots, I_\alpha$ of interpretations satisfying following conditions:

1. $I_0 = \emptyset$.

2. For successor ordinals α with $0 < \alpha \leq \kappa$, I_α is a minimal extension of $I_{\alpha-1}$ satisfying the heads of all sequents whose bodies hold in $I_{\alpha-1}$, i.e.

$$I_\alpha \in \text{Min}\{I \in \boldsymbol{I}_H(\sigma) \mid I \geq I_{\alpha-1}, \text{ s.th. for all } s \in S_{I_{\alpha-1}} : I \models \bigvee Hs\}$$

3. For limit ordinals $\lambda \leq \kappa$,

$$I_\lambda = \bigcup_{\alpha<\lambda} I_\alpha$$

Since the heads of $s \in S$ do not contain negation the set $\mathrm{Min}\{I \in \boldsymbol{I}_H(\sigma) \parallel I \geq I_{\alpha-1}$, s.th. for all $s \in S_{I_{\alpha-1}} : I \models \bigvee Hs\}$ is not empty. Obviously, there is a least ordinal β such that $I_{\beta+1} = I_\beta$. Then, by construction, I_β satisfies the groundedness condition. It remains to show that I_β is a model of S. Assume $s \in S$ and $I \models B(s)$; then there exists a finite subset $I_0 \subseteq I$ such that $[I_0, I] \models B(s)$. Let $\alpha \leq \beta$ the least ordinal such that $I_0 \subseteq I_\alpha$. Then $I_\alpha \models B(s)$, and by construction of I it follows $I_{\alpha+1} \models Hs$. Since Hs does not contain negation it follows $I \models Hs$, hence $I \models S$. □

Remark: Claim 5 can be generalized to arbitrary ampliative extended generalized logic programs by admitting non-coherent models, i.e. every ampliative extended generalized logic program has a four-valued grounded model. On the other hand, as the following example shows, there are ampliative extended generalized logic programs without any coherent grounded model.

Example 4 $S = \{-b \Rightarrow a;\ -c \Rightarrow \sim a\}$. *S has the following coherent models* $\{\{b, c\}, \{b, \sim a\}, \{c, a\}\}$, *and none of them is grounded.*

One may argue that intended models should satisfy supportedness. Then, $S = \{-p \Rightarrow p\}$ would not have any intended model since the only model $\{p\}$ of S is not supported. This seems to contradict our intuition. Groundedness does not imply supportedness.

Example 5 *For* $S = \{\Rightarrow p;\ -q \Rightarrow r;\ p \Rightarrow q\}$, *we obtain* $\mathrm{Mod}_c^g(S) = \{\{p, r, q\}, \{p, q\}\}$. *However,* $\{p, r, q\}$ *is not supported by* S*: there is no rule* $s \in [S]$*, such that* $r \in Hs$*, and* $\{p, r, q\} \models Bs$*.*

There are many natural examples of intended models satisfying supportedness. If we postulate that an intended model has to be supported we have look for a stronger notion of groundedness. Such a strenghening of groundedness is realized by the concept of stable generatedness.

4 Stable Generated Models

The following definition of a *stable generated model* of extended generalized logic programs generalizes the *answer set* semantics of [GL91].

Definition 10 (Stable Generated Model) *Let* $S \subseteq \mathrm{EGLP}(\sigma)$. $M \in \mathrm{Mod}_c(S)$ *is called a* stable generated model *of* S*, symbolically* $M \in \mathrm{Mod}_c^s(S)$*, if there is a chain of Herbrand interpretations* $I_0 \leq \ldots \leq I_\kappa$ *such that* $M = I_\kappa$*, and*

1. $I_0 = \emptyset$.
2. *For successor ordinals* α *with* $0 < \alpha \leq \kappa$*,* I_α *is a minimal extension of* $I_{\alpha-1}$ *satisfying the heads of all sequents whose bodies hold in* $[I_{\alpha-1}, M]$*, i.e.*

$$I_\alpha \in \mathrm{Min}\{I \in \boldsymbol{I}_c^H(\sigma) : I \geq I_{\alpha-1},\ \textit{and}\ I \models \bigvee Hs,\ \textit{f.a.}\ s \in S_{[I_{\alpha-1}, M]}\}$$

3. For limit ordinals $\lambda \leq \kappa$, $I_\lambda = \bigcup_{\alpha<\lambda} I_\alpha$.

We also say that M is generated *by the S-*stable chain *$I_0 \leq \ldots \leq I_\kappa$.*

The stable entailment relation is defined as follows:

$$S \models^s_c F \quad \text{iff} \quad \mathrm{Mod}^s_c(S) \subseteq \mathrm{Mod}_c(F)$$

where $F \in L(\sigma)$. In [HW97] we showed that stable generated models of disjunctive programs are not always minimal and that there are stable generated minimal models of disjunctive programs which are not fixpoints of the Gelfond-Lifschitz operator.

