Capturing Parallel Circumscription with Disjunctive Logic Programs ⋆

Tomi Janhunen and Emilia Oikarinen

Helsinki University of Technology, Laboratory for Theoretical Computer Science
P.O. Box 5400, FIN-02015 HUT, Finland
`Tomi.Janhunen@hut.fi`, `Emilia.Oikarinen@hut.fi`

Abstract. The stable model semantics of disjunctive logic programs is based on classical models which are minimal with respect to subset inclusion. As a consequence, every atom appearing in a disjunctive program is false by default. This is sometimes undesirable from the knowledge representation point of view and a more refined control of minimization is called for. Such features are already present in Lifschitz's parallel circumscription where certain atoms are allowed to vary or to have fixed values while all other atoms are minimized. In this paper, it is formally shown that the expressive power of minimal models is properly increased in the presence of varying atoms. In spite of this, we show how parallel circumscription can be embedded into disjunctive logic programming in a relatively systematic fashion using a linear and faithful, but *non-modular* translation. This enables the conscious use of varying atoms in disjunctive logic programs — leading to more elegant and concise problem representations in various domains.

1 Introduction

In disjunctive logic programming, a rule-based language which allows disjunctions in the heads of rules is used for knowledge representation. Along the development of efficient implementations such as dlv [15] and GnT [13], various problems have been formalized as disjunctive logic programs. The semantics of disjunctive logic programs is determined by *stable models* [8, 20] which are minimal with respect to subset inclusion. This makes every atom appearing in a disjunctive logic program false by default. In many cases, this is highly desirable, but certain problems become awkward to formalize if all atoms are blindly subject to minimization. This suggests a revision of the stable model semantics in order to incorporate atoms that are not false by default.

The need of atoms, which are not subject to minimization, has already been realized in conjunction with *normal logic programs* which form a special case of disjunctive logic programs. Simons [23] introduces *choice rules* which allow the definition of atoms not being false by default. The same effect can be obtained by allowing negation as failure in the heads of disjunctive rules [9]: a rule of the form $a \vee {\sim}a$ represents the fact that a can be true or false. As shown by the first author [11], negation as failure can be removed from the heads of disjunctive rules using a linear transformation. This implies

⋆ The research reported in this paper has been partially funded by the Academy of Finland (project #53695) and the European Commission (contract IST-FET-2001-37004).

that choice rules can be effectively expressed using disjunctive rules. However, it is important to realize that atoms definable in this way are essentially *fixed* atoms in the sense of *parallel circumscription* [16] which is based on a refined notion of minimality.

In addition to fixed atoms and those subject to minimization, parallel circumscription incorporates yet another category of atoms, namely atoms that are allowed to *vary*. As demonstrated by Lifschitz's ostrich example [16], varying atoms tend to increase the knowledge representation capabilities of ordinary circumscription [18] where all atoms are subject to minimization. Unfortunately, varying atoms are not yet well-supported in disjunctive logic programming, although serious attempts to embed parallel circumscription into disjunctive logic programming have already been made. The approach by Gelfond and Lifschitz [7] is restricted to the stratifiable case and the one by Sakama and Inoue [22] involves *characteristic clauses* which imply an exponential time/space complexity in the worst case. Quite recently, Lee and Lin [14] characterize parallel circumscription in terms of *loop formulas* and then embed parallel circumscription in disjunctive logic programming using them. However, the number of loops can be exponential in the worst case. Thus it remains open whether an efficient translation from parallel circumscription into disjunctive logic programs is feasible in the general case.

The goal of this paper is to develop such a translation — enabling the conscious use of varying atoms in disjunctive logic programs. We proceed as follows. In Section 2, we review the syntax and semantics of disjunctive logic programs and present the notion of *visible equivalence* to enable natural comparisons of programs. Then the effects of varying and fixed atoms on the expressiveness of positive disjunctive programs are studied in Section 3. The key result is that varying atoms lead to a proper increase in expressive power which we believe to explain the above mentioned difficulties in translating parallel circumscription. A linear but non-modular translation for removing varying atoms is presented in Section 4. This paper is concluded by Section 5 where we also sketch potential applications of varying atoms in disjunctive logic programming.

2 Disjunctive Logic Programs Revisited

In this section, we review the basic concepts of disjunctive logic programming in the propositional case. A *disjunctive logic program* (DLP) Π is a set of *rules* of the form

$$a_1 \lor \cdots \lor a_n \leftarrow b_1, \ldots, b_m, {\sim}c_1, \ldots, {\sim}c_k, \tag{1}$$

where $n, m, k \geq 0$ and $a_1, \ldots, a_n$, $b_1, \ldots, b_m$, and $c_1, \ldots, c_k$ are propositional atoms. The *head* of the rule $a_1 \lor \cdots \lor a_n$ is interpreted disjunctively while the rest forming the *body* of the rule is interpreted conjunctively. The symbol "$\sim$" denotes *negation as failure to prove*; or *default negation* for short. Intuitively, a rule of the form (1) acts as an inference rule: any of the head atoms $a_1, \ldots, a_n$ can be inferred given that the positive body atoms $b_1, \ldots, b_m$ can be inferred and the negative body atoms $c_1, \ldots, c_k$ cannot.

We define *literals* in the standard way using $\sim$ as the connective for negation. For any set of atoms A, we define a set of negative literals ${\sim}A = \{{\sim}a \mid a \in A\}$. Since the order of atoms is insignificant in a rule (1), we use a shorthand $A \leftarrow B, {\sim}C$ where A, B and C are the sets of atoms involved in (1). If necessary, we separate rules with full stops and we drop the symbol "$\leftarrow$" in case of an empty body. An empty head ($n = 0$) is denoted by "$\bot$" and a rule with an empty head is called an *integrity constraint*. A DLP Π is *positive* if and only if $k = 0$ holds for every rule (1) of Π. We remind the reader that positive DLPs (PDLPs) can be viewed as propositional theories in conjunctive normal form (CNF) which can be obtained in linear time using new atoms.

2.1 Semantics: Minimal and Stable Models

We define the *Herbrand base* $\mathrm{Hb}(\Pi)$ of a DLP Π as a set of atoms which contains all atoms appearing in Π. Due to flexibility of this definition, we view $\mathrm{Hb}(\Pi)$ as the *symbol table* of Π so that it contributes to the length of Π in symbols, denoted by $\|\Pi\|$. Following the ideas from [12], we partition $\mathrm{Hb}(\Pi)$ into two parts $\mathrm{Hb}_{\mathrm{v}}(\Pi)$ and $\mathrm{Hb}_{\mathrm{h}}(\Pi)$ which determine the *visible* and the *hidden* parts of $\mathrm{Hb}(\Pi)$, respectively. The visibility of atoms becomes important in Section 2.2 where the equivalence of DLPs is of interest, but for now we concentrate on defining the semantics of propositional DLPs.

An *interpretation* $I \subseteq \mathrm{Hb}(\Pi)$ of Π determines which atoms $a \in \mathrm{Hb}(\Pi)$ are true ($a \in I$) and which are false ($a \notin I$). An interpretation I is a (classical) *model* of Π, denoted by $I \models \Pi$, if and only if for every rule $A \leftarrow B, \sim C$ of Π, $B \subseteq I$ and $C \cap I = \emptyset$ imply $A \cap I \neq \emptyset$, i.e. the satisfaction of the rule body implies that one of the head atoms must also be true. It is customary to distinguish *minimal models* of a DLP Π, i.e. models $M \models \Pi$ for which there are no other models $N \models \Pi$ such that $N \subset M$. The set of minimal models of Π is denoted by $\mathrm{MM}(\Pi)$. If Π is a positive DLP, then $\mathrm{MM}(\Pi)$ determines the standard minimal model semantics of Π. Unfortunately, minimal models do not properly capture intuitions behind DLPs involving default negation, but *stable models* [8, 20] provide a reasonable semantics for such programs.

