Well-Supported Models of Disjunctive Logic Programs

Martin Baláž
Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava, Slovak Republic
balaz@ii.fmph.uniba.sk

June 1, 2007

Abstract

The stable model semantics [GL88] is considered to be the most accepted semantics for normal logic programs. The well-supported model semantics [Fag90] is another characterization of stable models. It uses the concept of level mappings which can be used to detect cyclic rules.

The notion of stable model was also generalized for logic programs with disjunction [IS93]. We are interested in a characterization of stable models with level mappings in similar way as it was done for logic programs without disjunction.

Keywords: Disjunctive logic program, stable model, well-supported model

1 Introduction

Stable model is defined as a set of atoms derived from default assumptions. If we partially evaluate all default literals in a logic program, a stable model must be a minimal model of the reduced program.

First of all we would like to find some derivation procedure for minimal models of disjunctive logic programs without negation. Then we can assign the level of derivation for every atom in a model and study how stable models are related with those level mappings.

2 Preliminaries

A *language* $\mathcal{L}$ is a countable (possibly infinite) set of *atoms*. A *literal* is an atom A or a *default literal not* A. A *rule* r is a formula of the form $A_1 \vee \cdots \vee A_k \leftarrow A_{k+1} \wedge \cdots \wedge A_m \wedge not\, A_{m+1} \wedge \cdots \wedge not\, A_n$ where $1 \leq k \leq m \leq n$ and A_i, $1 \leq i \leq n$ are atoms. The disjunction $head(r) = A_1 \vee \cdots \vee A_k$ is called the *head of* r and the conjunction $body(r) = A_{k+1} \wedge \cdots \wedge A_m \wedge not\, A_{m+1} \wedge not\, A_n$ is called the *body of* r.

A *disjunctive logic program* (DLP) is a countable set of rules. A *normal logic program* (NLP) does not contain disjunction, it contains only rules with $k = 1$. A *definite logic program* (PLP) is a normal logic program without negation and a *definite disjunctive logic program* (PDLP) is a disjunctive logic program without negation, i.e. for all rules holds $m = n$.

The nodes in the *dependency graph of* a disjunctive logic program are atoms. Two atoms are connected with the edge if exists a rule containing one atom in the head and the other one in the body. A disjunctive logic program is *head cycle free* if does not contain rule with two atoms in the head which appear in the same cycle in the dependency graph.

An *interpretation* I is a member of the power set $\mathcal{P}(\mathcal{L})$ of a language $\mathcal{L}$. I *satisfies* an atom A if $A \in I$ and I *satisfies* a default literal $not\, A$ if $A \notin I$. I *satisfies* a conjunction of literals if it satisfies all literals in the conjunction and I *satisfies* a disjunction of literals if it satisfies some literal in the disjunction. I is a *model* of a rule r if I satisfies the body of r implies I satisfies the head of r.

A rule r in a normal logic program P *supports* an atom A *with respect to* an interpretation I if A is the head of r and I satisfies the body of r. A rule r in a disjunctive logic program P *weakly supports* an atom A *with respect to* an interpretation I if A is in the head of r and I satisfies the body of r. A rule r in a disjunctive logic program P *supports* an atom A with respect to an interpretation I if r weakly supports A w.r.t. I and for all atoms B in the head of r holds I satisfies B implies B equals A.

I is a *model* of a logic program P if it satisfies all rules in P. A model I is *minimal* if does not exists a model J such that $J \subseteq I$ implies $J = I$. A model I is *least* if for all models J holds $I \subseteq J$. $MM(P)$ denotes the set of all minimal models of a logic program P.

A *program reduct* of a logic program P with respect to an interpretation I is a logic program obtained from P by deleting all rules with default literal in the body not satisfied by I and by deleting all remaining default literals. I is a *stable model* of P if I is a minimal model of the program reduct of P with respect to I. $SM(P)$ denotes the set of all stable models of a logic program P.

3 Minimal Models

The set of minimal models is generally accepted as the semantics of logic programs without negation. In the following section we try to find a derivation procedure to compute minimal models.