Claim 6 *In the case of a (non-disjunctive) extended logic program, stable coherent models agree with answer sets as defined in [GL90].*

Proof: Analogously as in [HW97].

Claim 7 *The length of a S-stable chain generating a stable model can be restricted by ω.*

The proof is analogoues as in [HW97].

Remark 1 Using the notion of end-regulated induction as in [Mak93] ond could modify definition as follows. A model M of $P \subseteq EGLP$ is said to be *end-regulated* if there is a sequence $\{I_\alpha : \alpha < \kappa\}$ satisfying $I_\alpha \subseteq I_{\alpha+1}$, $I_0 = \emptyset$, and $M = \bigcup_{\alpha<\kappa} I_\alpha$ such that for sucessor ordinals α, $0 < \alpha < \kappa$, I_α is a minimal extension of $I_{\alpha-1}$ satisfying the heads of all sequents in $[P]$ whose bodies hold in $\{I_\beta : \alpha \leq \beta\}$. For limit ordinals $\lambda \leq \kappa$ $I_\lambda = \bigcup_{\alpha<\lambda} I_\alpha$.

Proposition 8 *Let P be a generalized logic program such that the bodies of the rules are conjunctions of literals. Then a model I of P is end-regulated if and only if it is stable generated.*

Proof: Assume I is end-regulated and $\{I_\alpha : \alpha < \kappa\}$ an associated sequence defining I. Then, $\{I_\alpha : \alpha < \kappa\}$ is a P- stable chain generating I. This follows immediately from the fact that for a conjunction F of ground literals we have: $\{I_\beta : \alpha \leq \beta \leq \kappa\} \models F$ if and only if $[I_\beta, I] \models F$. Analogously one concludes that every stable generated model of P is end-regulated. $\square$

Not every end-regulated model of a generalized logic program is stable generated as the following example shows.

Example 6 $P = \{\neg b \vee a \Rightarrow a; a \Rightarrow b\}$. *Then $\{a, b\}$ is an end-regulated a model of P using the sequence $\emptyset \subseteq \{a\} \subseteq \{a, b\}$. But $\{a, b\}$ is not stable generated because none of the rules can be applied to $\emptyset$ since $P_{[\emptyset,\{a,b\}]} = \emptyset$ (note that none of the bodies of the rules is satisfied in $\{b\}$).*

Question 1 *Is every stable generated model of a generalized logic program end-regulated?*

Claim 9 *Let $S \subseteq EGLP$, and I a stable generated model of S. Then I is grounded and supported.*

Proof: Since every stable chain is S-grounded it remains to show that I is supported. Let $I_0 \leq I_1 \leq \ldots \leq I_n \leq \ldots$ a stable chain generating I. and $l \in I$. Then there is a $n \in I_{n+1} - I_n$. It is easy to show that there is a $s \in S_{[I_n,I]}$ such that $l \in Pos(Hs)$. Assume this is not the case; then, by the lemma 1, there is minimal model of $I_n \cup \{\wedge Hs : s \in S_{[I_n,I]}\}$ containing l; this contradicts the condition $l \in I_{n+1}$. □.
The converse does not hold.

Example 7 $S = \{-p \Rightarrow q; q \Rightarrow p; p \Rightarrow q\}$. *The only model $I = \{p, q\}$ is supported and grounded; but it is not stable since $S_{[\emptyset,I]} = \emptyset$.*

Finally, we discuss the following property of stable models: the existence of founded support functions.