Definition 1. *Given a DLP Π and an interpretation $M \subseteq \mathrm{Hb}(\Pi)$, the Gelfond-Lifschitz reduct of Π is a positive DLP*

$$\Pi^M = \{A \leftarrow B \mid A \leftarrow B, \sim C \in \Pi \text{ and } M \cap C = \emptyset\}. \tag{2}$$

An interpretation $M \subseteq \mathrm{Hb}(\Pi)$ is a stable model of Π if and only if $M \in \mathrm{MM}(\Pi^M)$.

Given a DLP Π, we let $\mathrm{SM}(\Pi)$ denote the set of stable models of Π. Any two DLPs Π and Π' are considered to be *equivalent* under the stable model semantics, denoted by $\Pi \equiv \Pi'$, if and only if $\mathrm{SM}(\Pi) = \mathrm{SM}(\Pi')$. For instance, we have $\Pi \equiv \Pi'$ for $\Pi = \{a \vee b.\ \}$ and $\Pi' = \{a \leftarrow \sim b.\ b \leftarrow \sim a.\ \}$, as $\mathrm{SM}(\Pi) = \{\{a\}, \{b\}\} = \mathrm{SM}(\Pi')$. The preceding definition of $\equiv$ is justifiable from the viewpoint of formalizing a problem at hand as a DLP Π: the stable models of the program Π are often supposed to be in a one-to-one correspondence with the solutions of the problem. If $\Pi \equiv \Pi'$ holds for two programs $\Pi \neq \Pi'$ formalizing the same problem, then the same solutions are obtained.

2.2 Visible Equivalence

A drawback of the relation $\equiv$ is that it does not take the visibility of atoms into account. It is typical that a DLP Π contains atoms formalizing certain auxiliary concepts local to Π. Such atoms carry little relevance for other programs. This is why we adopt a slightly more general notion of equivalence [12] which treats the visible part $\mathrm{Hb_v}(\Pi)$ of the Herbrand base $\mathrm{Hb}(\Pi)$ as the *program interface* of Π. The key idea is that the hidden atoms in $\mathrm{Hb_h}(\Pi) = \mathrm{Hb}(\Pi) \setminus \mathrm{Hb_v}(\Pi)$ can be viewed local to Π and hence negligible as far as the equivalence of Π with other programs is concerned. The definition below is given relative to the sets of interpretations $\mathrm{SEM}(\Pi)$ and $\mathrm{SEM}(\Pi')$ which determine the semantics of Π and Π', respectively. We need this kind of flexibility in Section 3 when we compare PDLPs which are (possibly) based on different semantics than the stable semantics. The reader may assume $\mathrm{SEM}(\Pi) = \mathrm{SM}(\Pi)$ unless otherwise stated.

Definition 2. *Two DLPs Π and Π' are* visibly equivalent*, denoted by $\Pi \equiv_{\mathrm{v}} \Pi'$, if and only if $\mathrm{Hb_v}(\Pi) = \mathrm{Hb_v}(\Pi')$ and there is a bijection $f : \mathrm{SEM}(\Pi) \to \mathrm{SEM}(\Pi')$ such that for all interpretations $M \in \mathrm{SEM}(\Pi)$, $M \cap \mathrm{Hb_v}(\Pi) = f(M) \cap \mathrm{Hb_v}(\Pi')$.*

It is easy to verify that $\equiv_{\mathrm{v}}$ is an equivalence relation. To compare $\equiv_{\mathrm{v}}$ with $\equiv$, we note that these two relations coincide given that $\mathrm{Hb_h}(\Pi) = \mathrm{Hb_h}(\Pi') = \emptyset$ and $\mathrm{Hb}(\Pi) = \mathrm{Hb}(\Pi')$. The latter condition is actually of little account, as it can be readily satisfied e.g. by extending Herbrand bases with "useless" rules of the form $a \leftarrow a$.

Example 1. Consider logic programs $\Pi = \{a \leftarrow b.\ a \leftarrow c.\ b \leftarrow {\sim}c.\ c \leftarrow {\sim}b.\ \}$ and $\Pi' = \{a \leftarrow d, {\sim}e.\ a \leftarrow e, {\sim}d.\ d \vee e.\ \}$ with $\mathrm{Hb_v}(\Pi) = \mathrm{Hb_v}(\Pi') = \{a\}$. The stable models of Π are $M_1 = \{a, b\}$ and $M_2 = \{a, c\}$ whereas for Π' they are $N_1 = \{a, d\}$ and $N_2 = \{a, e\}$. Thus $\Pi \not\equiv \Pi'$ is clearly the case, but we have a bijection $f : \mathrm{SM}(\Pi) \to \mathrm{SM}(\Pi')$ which maps M_i to N_i for $i \in \{1, 2\}$. Hence $\Pi \equiv_{\mathrm{v}} \Pi'$. $\square$

3 Parallel Circumscription and Its Expressive Power

In this section, we analyze the expressive power of Lifschitz's *parallel circumscription* [16] by studying the effects of denying varying atoms and/or fixed atoms on the expressiveness of minimal models. In analogy to Section 2, we formulate parallel circumscription in the propositional case. Rather than using arbitrary propositional sentences to formulate propositional theories, we assume that the syntax of PDLPs is used. As discussed already in the introduction, parallel circumscription is based on a notion of minimality which partitions atoms in three disjoint categories.

Definition 3. *Let Π be a PDLP and let $V \subseteq \mathrm{Hb}(\Pi)$ and $F \subseteq \mathrm{Hb}(\Pi)$ be two sets of atoms satisfying $V \cap F = \emptyset$. A model $M \models \Pi$ is $\langle V, F\rangle$-minimal $\iff$ $\nexists N \models \Pi$ such that (i) $N \setminus (V \cup F) \subset M \setminus (V \cup F)$ and (ii) $N \cap F = M \cap F$.*

The idea is that the atoms in $\mathrm{Hb}(\Pi) \setminus (V \cup F)$ are subject to minimization in analogy to Section 2.1. However, while such a minimization takes place, the truth values of the atoms in V may *vary* freely and the truth values of the atoms in F are kept *fixed*. The set of all $\langle V, F\rangle$-minimal models of Π is denoted by $\mathrm{MM}_{V,F}(\Pi)$. It is customary in disjunctive logic programming that all atoms are subject to minimization, i.e. $\langle\emptyset, \emptyset\rangle$-minimal models of a positive DLP Π are of interest. Under this restriction, the first condition of Definition 3 is equivalent to $N \subset M$ while the second condition becomes void. Thus $\mathrm{MM}(\Pi) = \mathrm{MM}_{\emptyset,\emptyset}(\Pi)$. In the sequel, we are interested in the problem of determining $\langle V, F\rangle$-minimal models for a given *positive* DLP Π. Note that $V \subseteq$

$\mathrm{Hb}(\Pi)$ and $F \subseteq \mathrm{Hb}(\Pi)$ are separately specified for each program Π and are thus viewed as parts of the respective programs. For now, we concentrate on answering the following question: is it possible to remove fixed and varying atoms by translating a PDLP involving such atoms into another PDLP not containing such atoms?

3.1 PFM Translation Functions

To answer the preceding question, we apply an analysis method [11, 12] which is based on the existence of polynomial, faithful and modular translation functions between classes of logic programs. These properties are formalized in Definition 5 below, but first we state conditions on which two DLPs Π and Π' are viewed as separate program modules that can be combined together to form a larger program $\Pi \sqcup \Pi'$.[1]

Definition 4. *Two PDLPs Π and Π' satisfy module conditions if and only if $\Pi \cap \Pi' = \emptyset$, $\mathrm{Hb_v}(\Pi) = \mathrm{Hb_v}(\Pi')$, $\mathrm{Hb_h}(\Pi) \cap \mathrm{Hb}(\Pi') = \emptyset$, $\mathrm{Hb}(\Pi) \cap \mathrm{Hb_h}(\Pi') = \emptyset$.*

The intuition behind the conditions listed in Definition 4 is that the program modules Π and Π' possess identical program interfaces for mutual interaction and they do not share rules nor hidden atoms. If Π and Π' share rules, then $\Pi \setminus \Pi'$, $\Pi' \setminus \Pi$, and $\Pi \cap \Pi'$ might be identified as disjoint program modules, if admitted by the other conditions.