3.1 Definite Logic Programs

In [vEK76] authors showed that there always exists the least model of a PLP and it can be obtained from the empty interpretation by iterating the immediate consequence operator.

Definition 1 (Immediate Consequence Operator)**.** Let I be an interpretation of a PLP P. The *immediate consequence operator* T_P and its iteration $T_P \uparrow \alpha$ are defined as follows:

$$T_P(I) = \{head(r) \mid r \in P, I \models body(r)\}$$

$$T_P \uparrow \alpha = \begin{cases} \emptyset & \text{if } \alpha \text{ is } 0 \\ T_P(T_P \uparrow \beta) & \text{if } \alpha \text{ is a successor ordinal of } \beta \\ \bigcup\limits_{\beta<\alpha} T_P \uparrow \beta & \text{if } \alpha \text{ is a limit ordinal} \end{cases}$$

An interpretation I is a fixpoint of the operator T_P if $I = T_P(I)$. It is a minimal fixpoint if it is a fixpoint and for all fixpoints $J \subseteq I$ holds $J = I$.

Proposition 1. *An interpretation I is a minimal model of a PLP P iff I is a minimal fixpoint of T_P.*

Proposition 2. *Let P be a PLP. Then $T_P \uparrow \omega$ is the least model of P.*

3.2 Definite Disjunctive Logic Programs

There are more choices to satisfy a rule with disjunction in the head. In general a PDLP can have more minimal models. We present the nondeterministic immediate consequence operator introduced in [PT02].

Definition 2 (Nondeterministic Immediate Consequence Operator)**.** Let I be an interpretation and P be a PDLP. The *nondeterministic immediate consequence operator* N_P is defined as follows:

$$N_P(I) = MM(\{head(r) \mid r \in P, I \models body(r)\})$$

An interpretation I is a fixpoint of the operator N_P if $I \in N_P(I)$. It is a minimal fixpoint if it is a fixpoint and for all fixpoints $J \subseteq I$ holds $J = I$.

Proposition 3. *An interpretation I is a minimal model of a PDLP P iff I is a minimal fixpoint of N_P.*

Example 1.

$$\begin{array}{rcllcl} P & = & \{ & a \vee b \vee c & \leftarrow & \\ & & & b & \leftarrow & a \\ & & & c & \leftarrow & b \\ & & & a & \leftarrow & c \quad \} \end{array}$$

We show that the nondeterministic immediate consequence operator N_P can not be directly used to compute the minimal model $M = \{a, b, c\}$ by its iteration. It holds that $N_P(\emptyset) = \{\{a\}, \{b\}, \{c\}\}$ and $N_P(\{a\}) = \{\{b\}\}$, $N_P(\{b\}) = \{\{c\}\}$, $N_P(\{c\}) = \{\{a\}\}$. Every iteration of the operator N_P would oscillate and not converge to M.

However if we do not drop intermediate results there are three possible ways how to compute M. If we start with the empty interpretation $\emptyset$, we have three choices how to satisfy the rule $a \vee b \vee c \leftarrow$. We choose either the interpretation $\{a\}$, $\{b\}$ or $\{c\}$. Then we directly compute the rest of the model M by applying the remaining rules. All three computations $\emptyset, \{a\}, \{a, b\}, \{a, b, c\}$; $\emptyset, \{b\}, \{b, c\}, \{a, b, c\}$ and $\emptyset, \{c\}, \{a, c\}, \{a, b, c\}$ terminate in the same model M.

Example 2.

$$\begin{array}{rcllcl} P & = & \{ & a \vee b & \leftarrow & \\ & & & c & \leftarrow & \\ & & & a & \leftarrow & c \\ & & & b & \leftarrow & c \quad \} \end{array}$$

In the previous example we have seen that disjunctive rules cause splitting of the computation. Also in this example both computations $\emptyset, \{a, c\}, \{a, b, c\}$ and $\emptyset, \{b, c\}, \{a, b, c\}$ terminates in the same minimal model $M = \{a, b, c\}$. However if we do not apply the disjunctive rule $a \vee b \leftarrow$ too soon, the computation $\emptyset, \{c\}, \{a, b, c\}$ terminates in M in deterministic way.