Definition 11 *Let $S \subseteq EGLP$, $I \models S$, and $g : I \to (\omega - \{0\}$ a function from I onto a initial segment of the natural numbers $\omega - \{0\}$.*

1. *g is said to be a* S-founded support function *iff the following conditions are satisfied: For all $l \in I$, $g(l) = k$, there exists a sequent $s \in [S]$ such that $l \in Pos(Hs)$, and for $I_k = \{l : g(l) < k\}$, $[I_k, I] \models_c \bigwedge Bs$. A model if S possessing a S-founded support function is called a* founded *model of S.*

2. *g is said to be a* minimally S-founded support function *iff the following conditions are satisfied: For all $l \in I$, $g(l) = k$, there exists a finite subset $s \in [S]$ such that $l \in \bigcup Mod_c^m(\{\bigvee Hs : s \in X\})$, and for $I_k = \{l : g(l) < k\}$, and for every $s \in X$ $[I_k, I] \models_c \bigwedge Bs$. A model if S possessing a minimally S-founded support function is called a* minimally founded *model of S.*

Obviously, any model of S with a S-founded support function is supported.
Remark: If P is a positive disjunctive program then a model I of P is founded iff it is sustained in the sense of [DC93].

Claim 10 *1. Let I be a stable model of S; then I has a S-founded support function.*

2. There are grounded and supported models I of S without any S-founded support function.

Proof: 1) Let I be a stable generated model of S and $\{I_n : n < \kappa\}$ stable chain for I, $\kappa \in \omega \cup \{\omega\}$. $g : I \to \omega - \{0\}$ is the following function: $g(a) = n$ iff $a \in I_n - I_{n-1}$. We show that g is a S-founded support function for I. Let $a \in I$ and $g(a) = n$, then $a \in I_n - I_{n-1}$, hence there is a rule $s \in [S]$, such that $[I_{n-1}, I] \models Bs$ and $a \in Pos(Hs)$. Obviously, $I_{n-1} = \{a : g(a) < n\}$, which implies the property of S-foundedness.
2) It is easy to see that the model $I = \{p, q\}$ of example 7 has no S-founded support function. □

The converse of the first part of claim 10 does not hold: there are generalized programs S with non-stable models I having a S-founded support function.

Example 8 *$S = \{\Rightarrow a, a \Rightarrow b \vee c\}$. $I = \{a, b, c\}$ has a minimally S-founded support function, but obviously I is not stable.*

One may ask, whether the converse of the first part of claim 10 is true if we restrict the class of programs. This is clarified by the following claim.

Claim 11 *A model of a normal program S is stable if and only if it has a S-founded support function.*

Proof: It is sufficient to prove the following. Let I be a model of S with the S-founded function g. Then I is stable. Let be $M_p = \{a : g(a) = p\}$; then $I = \bigcup_{1 \leq m \leq \kappa} M_m$, for a number $\kappa \leq \omega$. We introduce the rank of $a \in I$, denoted by rk(a), as follows rk(a) $= min\{m$: there is as rule $s \in [S]$ such that $Hs = a$ and $[N_m, I] \models Bs\}$, where $N_m = \{a : g(a) < m\}$. Define $J_n = \{a : rk(a) \leq n\}$. Let $\{I_k\}$ the chain of I with stability condition; we have to prove that I is reached by this sequence. For this purpose it is sufficient to show that $J_n \subseteq I_n$ for every $n < \kappa$. We show this inductively on n. Obviously, $J_0 \subseteq I_0$, and assume $J_n \subseteq I_n$. We have to show that $J_{n+1} \subseteq I_{n+1}$. Let $a \in J_{n+1}$, then $rk(a) \leq n+1$, and definition there is a rule $s \in [S]$, such that $[N_m, I] \models Bs$, where $m \leq n+1$. Let $Bs = a_1 \wedge \ldots a_s \wedge \ldots \wedge \neg b_1 \wedge \ldots \wedge \neg b_t$; then $\{a_1, \ldots, a_s\} \subseteq N_m$ and $\{b_1, \ldots b_t\} \cap I = \emptyset$. Since $N_m = \{a : g(a) < m\}$, for every $a_i : rk(a_i) < n+1$, hence $\{a_1, \ldots, a_s\} \subseteq J_n$, and, by induction hypothesis $J_n \subseteq I_n$, hence $\{a_1, \ldots, a_s\} \subseteq I_n$. Because of $N_m \subseteq I_n$ it follows finally $[I_n, I] \models Bs$, hence $a \in I_{n+1}$. □

Claim 11 implies that a model I of a normal program has a S-founded support function if and only if it is well-supported in the sense of [Fag91].

Obviously, every stable generated model is stable grounded. The converse is not true as the following example shows.