Definition 5. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two classes of logic programs. A translation function $\mathrm{Tr} : \mathcal{C} \to \mathcal{C}'$ is defined to be*

1. ***polynomial**, iff for all programs $\Pi \in \mathcal{C}$, the translation $\mathrm{Tr}(\Pi) \in \mathcal{C}'$ can be computed in time (and hence also space) polynomial to $\|\Pi\|$;*
2. ***faithful**, iff for all programs $\Pi \in \mathcal{C}$, $\Pi \equiv_{\mathrm{v}} \mathrm{Tr}(\Pi)$;*
3. ***modular**, iff for all programs $\Pi \in \mathcal{C}$ and $\Pi' \in \mathcal{C}$ satisfying module conditions, the translation $\mathrm{Tr}(\Pi \sqcup \Pi') = \mathrm{Tr}(\Pi) \sqcup \mathrm{Tr}(\Pi')$ where the translations $\mathrm{Tr}(\Pi)$ and $\mathrm{Tr}(\Pi')$ satisfy module conditions.*

It can be shown that these three properties are preserved under compositions [12]. In particular, the modularity condition differs from the one used in [11]. This is to support translation functions between classes of logic programs (or like) that do not share syntax. Moreover, the module conditions in Definition 4 are more liberal than those used by Eiter et al. [6] which enables richer interaction between program modules.

In the sequel, we use the existence of a polynomial, faithful and modular (PFM) translation function as a criterion when comparing classes of logic programs by expressive power. A class of logic programs $\mathcal{C}$ is *at least as expressive as* another class $\mathcal{C}'$ iff there is a PFM translation function $\mathrm{Tr} : \mathcal{C}' \to \mathcal{C}$. We write $\mathcal{C}' \leq_{\mathrm{PFM}} \mathcal{C}$ to denote such a relationship. If both $\mathcal{C} \leq_{\mathrm{PFM}} \mathcal{C}'$ and $\mathcal{C}' \leq_{\mathrm{PFM}} \mathcal{C}$ hold, then $\mathcal{C}$ and $\mathcal{C}'$ are regarded as *equally expressive* classes, denoted by $\mathcal{C} =_{\mathrm{PFM}} \mathcal{C}'$. In certain cases, we succeed to find a counter-example to establish a negative relationship $\mathcal{C} \not\leq_{\mathrm{PFM}} \mathcal{C}'$.[2] If, in addition, $\mathcal{C}' \leq_{\mathrm{PFM}} \mathcal{C}$ holds, then $\mathcal{C}$ is *strictly* more expressive than $\mathcal{C}'$, denoted by $\mathcal{C}' <_{\mathrm{PFM}} \mathcal{C}$. Finally, two classes may also turn out to be *incomparable* in terms of PFM translation functions, denoted by $\mathcal{C} \neq_{\mathrm{PFM}} \mathcal{C}'$, if and only if both $\mathcal{C}' \not\leq_{\mathrm{PFM}} \mathcal{C}$ and $\mathcal{C} \not\leq_{\mathrm{PFM}} \mathcal{C}'$ hold.

[1] The symbol $\sqcup$ denotes disjoint union.

[2] Sometimes we do not need all the three properties to form a counter-example and we may drop the respective letters from the notation. E.g. $\mathcal{C} \not\leq_{\mathrm{FM}} \mathcal{C}'$ implies $\mathcal{C} \not\leq_{\mathrm{PFM}} \mathcal{C}'$ in general.

3.2 Expressiveness Analysis

In this section, we apply the classification method presented in Section 3.1 to analyze $\mathcal{D}^{+}_{\mathrm{mvf}}$ which is defined as the class of PDLPs involving atoms being minimized (m), varying atoms (v), and fixed atoms (f). The semantics of a PDLP $\varPi$ from this class is determined by $\mathrm{MM}_{V,F}(\varPi)$ rather than $\mathrm{SM}(\varPi) = \mathrm{MM}(\varPi)$; recall that $\varPi$ has the sets $V \subseteq \mathrm{Hb}(\varPi)$ and $F \subseteq \mathrm{Hb}(\varPi)$ associated with it. We obtain six subclasses of $\mathcal{D}^{+}_{\mathrm{mvf}}$ by insisting that one or two of the sets $\mathrm{Hb}(\varPi) \setminus (V \cup F)$, V, and F are empty for PDLPs included in the subclass. Such a restriction corresponds to denying minimized/varying/fixed atoms and we drop the corresponding letter(s) from the notation when referring to the respective subclass of $\mathcal{D}^{+}_{\mathrm{mvf}}$. For instance, $\mathcal{D}^{+}_{\mathrm{m}}$ denotes the class of PDLPs under the standard semantics according to which all atoms are subject to minimization, i.e. the sets V and F are both empty for all PDLPs $\varPi$ within this class.

We begin the analysis with fixed atoms. It is a well-known fact that they can be eliminated in general [3], but our interest in this respect is to check that the elimination can be accomplished using a PFM translation function.

Theorem 1. *$\mathcal{D}^{+}_{\mathrm{mfv}} \leq_{\mathrm{PFM}} \mathcal{D}^{+}_{\mathrm{mv}}$ and $\mathcal{D}^{+}_{\mathrm{mf}} \leq_{\mathrm{PFM}} \mathcal{D}^{+}_{\mathrm{m}}$.*

Proof. (sketch) Let $\varPi$ be a PDLP, and V and F the sets of varying and fixed atoms, respectively. The class $\mathcal{D}^{+}_{\mathrm{mf}}$ can be covered by further assuming $V = \emptyset$. De Kleer and Konolige [3] propose the following technique to remove F. A new atom $f' \notin \mathrm{Hb}(\varPi)$ is introduced for each $f \in F$. The translation $\mathrm{Tr}_{\mathrm{KK}}(\varPi) = \varPi \cup \{f \vee f'.\ \bot \leftarrow f, f'.\ \mid f \in F\}$ with the set of atoms $(\mathrm{Hb}(\varPi) \setminus V) \cup \{f' \mid f \in F\}$ subject to minimization. The visible Herbrand base $\mathrm{Hb}_{\mathrm{v}}(\mathrm{Tr}_{\mathrm{KK}}(\varPi))$ can be defined as $\mathrm{Hb}_{\mathrm{v}}(\varPi)$.

It is easy to see that $\mathrm{Tr}_{\mathrm{KK}}$ is linear. For the faithfulness of $\mathrm{Tr}_{\mathrm{KK}}$, we note that $\langle V, F\rangle$-minimal models M of $\varPi$ are in a bijective relationship with the $\langle V, \emptyset\rangle$-minimal models $M' = M \cup \{f' \mid f \in F \text{ and } f \notin M\}$ of $\mathrm{Tr}_{\mathrm{KK}}(\varPi)$. For the modularity of $\mathrm{Tr}_{\mathrm{KK}}$, we suppose that two PDLPs $\varPi$ and $\varPi'$ with the sets of varying atoms V and V' and the sets of fixed atoms F and F', respectively, satisfy the module conditions. It is clear that $\mathrm{Tr}_{\mathrm{KK}}(\varPi)$ and $\mathrm{Tr}_{\mathrm{KK}}(\varPi')$ are disjoint and $\mathrm{Tr}_{\mathrm{KK}}(\varPi \sqcup \varPi') = \mathrm{Tr}_{\mathrm{KK}}(\varPi) \sqcup \mathrm{Tr}_{\mathrm{KK}}(\varPi')$, as $\varPi$ and $\varPi'$ as well as F and F' are disjoint by the module conditions. Moreover, we have $\mathrm{Hb}_{\mathrm{v}}(\mathrm{Tr}_{\mathrm{KK}}(\varPi)) = \mathrm{Hb}_{\mathrm{v}}(\mathrm{Tr}_{\mathrm{KK}}(\varPi'))$ by definition, because $\mathrm{Hb}_{\mathrm{v}}(\varPi) = \mathrm{Hb}_{\mathrm{v}}(\varPi')$ by the module conditions. Finally, the translations $\mathrm{Tr}_{\mathrm{KK}}(\varPi)$ and $\mathrm{Tr}_{\mathrm{KK}}(\varPi')$ do not share hidden atoms as the modules $\varPi$ and $\varPi'$ do not. $\square$

Thus $\mathcal{D}^{+}_{\mathrm{mv}} \subseteq \mathcal{D}^{+}_{\mathrm{mfv}}$ and $\mathcal{D}^{+}_{\mathrm{m}} \subseteq \mathcal{D}^{+}_{\mathrm{mf}}$ imply $\mathcal{D}^{+}_{\mathrm{mv}} =_{\mathrm{PFM}} \mathcal{D}^{+}_{\mathrm{mfv}}$ and $\mathcal{D}^{+}_{\mathrm{m}} =_{\mathrm{PFM}} \mathcal{D}^{+}_{\mathrm{mf}}$.