3.3 AND/OR Graphs

Definition 3 (AND/OR Graph)**.** The *AND/OR graph of* a PDLP P in a language $\mathcal{L}$ is a directed acyclic graph $G = (V, E)$. The set of vertices V contains

- *AND-nodes* $(I, R) \in \mathcal{P}(\mathcal{L}) \times \mathcal{P}(P)$ where $\emptyset \subset R \subseteq \{r \in P \mid I \not\models r\}$.
- *OR-nodes* $I \in \mathcal{P}(\mathcal{L})$ where $I \not\models P$
- *terminals* $I \in \mathcal{P}(\mathcal{L})$ where $I \models P$

There is an edge between

- OR-node I and AND-node (I, R)
- AND-node (I, R) and OR-node or terminal J if $J \in \{I \cup M \mid M \in MM(\{head(r) \mid r \in R\})\}$

Terminals are models we want to reach from the empty interpretation. In an OR-node we choose a set of rules to satisfy in the next step. In general we don't choose all not satisfied rules to avoid unnecessary splitting (see the example 2). It is called an OR-node because it is enough to choose only one set of rules. In an AND-node we must check if from every minimal expansion satisfying chosen rules we reach the same terminal. It is called an AND-node because we must check all such minimal expansions.

Definition 4 (Computation)**.** A *computation* in the AND/OR graph $G = (V, E)$ of a PDLP P is a sequence of interpretations $\{I_\alpha\}_{\alpha=0}^{\gamma}$ such that

- $I_\alpha = \emptyset$ if α is 0
- $((I_\beta, R_\beta), I_\alpha) \in E$ for some $R_\beta \in \mathcal{P}(P)$ if α is a successor ordinal of β
- $I_\alpha = \bigcup\limits_{\beta<\alpha} I_\beta$ if α is a limit ordinal

A computation $\{I_\alpha\}_{\alpha=0}^{\gamma}$ *terminates in* an interpretation I *after at most* δ steps if $I = \bigcup\limits_{\alpha=0}^{\gamma} I_\alpha$ is a terminal and $\gamma \leq \delta$.

Definition 5 (Derivation)**.** A *derivation* of an interpretation M is a minimal subgraph $G' = (V', E')$ of the AND/OR graph $G = (V, E)$ of a PDLP P such that

- $\emptyset \in V'$
- if $I \in V'$ is an OR-node then there exist an AND-node $(I, R) \in V'$ and an edge $(I, (I, R)) \in E'$
- if $(I, R) \in V'$ is an AND-node then $J \in V'$ and $((I, R), J) \in E'$ for all $((I, R), J) \in E$, $J \subseteq M$
- the set of OR-nodes and terminals is *closed*, i.e. if $\{I_\alpha\}_{\alpha=0}^{\gamma}$ is a computation in G' then $\bigcup\limits_{\alpha=0}^{\gamma} I_\alpha \in V'$

A derivation G' *terminates in* an interpretation I *after at most* δ steps if all terminating computations in G' terminates in I after at most δ steps.

Proposition 4. *If I is a minimal model of a PDLP P then there exists a derivation of I in the AND/OR graph of P terminating in I after at most ω steps.*

Proposition 5. *Let I be an interpretation and P be a PDLP. Then I is a minimal model of P if exists a derivation of I in the AND/OR graph of P terminating in I.*

Proposition 6. *An interpretation I is a minimal model of a HCF PDLP P if exists a derivation of I in the AND/OR graph of P terminating in I after at most ω steps containing just one terminating computation.*

4 Well-Supported Models

A level mapping assigns to each atom its level of derivation. Intuitively an atom in the head can be derived only when all literals in the body were already derived.

4.1 Normal Logic Programs

Definition 6 (Well-Support)**.** Let I be an interpretation, P be a NLP and ℓ be a level mapping. A rule $r \in P$ *well-supports* an atom $A \in I$ with respect to I and ℓ if r supports A with respect to I and

- r supports A with respect to I
- $\forall L \in body(r)\colon \ell(A) > \ell(L)$

A model I of a NLP P is *well-supported* with respect to a level mapping ℓ if for all atoms $A \in I$ exists a rule $r \in P$ well-supporting A with respect to I and ℓ.