Example 9 *Let $P = \{\Rightarrow a, b, \quad \Rightarrow a, c\}$. The model $\{a, b\}$ is stable grounded, but not stable generated.*

Question 2 *Is every stable generated model of an extended generalized logic program minimally supported ?*

We show that there are stable generated models which are not minimally 1-supported. Are there counter examples for n-supportedness, n a natural number?

Example 10 *P is the following program $\{\Rightarrow a \vee \neg b; \quad \Rightarrow c \vee \neg d; \quad \Rightarrow b\}$. The model $\{a, b\}$ is stable, but not minimally 1-supported, because there is no rule $r \in P$ such that $I \models Br$ and $a \in Lit_m(Hr)$. Obviously, I is minimally 3-supported.*

Summarizing the definitions of this section we introduced four new model operators $\mathrm{Mod}_c^s, \mathrm{Mod}_c^{sf}, \mathrm{Mod}_c^{gs}, \mathrm{Mod}_c^g$, where:

- $\mathrm{Mod}_c^{gs}(S)$ = class of all grounded and supported models of S;

- $\mathrm{Mod}_c^{sf}(S)$ = class of all grounded models of S having a S-grounded support function.

Then, $\mathrm{Mod}_c^s(S) \subseteq Mod_c^{sf}(S) \subseteq Mod_c^{gs}(S) \subset Mod_c^g(S)$. These considerations lead to the following general semantics $\mathrm{Mod}_c^a(S)$ for programs $S \subseteq EGLP$, defining the *acceptable models* of S:

$$\mathrm{Mod}_c^a(S) = \begin{cases} \mathrm{Mod}_c^s(S), \text{ if } \mathrm{Mod}_c^s(S) \neq \emptyset \\ \mathrm{Mod}_c^g(S) \text{ otherwise} \end{cases}$$

The semantics of acceptable models is a natural extension of the stable semantics for programs $S \subseteq EGLP$. In particlar, there are programs without stable models having acceptable models. On the other hand, there are consistent programs without groundedn models. It seems to be not reasonable to weaken the notion of groundedness, since then the directness of the rules are lost.

Example 11 *$S = \{-p \Rightarrow p\}$. $\{p\}$ should be an intended model of S. Obviously, $\{p\}$ is acceptable, but not stable.*
The following program $\{-p \Rightarrow p; -q \Rightarrow -p\}$ has $\{p, q\}$ as its only model which is not grounded.

5 Transformations of Logic Programs

Two programs P and Q are said to be *stable equivalent* iff P and Q have the same stable models. In this section we study the problem whether every program P can be transformed into a stable equivalent program Q whose rules have a certain normal form. For simplicity we assume that $P \subseteq GLP$, the presented proofs work also for the more general case of extended generalized logic programs. A rule $s \in GLP$ has *normal form* if $s := F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$, where every F_i is a disjunction of literals, and every G_j is a literal. s has *special normal* form if s has normal form and every F_i is a literal. We show that every program can be transformed into a stable equivalent program whose rules have normal form. Let $\mathrm{Mod}_s(P)$ the set of all stable models of the gneralized logic program $P \subseteq GLP$. $P \equiv_{st} Q$ iff $\mathrm{Mod}_s(P) = \mathrm{Mod}_s(Q)$. Let $Fin(Seq) = \{X : X \subseteq Seq$ and X is finite $\}$.

Definition 12 (Transformation Rule) *A relation $r \subseteq Fin(Seq) \times Fin(Seq)$ is said to be an admissable transformation rule if r is decidable and for every program $P \subseteq GLP$, and $s \in P$, $\{X, Y\} \subseteq Fin(Seq)$ such that $r(Y, X)$ the condition $P \equiv_{st} (P - Y) \cup X$ is satisfied. Let $R = \{r_1, ..., r_m\}$ be a finite set of admissable transformation rules, and P, Q generalized logic programs. The relation $\longrightarrow_R$ between logic programs is defined as follows: $P \longrightarrow_R Q$ iff there is a rule $r \in R$, $Y \subseteq P$, $X, Y \in Fin(Seq)$ such that $r(Y, X)$ and $Q = (P - Y) \cup X$. Let $\longrightarrow_R^\star$ the transitive closure of $\longrightarrow_R$. P can be transformed into Q by the rule system R if $P \longrightarrow_R^\star Q$.*

Lemma 12 *Let $P \subseteq GLP$, and $s \in P$, $F \equiv \bigwedge Bs$, and $G \equiv \bigvee Hs$. Then $P \equiv_{st} (P - \{s\}) \cup \{Bs \Rightarrow G\} \equiv_{st} (P - \{s\}) \cup \{F \Rightarrow Hs\}$.*