Theorem 2. *$\mathcal{D}^{+}_{\mathrm{mv}} \not\leq_{\mathrm{FM}} \mathcal{D}^{+}_{\mathrm{m}}$*

Proof. Let us assume that there is a polynomial and faithful translation function $\mathrm{Tr} : \mathcal{D}^{+}_{\mathrm{mv}} \rightarrow \mathcal{D}^{+}_{\mathrm{m}}$ that effectively removes varying atoms. Then consider two disjoint logic programs $\varPi_1 = \{a \vee b.\ \}$ and $\varPi_2 = \{\bot \leftarrow b, a.\ \}$ based on $\mathrm{Hb}(\varPi_1) = \mathrm{Hb}(\varPi_2) = \{a, b\}$ with all atoms visible, i.e. $\mathrm{Hb}_{\mathrm{h}}(\varPi_1) = \mathrm{Hb}_{\mathrm{h}}(\varPi_2) = \emptyset$. Then let us define $V_1 = \{a\}$ and $V_2 = \{b\}$ as the sets of varying atoms associated with $\varPi_1$ and $\varPi_2$, respectively. As regards $\varPi_1$ and $\varPi_2$, it is straightforward to verify that

1. the only $\langle V_1, \emptyset\rangle$-minimal model of $\varPi_1$ is $M_1 = \{a\}$;

2. the program Π_2 has two $\langle V_2, \emptyset\rangle$-minimal models $M_2 = \{b\}$ and $M_3 = \emptyset$; and
3. the program $\Pi_1 \sqcup \Pi_2$ has two $\langle V_1 \sqcup V_2, \emptyset\rangle$-minimal models M_1 and M_2.

On the other hand, the translations $\mathrm{Tr}(\Pi_1, V_1)$, $\mathrm{Tr}(\Pi_2, V_2)$, and $\mathrm{Tr}(\Pi_1 \sqcup \Pi_2, V_1 \sqcup V_2)$ are PDLPs whose all atoms are subject to minimization. Since Tr is faithful, we know that $\mathrm{Tr}(\Pi_1, V_1)$ has a $\langle\emptyset, \emptyset\rangle$-minimal model N such that $N \cap \mathrm{Hb}(\Pi_1) = M_1$, and $\mathrm{Tr}(\Pi_1 \sqcup \Pi_2, V_1 \sqcup V_2)$ has two $\langle\emptyset, \emptyset\rangle$-minimal models N_1 and N_2 such that $N_1 \cap \mathrm{Hb}(\Pi_1 \sqcup \Pi_2) = M_1 = \{a\}$ and $N_2 \cap \mathrm{Hb}(\Pi_1 \sqcup \Pi_2) = M_2 = \{b\}$.

Using the modularity of Tr, we obtain $\mathrm{Tr}(\Pi_1 \sqcup \Pi_2, V_1 \sqcup V_2) = \mathrm{Tr}(\Pi_1, V_1) \sqcup \mathrm{Tr}(\Pi_2, V_2)$. Since $N_2 \models \mathrm{Tr}(\Pi_1 \sqcup \Pi_2, V_1 \sqcup V_2)$, we obtain $N_2 \models \mathrm{Tr}(\Pi_1, V_1)$. It follows that $N' \models \mathrm{Tr}(\Pi_1, V_1)$ holds for the restricted model $N' = N_2 \cap \mathrm{Hb}(\mathrm{Tr}(\Pi_1, V_1))$ from which the local atoms of $\mathrm{Tr}(\Pi_2, V_2)$ have been removed. Recall that $\mathrm{Hb}(\Pi_1) \subseteq \mathrm{Hb}(\mathrm{Tr}(\Pi_1, V_1))$ by the faithfulness of Tr. Because $N' \models \mathrm{Tr}(\Pi_1, V_1)$ and N is the unique $\langle\emptyset, \emptyset\rangle$-minimal model of $\mathrm{Tr}(\Pi_1, V_1)$, we obtain $N \subseteq N'$. A contradiction, since $a \in N$ but $a \notin N'$. To conclude, such a translation function Tr does not exist. $\square$

It follows that $\mathcal{D}^+_{\mathrm{m}} <_{\mathrm{PFM}} \mathcal{D}^+_{\mathrm{mv}}$, since $\mathcal{D}^+_{\mathrm{m}}$ is a subclass of $\mathcal{D}^+_{\mathrm{mv}}$. The first condition of Definition 3 implies that the classes $\mathcal{D}^+_{\mathrm{v}}$, $\mathcal{D}^+_{\mathrm{f}}$, and $\mathcal{D}^+_{\mathrm{vf}}$ collapse to classical logic, i.e. the semantics assigned to a PDLP Π is $\mathrm{CM}(\Pi) = \{M \subseteq \mathrm{Hb}(\Pi) \mid M \models \Pi\}$. Moreover, PFM translation functions are easily obtained for each pair of classes. E.g., a translation function from $\mathcal{D}^+_{\mathrm{v}}$ to $\mathcal{D}^+_{\mathrm{f}}$ simply exchanges the roles of varying and fixed atoms. This is semantically irrelevant, as no atoms are subject to minimization. Such a translation function is trivially PFM. Hence, we have $\mathcal{D}^+_{\mathrm{v}} =_{\mathrm{PFM}} \mathcal{D}^+_{\mathrm{f}} =_{\mathrm{PFM}} \mathcal{D}^+_{\mathrm{vf}}$ and there is only one relationship to be further explored.

Theorem 3. $\mathcal{D}^+_{\mathrm{m}} \not\leq_{\mathrm{FM}} \mathcal{D}^+_{\mathrm{v}}$

Proof. Consider $\Pi_1 = \{a \leftarrow a\}$ and $\Pi_2 = \{a\}$ which have unique $\langle\emptyset, \emptyset\rangle$-minimal models $M_1 = \emptyset$ and $M_2 = \{a\}$, respectively. Assuming the existence of a faithful and modular translation function Tr, we obtain that $\mathrm{Tr}(\Pi_1)$ and $\mathrm{Tr}(\Pi_2)$ have unique classical models N_1 and N_2, respectively, such that $N_i \cap \mathrm{Hb}(\Pi_i) = M_i$ for $i \in \{1, 2\}$. Thus $\mathrm{Tr}(\Pi_1 \sqcup \Pi_2) = \mathrm{Tr}(\Pi_1) \sqcup \mathrm{Tr}(\Pi_2)$ is necessarily inconsistent — contradicting the faithfulness of Tr, as M_2 is the unique $\langle\emptyset, \emptyset\rangle$-minimal model of $\Pi_1 \sqcup \Pi_2$. $\square$

The resulting expressive power hierarchy is summarized in Figure 1. There are three equivalence classes under PFM translation functions. The most expressive class corresponds to Lifschitz's parallel circumscription [16] while the class in the middle captures ordinary circumscription proposed by McCarthy [18]. The class at the bottom corresponds to classical logic. In spite of certain differences, these results can be understood as a refinement to an analogous hierarchy derived for non-monotonic logics [10] where the lower end of the hierarchy consists of parallel circumscription and classical logic; the former ranked strictly more expressive than the latter. Let us also note that current disjunctive solvers [15, 13] cover the hierarchy up to the class in the middle.