Proposition 7. *Let I be an interpretation and P be a NLP. Then I is a stable model of P iff exists a level mapping ℓ such that I is a well-supported model of P with respect to ℓ.*

4.2 Disjunctive Logic Programs

If the rules can contain the disjunction in the head, we should distinguish two kinds of support [BD95]. The weaker form of support does not require the supported atom to be the only one satisfied atom in the head as it is in the case of stronger form. The reason are possible cycles in the head of a rule. But what we require is that the well-supported atom is derived as the first atom in the head. The consequence is that a rule can well-support at most one atom.

Definition 7 (Well-Support). Let I be an interpretation, P be a DLP and ℓ be a level mapping. A rule $r \in P$ *weakly well-supports* an atom $A \in I$ with respect to I and ℓ if

- r weakly supports A with respect to I
- $\forall L \in \mathit{body}(r)\colon \ell(A) > \ell(L)$
- $\forall B \in \mathit{head}(r)\colon I \models B, B \neq A \implies \ell(B) > \ell(A)$

A rule $r \in P$ *well-supports* an atom $A \in I$ with respect to I and ℓ if

- r supports A with respect to I
- $\forall L \in \mathit{body}(r)\colon \ell(A) > \ell(L)$

A model I of a DLP P is *(weakly) well-supported* with respect to a level mapping ℓ if for all atoms $A \in I$ exists a rule $r \in P$ (weakly) well-supporting A with respect to I and ℓ.

Proposition 8. *Let I be a model of a DLP P. Let $\{I_\alpha\}_{\alpha=0}^{\gamma}$ be a computation in a derivation of I terminating in I in the AND/OR graph of P^I. Let ℓ be a level mapping such that $\ell(A) = \alpha$ if $A \in I_{\alpha+1} \setminus I_\alpha$, otherwise $\ell(A) = 0$. Then I is a weakly well-supported model of P with respect to ℓ.*

Proposition 9. *Let I be an interpretation and P be a HCF DLP. Then I is a stable model of P iff exists a level mapping ℓ such that I is a well-supported model of P with respect to ℓ.*

5 Conclusions

We have showed how minimal models can be derived in AND/OR graphs of logic programs without negation. We have seen that disjunction in the head can cause splitting of computations. The splitting is necessary only if there is a cycle in the head of a rule. Finally we have showed how stable models of disjunctive logic programs are related with (weakly) well-supported models.

References

[BD95] Stefan Brass and Jürgen Dix. Characterizations of the stable semantics by partial evaluation. In *Proceedings of the 3rd International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR'95*, pages 85–98. Springer-Verlag, 1995.

[Fag90] François Fages. A new fixpoint semantics for general logic programs compared with the well-founded and the stable model semantics. In *Proceedings of the 7th International Conference on Logic Programming, ICLP'90*, pages 442–458. MIT Press, 1990.

[GL88] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proceedings of the 5th International Conference on Logic Programming, ICLP'88*, pages 1070–1080. MIT Press, 1988.

[IS93] Katsumi Inoue and Chiaki Sakama. Transforming abductive logic programs to disjunctive programs. In *Proceedings of the 10th International Conference on Logic Programming, ICLP'93*, pages 335–353. MIT Press, 1993.

[PT02] Nikolay Pelov and Mirosław Truszczyński. Semantics of disjunctive programs with monotone aggregates - an operator-based approach. In *Proceedings of the 10th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR 2002*, pages 327–334. Springer-Verlag, 2002.

[vEK76] Maarten van Emden and Robert Anthony Kowalski. The semantics of predicate logic as a programming language. *Journal of the ACM*, 23(4):733–742, 1976.

Martin Baláž (RNDr) is a lecturer at the Faculty of Mathematics, Physics and Informatics of the Comenius University in Bratislava. His PhD-thesis supervisor is assistant professor PhDr. Ján Šefránek, CSc.