Proof: Let $s_1 := Bs \Rightarrow G$, and $Q := (P - \{s\}) \cup \{Bs \Rightarrow G\}$; we show that $P \equiv_{st} Q$. Let I be a stable generated model of P and $(I_n)_{n<\kappa}$, $\kappa \leq \omega$, a stable chain for I. It is sufficient to show that I_{n+1} is a minimal model of $Hd = I_n \cup \{Hr : r \in Q_{[I_n,I]}\}$. It is easy to show that that Hd has the same Herbrand models than the set $Hd_1 = I_n \cup \{Hr : r \in P_{[I_n,I]}\}$, and from this follows that Hd and Hd_1 have the same minimal models. The other case is proved analogously. □

Lemma 13 *Let $P \subseteq GLP$, and $s \in P$, $s := Bs \Rightarrow F_1 \wedge F_2$, F_1, F_2 open formulas. Then $P \equiv_{st} (P - \{s\}) \cup \{Bs \Rightarrow F_1, Bs \Rightarrow F_2\}$.*

Proof: Let $Q := (P - \{s\}) \cup \{Bs \Rightarrow F_1, Bs \Rightarrow F_2\}$, and $tr(s) = \{Bs \Rightarrow F_1, Bs \Rightarrow F_2\}$. We show that $Mod_s(P) = Mod_s(Q)$.
1) $Mod_s(P) \subseteq Mod_s(Q)$.
Let I be a stable amodel of P and $(I_n)_{n<\omega}$ a stable chain for I. We show that $(I_n)_{n<\omega}$ is Q-stable chain for I. It suffices to prove that I_{n+1} is a minimal model of $I_n \cup \{Hr : r \in Q_{[I_n,I]}\}$. This is implied by the fact that the sets $H_n(Q) = \{Hr : r \in Q_{[I_n,I]}\}$ and $H_n(P) = \{Hr : r \in P_{[I_n,I]}\}$ have the same Herbrand models. Let for $r \in [\{s\}]$ denote $r := Br \Rightarrow F_1^r \wedge F_2^r$. We show the following condition $(\star)$:

For all $r \in [\{s\}]$: $F_1^r \wedge F_2^r \in H_n(P)$ if and only if $\{F_1^r, F_2^r\} \subseteq H_n(Q)$.

If $F_1^r \wedge F_2^r \in H_n(P)$ then $[I_n, I] \models Br$, and hence $\{F_1^r, F_2^r\} \subseteq H_n(Q)$. If $\{F_1^r, F_2^r\} \subseteq H_n(Q)$, then again $[I_n, I] \models Br$ and this implies $F_1^r \wedge F_2^r \in H_n(P)$. The condition $(\star)$ implies that the sets $H_n(P)$ and $H_n(Q)$ have the same Herbrand models and hence the same minimal models.
2) $Mod_s(Q) \subseteq Mod_s(P)$. This condition is proved analogously as 1). □

Corollary 14 *There is a finite set R of admissable transformation rules such that for every finite general logic program P there exists a program Q in normal form such that $P \longrightarrow_R^\star Q$.*

Proof Let P be a generalized program, $s \in P$. We carry out the following steps. $s \longrightarrow \{F \Rightarrow G\}$, where $F := \bigwedge Bs$, $G := \bigvee Hs$, then F is transformed into a logically equivalent conjunctive normal form F_1, and G into a logically equivalent conjunctive normal form $G_1 = C_1 \wedge \ldots C_n$. This can be done by a finite set of admissable transformation rules. By lemma 12 $P \equiv_{st} (P - \{s\}) \cup \{F_1 \Rightarrow G_1\}$. By repeated use of the rule described in lemma 13 we get finally a program Q in normal form being stable equivalent to P. □

Corollary 15 *There is a finite set R of admissable transformation rules such that for every general logic program P there exists a program Q in normal form such that $P \equiv_{st} Q$. If P is recursively enumerable then Q can be chosen to be recursively enumerable.*

Proof: Let $P = \{s_1, s_2, \ldots, s_n, \ldots\}$ be a recursive enumeration of P, and R a finite set of admissable transformation rules satisfying the conditions of corollary 14. Then we construct, according to corollary 14, for every $\{s_i\}$ a set Q_i of rules in normal form such that $\{s_i\} \longrightarrow_R^\star Q_i$. Analogous to the proofs of lemma 12 and lemma **??** one shows that the set $Q = \bigcup_{j<\omega} Q_j$ satisfies the desired condition. $\Box$.