$$\begin{array}{ccccc}
\mathcal{D}^+_{\mathrm{mv}} & =_{\mathrm{PFM}} & \mathcal{D}^+_{\mathrm{mvf}} & & \\
\rotatebox{90}{$<_{\mathrm{PFM}}$} & & \rotatebox{90}{$<_{\mathrm{PFM}}$} & & \\
\mathcal{D}^+_{\mathrm{m}} & =_{\mathrm{PFM}} & \mathcal{D}^+_{\mathrm{mf}} & & \\
\rotatebox{90}{$<_{\mathrm{PFM}}$} & & \rotatebox{90}{$<_{\mathrm{PFM}}$} & & \\
\mathcal{D}^+_{\mathrm{v}} & =_{\mathrm{PFM}} & \mathcal{D}^+_{\mathrm{f}} & =_{\mathrm{PFM}} & \mathcal{D}^+_{\mathrm{vf}}
\end{array}$$

Fig. 1: Hierarchy Implied by the Expressiveness Analysis

4 Eliminating Varying Atoms

In this section, we present a non-modular translation function $\mathrm{Tr_{BLIND}}$ which enables us to remove varying atoms from a PDLP Π in a faithful way, i.e. $\langle V, F\rangle$-minimal models M of Π and the stable models N of its translation are in a bijective relationship such that $M = N \cap \mathrm{Hb}(\Pi)$ holds for each pair of models. For the sake of simplicity, we assume that fixed atoms have already been removed (recall $\mathrm{Tr_{KK}}$ from Theorem 1).

The translation function $\mathrm{Tr_{BLIND}}$ introduces new atoms, which do not appear in $\mathrm{Hb}(\Pi)$, as follows. For each $a \in \mathrm{Hb}(\Pi)$, the complement $\overline{a}$ of a expresses the falsity of a. Moreover, a renamed copy a^* of each $a \in \mathrm{Hb}(\Pi)$ is needed when formulating a test for $\langle V, \emptyset\rangle$-minimality. Likewise, a vector of new atoms $d_1, \ldots, d_n$ is introduced for the set of atoms $P = \mathrm{Hb}(\Pi) \setminus V = \{a_1, \ldots, a_n\}$ subject to minimization. Yet another new atom, namely u, will be used in the translation. Given a set of atoms $A \subseteq \mathrm{Hb}(\Pi)$, we introduce shorthands $\overline{A}$ and A^* for the sets $\{\overline{a} \mid a \in A\}$ and $\{a^* \mid a \in A\}$, respectively.

Definition 6. *Let Π be a PDLP and $V \subseteq \mathrm{Hb}(\Pi)$ a set of varying atoms. Let us define $P = \mathrm{Hb}(\Pi) \setminus V = \{a_1, \ldots, a_n\}$ and a translation $\mathrm{Tr_{BLIND}}(\Pi)$ containing*

1. *rules $a \leftarrow {\sim}\overline{a}$ and $\overline{a} \leftarrow {\sim}a$ for each $a \in \mathrm{Hb}(\Pi)$;*
2. *a rule $\bot \leftarrow {\sim}A, {\sim}\overline{B}$ for each rule $A \leftarrow B$ in Π;*
3. *a rule $A^* \cup \{u\} \leftarrow B^*$ for each rule $A \leftarrow B$ in Π;*
4. *a rule $d_1 \vee \cdots \vee d_n \vee u$;*
5. *rules $u \leftarrow d_i, {\sim}a_i$ and $u \leftarrow a_i^*, {\sim}a_i$ for each $1 \leq i \leq n$;*
6. *rules $u \leftarrow d_i, a_i^*, {\sim}\overline{a_i}$ and $u \vee d_i \vee a_i^* \leftarrow {\sim}\overline{a_i}$ for each $1 \leq i \leq n$;*
7. *a rule $a^* \leftarrow u$ for each $a \in \mathrm{Hb}(\Pi)$;*
8. *a rule $d_i \leftarrow u$ for each $1 \leq i \leq n$; and*
9. *a rule $\bot \leftarrow {\sim}u$.*

The rules included in $\mathrm{Tr_{BLIND}}(\Pi)$ serve the following purposes. (1.) An arbitrary interpretation $M \subseteq \mathrm{Hb}(\Pi)$ is chosen for the PDLP Π. (2.) It is ensured that $M \models \Pi$ holds in the classical sense. (3.) A renamed copy of Π is created to check the $\langle V, \emptyset\rangle$-minimality of M. In analogy to [13], this can be achieved by checking whether

$$\mathrm{Tr_{UNSAT}}(\Pi, P, M) = \Pi \cup \{\bot \leftarrow P \cap M\} \cup \{\bot \leftarrow a \mid a \in P \setminus M\} \tag{3}$$

is unsatisfiable for M and the set of atoms $P = \{a_1, \ldots, a_n\}$ subject to minimization. This is why the intuitive reading of u is *unsatisfiable* which captures the desired state of affairs, implying the $\langle V, \emptyset\rangle$-minimality of M. (4.) The disjunction $d_1 \vee \cdots \vee d_n$ captures the rule $\bot \leftarrow P \cap M$ from (3). This rule depends dynamically on M and it effectively states the *falsity* of at least one atom a_i that is both subject to minimization ($a_i \in P$) and true in M ($a_i \in M$). (5.) The rules cover the case that a_i is false in M, i.e. $a_i \in P \setminus M$. Conforming to (3), both d_i and a_i^* are implicitly assigned to false, as they imply u. Otherwise, a_i is true in M which activates the rules in (6.) enforcing d_i equivalent to the negation of a_i^*. The net effect of the rules included in (4.) – (6.) is that any *potential* counter-model $N \models \Pi$ for the $\langle V, \emptyset\rangle$-minimality of M, expressed in $\mathrm{Hb}(\Pi)^*$ rather than $\mathrm{Hb}(\Pi)$, must satisfy $N \cap P \subset M \cap P$ ($\iff N \setminus V \subset M \setminus V$).

The rules given in items (7.) – (9.) are directly related to the unsatisfiability check which effectively proves that counter-models like N above do not exist. To implement the test for unsatisfiability, we adopt the technique used earlier by Eiter and Gottlob [5].

Example 2. Consider a program $\Pi = \{f \vee ab.\}$ which is a simplified version of Lifschitz's ostrich example [16]. This program has a unique $\langle\{f\},\emptyset\rangle$-minimal model $M = \{f\}$. The translation $\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)$ includes the following rules: (1.) $f \leftarrow \sim\overline{f}$. $\overline{f} \leftarrow \sim f$. $ab \leftarrow \sim\overline{ab}$. $\overline{ab} \leftarrow \sim ab$. (2.) $\bot \leftarrow \sim f, \sim ab$. (3.) $f^* \vee ab^* \vee u$. (4.) $d \vee u$. (5.) $u \leftarrow d, \sim ab$. $u \leftarrow ab^*, \sim ab$. (6.) $u \leftarrow d, ab^*, \sim\overline{ab}$. $u \vee d \vee ab^* \leftarrow \sim\overline{ab}$. (7.) $ab^* \leftarrow u$. $f^* \leftarrow u$. (8.) $d \leftarrow u$. (9.) $\bot \leftarrow \sim u$. There is only one stable model for $\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)$, i.e. $N = \{f, \overline{ab}, f^*, ab^*, d, u\}$ for which $M = N \cap \{f, ab\}$ holds. $\square$

Our next objective is to establish that the translation function $\mathrm{Tr}_{\mathrm{BLIND}}$ given in Definition 6 is faithful, i.e. the $\langle V,\emptyset\rangle$-minimal models of a PDLP Π are in a bijective relationship with the stable models of $\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)$. In analogy to [19], we implement the test for $\langle V,\emptyset\rangle$-minimality through propositional unsatisfiability.

Lemma 1. *Given a PDLP Π and $V \subseteq \mathrm{Hb}(\Pi)$, a model $M \subseteq \mathrm{Hb}(\Pi)$ of Π is $\langle V,\emptyset\rangle$-minimal if and only if $\mathrm{Tr}_{\mathrm{UNSAT}}(\Pi, \mathrm{Hb}(\Pi) \setminus V, M)$, as defined in (3), is unsatisfiable.*

We split the translation $\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)$ in two parts using the *Splitting Set Theorem* [17] which we formulate for stable models rather than *answer sets* used in [17]. A *splitting set* for a DLP Π is any set $U \subseteq \mathrm{Hb}(\Pi)$ such that for every rule $A \leftarrow B, \sim C \in \Pi$, if $A \cap U \neq \emptyset$ then $A \cup B \cup C \subseteq U$. The set of rules $A \leftarrow B, \sim C \in \Pi$ such that $A \cup B \cup C \subseteq U$ is the *bottom* of Π relative to U, denoted by $\mathrm{b}_U(\Pi)$. The set $\mathrm{t}_U(\Pi) = \Pi \setminus \mathrm{b}_U(\Pi)$ is the *top* of Π relative to U which can be partially evaluated with respect to an interpretation $X \subseteq U$. The result is a DLP $\mathrm{e}_U(\mathrm{t}_U(\Pi), X)$ defined as $\{A \leftarrow (B \setminus U), \sim(C \setminus U) \mid A \leftarrow B, \sim C \in \mathrm{t}_U(\Pi),\ B \cap U \subseteq X \text{ and } (C \cap U) \cap X = \emptyset\}$. Given a splitting set U for a program Π, a *solution* to Π with respect to U is a pair $\langle X, Y\rangle$ such that (i) $X \subseteq U$ is a stable model of $\mathrm{b}_U(\Pi)$ and (ii) $Y \subseteq \mathrm{Hb}(\Pi) \setminus U$ is a stable model of $\mathrm{e}_U(\mathrm{t}_U(\Pi), X)$. Solutions and stable models relate as follows.

Theorem 4 (Splitting Set Theorem [17]). *Let U be a splitting set for a DLP Π and $M \subseteq \mathrm{Hb}(\Pi)$ an interpretation. Then $M \in \mathrm{SM}(\Pi)$ if and only if the pair $\langle X, Y\rangle$ with $X = M \cap U$ and $Y = M \setminus U$ is a solution to Π with respect to U.*

We use the set of atoms $U = \mathrm{Hb}(\Pi) \cup \{\overline{a} \mid a \in \mathrm{Hb}(\Pi)\}$ to split $\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)$: the bottom $\mathrm{b}_U(\mathrm{Tr}_{\mathrm{BLIND}}(\Pi))$ consists of items 1 and 2 in Definition 6, whereas the partially evaluated top $\mathrm{e}_U(\mathrm{t}_U(\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)), X)$ consists of items 3, 4 and 7–9 in Definition 6 as such and the following rules corresponding to rules in items 5 and 6:

5.' $u \leftarrow d_i$ and $u \leftarrow a_i^*$ where $1 \leq i \leq n$ and $a_i \in P \setminus X$; and
6.' $u \leftarrow d_i, a_i^*$ and $u \vee d_i \vee a_i^*$ where $1 \leq i \leq n$ and $a_i \in P \cap X$.

Thus $\mathrm{Hb}(\mathrm{e}_U(\mathrm{t}_U(\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)), X)) = \{a^* \mid a \in \mathrm{Hb}(\Pi)\} \cup \{d_i \mid 1 \leq i \leq n\} \cup \{u\}$. We use the notation $\mathrm{E}_U(\Pi, X) = \mathrm{e}_U(\mathrm{t}_U(\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)), X)$ for the sake of brevity.

It is shown next that there is one-to-one correspondence between the models in $\mathrm{SM}(\mathrm{b}_U(\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)))$ and $\mathrm{CM}(\Pi)$. As a consequence, the stable models of the bottom $\mathrm{b}_U(\mathrm{Tr}_{\mathrm{BLIND}}(\Pi))$ are classical models of Π extended to $\mathrm{Hb}(\mathrm{b}_U(\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)))$.

Proposition 1. *Let Π be a PDLP.*

The function $\mathrm{Ext}_{\mathrm{B}} : \mathrm{CM}(\Pi) \rightarrow \mathbf{2}^{\mathrm{Hb}(\mathrm{b}_U(\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)))}$ defined by $\mathrm{Ext}_{\mathrm{B}}(M) = M \cup \{\overline{a} \mid a \in \mathrm{Hb}(\Pi) \setminus M\}$ is a bijection from $\mathrm{CM}(\Pi)$ to $\mathrm{SM}(\mathrm{b}_U(\mathrm{Tr}_{\mathrm{BLIND}}(\Pi)))$.

Proof. It is shown below that (i) the image of $\mathrm{CM}(\Pi)$ under $\mathrm{Ext_B}$ is a subset of $\mathrm{SM}(\mathrm{b}_U(\mathrm{Tr_{BLIND}}(\Pi)))$, (ii) $\mathrm{Ext_B}$ is an injection, and (iii) $\mathrm{Ext_B}$ is a surjection.

(i) Assume that $M \in \mathrm{CM}(\Pi)$, i.e. $M \models \Pi$. It is clear that $X \models \mathrm{b}_U(\mathrm{Tr_{BLIND}}(\Pi))$ holds for $X = \mathrm{Ext_B}(M)$ and it suffices to prove $X \in \mathrm{MM}(\mathrm{b}_U(\mathrm{Tr_{BLIND}}(\Pi))^X)$. Since $M \models \Pi$, the reduct $\mathrm{b}_U(\mathrm{Tr_{BLIND}}(\Pi))^X$ contains only the rules $a \leftarrow$ for $\overline{a} \notin X$ and $\overline{a} \leftarrow$ for $a \notin X$. Thus $X \in \mathrm{MM}(\mathrm{b}_U(\mathrm{Tr_{BLIND}}(\Pi))^X)$.

(ii) If $M_1 \neq M_2$, then $\mathrm{Ext_B}(M_1) \neq \mathrm{Ext_B}(M_2)$ follows by the definition of $\mathrm{Ext_B}$.

(iii) Consider any $X \in \mathrm{SM}(\mathrm{b}_U(\mathrm{Tr_{BLIND}}(\Pi)))$. We need to show that there is $M \in \mathrm{CM}(\Pi)$ such that $\mathrm{Ext_B}(M) = X$. Let us establish first that $M \models \Pi$ holds for $M = X \cap \mathrm{Hb}(\Pi)$. Since $X \in \mathrm{SM}(\mathrm{b}_U(\mathrm{Tr_{BLIND}}(\Pi)))$ and $\mathrm{b}_U(\mathrm{Tr_{BLIND}}(\Pi))$ contains the rules $a \leftarrow \sim\overline{a}$ and $\overline{a} \leftarrow \sim a$ for each $a \in \mathrm{Hb}(\Pi)$, it holds for every $a \in \mathrm{Hb}(\Pi)$ that $\overline{a} \notin X \iff a \in X$. Moreover, since $X \models \mathrm{b}_U(\mathrm{Tr_{BLIND}}(\Pi))$, we obtain $X \not\models \sim A \cup \sim\overline{B}$ for all rules $A \leftarrow B \in \Pi$. Thus for each rule $A \leftarrow B$ in Π, there is $a \in A$ such that $a \in X$, or $b \in B$ such that $\overline{b} \in X$ ($\iff b \notin X$). In either case, $M \models A \leftarrow B$ and therefore $M \models \Pi$, i.e. $M \in \mathrm{CM}(\Pi)$. It remains to establish that $\mathrm{Ext_B}(M) = X$. Since $M = X \cap \mathrm{Hb}(\Pi)$, we have $\mathrm{Ext_B}(M) = \mathrm{Ext_B}(X \cap \mathrm{Hb}(\Pi)) = (X \cap \mathrm{Hb}(\Pi)) \cup \{\overline{a} \mid a \in \mathrm{Hb}(\Pi) \setminus X\}$. Then $\mathrm{Ext_B}(M) = X$ follows by the fact that $\overline{a} \notin X \iff a \in X$ holds for any $a \in \mathrm{Hb}(\Pi)$. □

Finally, we show the connection between $\mathrm{SM}(\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M))) \neq \emptyset$ and the unsatisfiability of $\mathrm{Tr_{UNSAT}}(\Pi, P, M)$. A similar unsatisfiability check is used in [5].