Question 3 *Does there exists a finite set R of admissable transformation rules such that every generalized logic program can be transformed using R into a stable equivalent generalized logic program in special normal form?*

Remark: The following rule $\{F_1 \vee F_2 \Rightarrow H\} \longrightarrow \{F_1 \Rightarrow H, F_2 \Rightarrow H\}$ cannot be used because this rule, though logically true, does not preserve stability.

Example 12 *$P = \{b \vee \neg b \Rightarrow b\}$, and $P_1 = \{b \Rightarrow b; \neg b \Rightarrow b\}$. While P has the stable model $\{b\}$ P_1 has no stable model.*

In example 12 we could replace P by $\{\Rightarrow b\}$. This rule has the form $\{(Bs \Rightarrow Hs, \{\Rightarrow Hs\}) : \bigwedge Bs$ is a tautology $\}$. But also this is rule is not admissable because it is not decidable.

Question 4 *Does there exists for every generalized program P a generalized program Q in special normal form such that $P \equiv_{st} Q$? We may introduce the following relation between different languages of programs. $L_1 \leq L_2$ if for every program over L_1 there is a program over L_2 both having the same stable models. L_1 and L_2 are equivalent w.r.t. to expressive power iff $L_1 \leq L_2$ and $L_2 \leq L_1$.*

6 Conclusion

Generalizing the approach of [HW97] we introduced stable generated models for extended generalized programs implying groundedness and supportedness. Our concept of stable generated models can be used as a foundation of stable model theory (in the context of extended generalized logic programs) and provides a general tool for the classification and the study of the fine structure of the models of a logic program. Grounded models seem to be the best substitute for stable models in those cases where a logic program does not have stable models.

References

[ABW88] K. Apt, H.A. Blair, and A. Walker: Towards a Theory of Declarative Knowledge, in: J. Minker (Ed.), *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, 1988, 89-142.

[DC93] H. Decker and J. C. Casamayor: Sustained Models and Sustained Answers in First-Order Databases, in A. Olive (ed.): *Proc.4^{th} Int'l Workshop on the Deductive Approach to Information Systems and Databases*, Sept. 1993

[Fag91] F. Fages: A New Fixpoint Semantics for General Logic Programs Compared with the Wellfounded and the Stable Model Semantics, *New Generation Computing*, 9-4 (1991).

[GL88] M. Gelfond and V. Lifschitz: The Stable Model Semantics for Logic Programming, *Proc. ICLP 1988*, MIT Press, 1988.

[GL90] M. Gelfond and V. Lifschitz: Logic Programs with Classical Negation, *Proc. ICLP 1990*, MIT Press, 1990.

[GL91] M. Gelfond and V. Lifschitz: Classical Negation in Logic Programs and Disjunctive Databases, *J. New Generation Computing* **9** (1991), 365–385.

[HJW95] H. Herre, J. Jaspars and G. Wagner: Partial Logics with Two Kinds of Negation as a Foundation of Knowledge-Based Reasoning, Report Nr. 12 (1995), Institute fuer Informatik der Universitaet Leipzig; To appear in: in D. Gabbay and H. Wansing (Eds.), *What is Negation ?*, Oxford University Press, 1996.

[HW97] H. Herre and G. Wagner: Stable Models are Generated by a Stable Chain, *Journal of Logic Programming*, **30** , (2): 165-177, 1997

[Mak93] D. Makinson: General Patterns in Nonmonotonic Reasoning; in *Handbook of Logic in Artificial Intelligence and Logic Programming*, ed. D. Gabbay et al., Oxford, 1993

[MC96] N. McCain: Causal Explanation and Disjunctive Default Logic; presented at the *Seminar on Disjunctive Logic Programming and Data Bases: Nonmonotonic Aspects*, Dagstuhl, July 1996

[She88] J.C. Shepherdson:Introduction to the Theory of Logic Programming, *Proc. of the Logic Coll. 1986*, North-Holland, 1988

[Wag91] G. Wagner: A Database Needs Two Kinds of Negation, in: Proc. Third Int. Symp. on Mathematical Fundamentals of Database and Knowledge Based Systems, Springer LNCS 495 (1991), 357–371.