Proposition 2. *Let Π, V, and $P = \{a_1, \ldots, a_n\}$ be defined as in Definition 6 and* $\mathrm{Ext_B}$ *as in Proposition 1. Moreover, let $M \subseteq \mathrm{Hb}(\Pi)$ be a classical model of Π. Then (i) if $N \in \mathrm{SM}(\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M)))$, then $N = \mathrm{Hb}(\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M)))$, and (ii)* $\mathrm{Tr_{UNSAT}}(\Pi, P, M)$ *is unsatisfiable if and only if* $\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M))$ *has a stable model.*

Proof. (i) Assume that $N \in \mathrm{SM}(\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M)))$. Since $N \models \mathrm{E}_U(\Pi, \mathrm{Ext_B}(M))$ and the rule $\bot \leftarrow \sim u$ belongs to $\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M))$, we must have $u \in N$. Furthermore, since the rules $a^* \leftarrow u$ (for all $a \in \mathrm{Hb}(\Pi)$) and $d_i \leftarrow u$ (for all $1 \leq i \leq n$) belong to $\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M))$, it follows that $N = \mathrm{Hb}(\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M)))$.

(ii) "⇒" Assume that $\mathrm{Tr_{UNSAT}}(\Pi, P, M)$ is unsatisfiable. It is easy to see that $N \models \mathrm{E}_U(\Pi, \mathrm{Ext_B}(M))$ holds for $N = \mathrm{Hb}(\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M)))$. Let us then show that $N \in \mathrm{MM}(\mathrm{E}_U(\Pi, \mathrm{Ext_B}(M))^N)$ by assuming the opposite, i.e. there is $N' \subset N$ such that $N' \models \mathrm{E}_U(\Pi, \mathrm{Ext_B}(M))^N$. Let us then assume $u \notin N'$ and define an interpretation $M' = \{a \in \mathrm{Hb}(\Pi) \mid a^* \in N'\}$. The following observations can be made.

- We have $N' \models A^* \leftarrow B^*$ for each rule $A \leftarrow B \in \Pi$. Thus $M' \models \Pi$.
- Since $N' \models u \leftarrow d_i, a_i^*$ and $N' \models u \vee d_i \vee a_i^*$ for all $a_i \in P \cap \mathrm{Ext_B}(M) = P \cap M$, it holds $d_i \in N' \iff a_i^* \notin N'$ for all $a_i \in P \cap M$. Also $N' \models d_1 \vee \cdots \vee d_n$ and $N' \models u \leftarrow d_i$ for all $a_i \in P \setminus \mathrm{Ext_B}(M) = P \setminus M$. Thus there is $a_i \in P \cap M$ such that $d_i \in N'$ and $a_i^* \notin N'$, too. This implies $a_i \notin M'$ and $M' \models \bot \leftarrow P \cap M$.
- Since $N' \models u \leftarrow a_i^*$ for all $a_i \in P \setminus M$, we have $a_i^* \notin N'$ for all $a_i \in P \setminus M$. This implies $M' \models \{\bot \leftarrow a \mid a \in P \setminus M\}$.

Thus $M' \models \mathrm{Tr_{UNSAT}}(\Pi, P, M)$ which is a contradiction so that $u \in N'$ must be the case. Since $u \in N'$ and the rules $a^* \leftarrow u$ (for all $a \in \mathrm{Hb}(\Pi)$) and $d_i \leftarrow u$ (for all $1 \leq i \leq n$) belong to $\mathrm{E}_U(\varPi, \mathrm{Ext}_{\mathrm{B}}(M))^N$, we must have that $a^* \in N'$ for all $a \in \mathrm{Hb}(\varPi)$ and $d_i \in N'$ for $1 \leq i \leq n$. Thus $N' = N$ contradicting our previous assumption. Therefore $N \in \mathrm{SM}(\mathrm{E}_U(\varPi, \mathrm{Ext}_{\mathrm{B}}(M)))$ is necessarily the case.

(ii) "$\Leftarrow$" Consider any $N \in \mathrm{SM}(\mathrm{E}_U(\varPi, \mathrm{Ext}_{\mathrm{B}}(M)))$. It follows by (i) that $N = \mathrm{Hb}(\mathrm{E}_U(\varPi, \mathrm{Ext}_{\mathrm{B}}(M)))$. Let us then assume that $\mathrm{Tr}_{\mathrm{UNSAT}}(\varPi, P, M)$ is satisfiable, i.e. there is $M' \subseteq \mathrm{Hb}(\varPi)$ such that $M' \models \varPi$, $M' \not\models P \cap M$ and $a \notin M'$ for all $a \in P \setminus M$. It is established in the sequel that $N' \models \mathrm{E}_U(\varPi, \mathrm{Ext}_{\mathrm{B}}(M))^N$ holds for the interpretation N' defined as $(M')^* \cup \{d_i \mid a_i \in M \cap P \text{ and } a_i \notin M'\}$.

- Since $M' \models \varPi$, we have $N' \models A^* \cup \{u\} \leftarrow B^*$ for each rule $A \leftarrow B \in \varPi$.
- Since $M' \not\models P \cap M$, there is $d_i \in N'$ and thus $N' \models d_1 \vee \cdots \vee d_n \vee u$.
- The definition of N' implies $a_i^* \notin N'$ and $d_i \notin N'$ for all $a_i \in P \setminus M$, as $a_i \notin M'$ for all $a_i \in P \setminus M$. Thus $N' \models u \leftarrow d_i$ and $N' \models u \leftarrow a_i^*$ when $a_i \in P \setminus M$.
- Given $a_i \in P \cap M$, we have $d_i \in N' \iff a_i \notin M'$, i.e. $a_i^* \notin N'$ by the definition of N'. Thus $N' \models u \leftarrow d_i, a_i^*$ and $N' \models u \vee d_i \vee a_i^*$ hold whenever $a_i \in P \cap M$.
- Since $u \notin N'$, we have $N' \models a^* \leftarrow u$ for each $a \in \mathrm{Hb}(\varPi)$.
- Since $u \notin N'$, it follows that $N' \models d_i \leftarrow u$ for each $1 \leq i \leq n$.

Now $N' \subset N$ and $N' \models \mathrm{E}_U(\varPi, \mathrm{Ext}_{\mathrm{B}}(M))^N$, contradicting the assumption $N \in \mathrm{SM}(\mathrm{E}_U(\varPi, \mathrm{Ext}_{\mathrm{B}}(M)))$. Thus $\mathrm{Tr}_{\mathrm{UNSAT}}(\varPi, P, M)$ must be unsatisfiable. $\square$

We let $\mathcal{D}$ denote the class of DLPs under the stable model semantics [8, 20]. The translation function $\mathrm{Tr}_{\mathrm{BLIND}} : \mathcal{D}^+_{\mathrm{mv}} \to \mathcal{D}$ is clearly linear. Assuming that the visible Herbrand base $\mathrm{Hb}_{\mathrm{v}}(\mathrm{Tr}_{\mathrm{BLIND}}(\varPi)) = \mathrm{Hb}_{\mathrm{v}}(\varPi)$ by definition, the faithfulness of translation $\mathrm{Tr}_{\mathrm{BLIND}}(\varPi)$ follows by Theorem 4 from Lemma 1, and Propositions 1 and 2.

Theorem 5. $\mathcal{D}^+_{\mathrm{mv}} \leq_{\mathrm{PF}} \mathcal{D}$.

5 Discussion

The main result of this paper is a linear translation from parallel circumscription into disjunctive logic programs such that a bijective correspondence between the $\langle V, F\rangle$-minimal models of a PDLP $\varPi$ and the stable models of the respective translation $\mathrm{Tr}_{\mathrm{BLIND}}(\mathrm{Tr}_{\mathrm{KK}}(\varPi))$ is obtained. As suggested by the analysis performed in Section 3, the translation function $\mathrm{Tr}_{\mathrm{BLIND}}$ is non-modular — reflecting the global nature of varying atoms. In contrast to earlier attempts [7, 22, 14], our translation does not depend on syntactic restrictions and it has a linear time/space complexity. Cadoli et al. [2] achieve the same complexity, but their transformation has ordinary circumscription as the target formalism, and hence a bijective relationship of models cannot be obtained. However, the translation function $\mathrm{Tr}_{\mathrm{BLIND}}$ presented in this paper exploits default negation in order to establish faithfulness in the strict sense implied by Definitions 5 and 2.

Our results enable the systematic use of varying atoms in order to develop more compact formulations of problems as disjunctive logic programs. A good example in this respect is the consistency-based diagnosis of digital circuits [21]. Reiter-style *minimal diagnoses* are hard to formalize when all atoms are subject to minimization. Following the ideas from [1], a digital circuit can be modeled as follows. For instance, an *inverter* I is described by a propositional theory $(o_I \leftrightarrow \neg i_I) \vee ab_I$, where the atoms i_I and o_I model the input and the output of I, respectively, and ab_I expresses the fact that I is operating against its specification. This theory can be equivalently formulated as a PDLP $\Pi_I = \{ab_I \leftarrow i_I, o_I.\ i_I \vee o_I \vee ab_I.\ \}$ and minimal diagnoses correspond to $\langle\{i_I, o_I\}, \emptyset\rangle$-minimal models of Π_I augmented by observations. This line of thinking carries over to larger circuits which have also other gates than inverters as their components. Assuming the availability of varying atoms, the description of the circuit can be formed in a very modular fashion, component-by-component. Then the description can be translated into a valid input for disjunctive solvers like dlv [15] and GnT [13] using the translation function $\mathrm{Tr}_{\mathrm{BLIND}}$. On the other hand, we run into severe problems if all atoms are set subject to minimization. For example, the program Π_I which models an inverter I has three $\langle\emptyset, \emptyset\rangle$-minimal models $M_1 = \{i_I\}$, $M_2 = \{o_I\}$, and $M_3 = \{ab_I\}$. The first two minimal models capture natural explanations given no observations on I, but the third minimal model does not correspond to a Reiter-style minimal diagnosis, as I is faulty according to it. Similar spurious minimal models are also obtained for more complex circuits encoded in this way if all atoms are subject to minimization.

Our first experiments with large combinational circuits showed that our approach is not yet competitive with a special purpose engine [1] which exploits 1-fault assumption. The diagnosis front-end of the dlv system also covers Reiter-style minimal diagnoses [4], but models like the one described above are ruled out by syntactic restrictions. Moreover, contrary to $\mathrm{Tr}_{\mathrm{BLIND}}$, the translation used in the front-end yields only a *many-to-one* correspondence between stable models and diagnoses.

As a further application of varying atoms, a specific reduction from quantified Boolean formulas (QBFs) to DLPs [13] can be improved to produce all satisfying assignments for a $2, \exists$-QBF $\exists X \forall Y \phi$ given as input. Due to blind minimization, the current reduction does not yield a one-to-one correspondence between the satisfying assignments of $\exists X \forall Y \phi$ and the stable models of the resulting DLP. However, the validity of $\exists X \forall Y \phi$ is properly captured by the reduction.

To conclude, it might be a good idea to implement varying atoms directly in disjunctive solvers. This is a challenge, as existing algorithms [13, 15] rely much on the fact that all atoms are subject to minimization. A further question is how varying and fixed atoms should be incorporated into stable models. Is it enough to consider $\langle V, F\rangle$-minimal models of the Gelfond-Lifschitz reduct [8] or should V and F be dynamically determined? Finally, we remind the reader about a reduction from *prioritized circumscription* to parallel circumscription [16] which implies that even prioritized circumscription can be captured with disjunctive programs using the technique from Section 4.

References

1. P. Baumgartner, P. Fröhlich, U. Furbach, and W. Nejdl. Semantically guided theorem proving for diagnosis applications. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pages 460–465, Nagoya, 1997. Morgan Kaufmann.
2. M. Cadoli, T. Eiter, and G. Gottlob. An efficient method for eliminating varying predicates from a circumscription. *Artificial Intelligence*, 54(2):397–410, 1992.
3. J. de Kleer and K. Konolige. Eliminating the fixed predicates from a circumscription. *Artificial Intelligence*, 39(3):391–398, July 1989.

4. T. Eiter, W. Faber, N. Leone, and G. Pfeifer. The diagnosis frontend of the DLV system. *AI Communications*, 12(1-2):99–111, 1999.
5. T. Eiter and G. Gottlob. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics and Artificial Intelligence*, 15:289–323, 1995.
6. T. Eiter, G. Gottlob, and H. Veith. Modular logic programming and generalized quantifiers. In J. Dix, U. Furbach, and A. Nerode, editors, *Logic Programming and Nonmonotonic Reasoning*, pages 289–308, Dagstuhl Castle, Germany, July 1997. Springer-Verlag. LNAI 1265.
7. M. Gelfond and V. Lifschitz. Compiling circumscriptive theories into logic programs. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pages 455–449, St. Paul, MN, August 1988. AAAI Press.
8. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
9. K. Inoue and C. Sakama. Negation as failure in the head. *Journal of Logic Programming*, 35(1):39–78, 1998.
10. T. Janhunen. On the intertranslatability of non-monotonic logics. *Annals of Mathematics and Artificial Intelligence*, 27(1-4):79–128, 1999.
11. T. Janhunen. On the effect of default negation on the expressiveness of disjunctive rules. In T. Eiter, W. Faber, and M. Truszczyński, editors, *Logic Programming and Nonmonotonic Reasoning, Proceedings of the 6th International Conference*, pages 93–106, Vienna, Austria, September 2001. Springer-Verlag. LNAI 2173.
12. T. Janhunen. Translatability and intranslatability results for certain classes of logic programs. Series A: Research report 82, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, November 2003.
13. T. Janhunen, I. Niemelä, D. Seipel, P. Simons, and J.-H. You. Unfolding partiality and disjunctions in stable model semantics. *ACM Transactions on Computational Logic*, 2004. Accepted for publication, see `http://www.acm.org/tocl/accepted.html`.
14. J. Lee and F. Lin. Loop formulas for circumscription. In *Proceedings of the 19th National Conference on Artificial Intelligence*, pages 281–286, San Jose, California, July 2004. AAAI.
15. N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, and F. Scarcello. The DLV system for knowledge representation and reasoning. CoRR: cs.AI/0211004 v2, August 2003.
16. V. Lifschitz. Computing circumscription. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 121–127, Los Angeles, California, USA, August 1985. Morgan Kaufmann.
17. V. Lifschitz and H. Turner. Splitting a logic program. In *Proceedings of the 11th International Conference on Logic Programming*, pages 23–37. MIT Press, 1994.
18. J. McCarthy. Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.
19. Ilkka Niemelä. A tableau calculus for minimal model reasoning. In *Analytic Tableaux and Related Methods: Fifth Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, pages 278–294, 1996. LNCS 1071.
20. T.C. Przymusinski. Stable semantics for disjunctive programs. *New Generation Computing*, 9:401–424, 1991.
21. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.
22. C. Sakama and K. Inoue. Embedding circumscriptive theories in general disjunctive programs. In *Proceedings of the 3rd International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 344–357. Springer-Verlag, 1995.
23. P. Simons. Extending the stable model semantics with more expressive rules. In *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 305–316, El Paso, Texas, USA, 1999. Springer-Verlag.