On Eliminating Disjunctions in Stable Logic Programming*

Thomas Eiter, Michael Fink, Hans Tompits, and Stefan Woltran
Institut für Informationssysteme 184/3,
Technische Universität Wien,
Favoritenstraße 9-11, A-1040 Vienna, Austria
e-mail: {eiter, michael, tompits, stefan}@kr.tuwien.ac.at

Abstract

Disjunction is generally considered to add expressive power to logic programs under the stable model semantics, which have become a popular programming paradigm for knowledge representation and reasoning. However, disjunction is often not really needed, in that an equivalent program without disjunction can be given. In this paper, we consider the question, given a disjunctive logic program, P, under which conditions an equivalent normal (i.e., disjunction-free) logic program P' exists. In fact, we study this problem under different notions of equivalence, viz. for ordinary equivalence (considering the collections of all stable models of the programs) as well as for the more restrictive notions of strong and uniform equivalence. We resolve the issue for propositional programs on which we focus here, and present a simple, appealing semantic criterion from which all disjunctions can be eliminated under strong equivalence. Testing this criterion is coNP-complete, but the class of programs satisfying it has the same complexity as disjunctive logic programs in general. We also show that under ordinary and uniform equivalence, disjunctions can always be eliminated. In all cases, we give constructive methods to achieve this. However, we also provide evidence that disjunctive logic programs are a more succinct knowledge representation formalism than normal logic programs under all these notions of equivalence.

Introduction

Disjunctive logic programming extends normal logic programming by permitting disjunctions to appear in rule heads, and is generally regarded to add expressive power to logic programs under the stable model semantics. This view is supported by results on the expressiveness of disjunctive logic programs (DLPs) over finite structures, which show that properties at the second level of the polynomial hierarchy can be expressed by inference from function-free (datalog) DLPs (Eiter, Gottlob, & Mannila 1997), while normal logic programs (NLPs) can express only properties at the first level of the polynomial hierarchy (Schlipf 1995). However, disjunction is often not really needed, in that an equivalent normal logic program (i.e., without disjunction) can be given. For example, Eiter & Gottlob (1997) showed that in the presence of functions symbols, DLPs have over Herbrand models the same expressive power as NLPs, which is Π^1_1.

Given the availability of efficient solvers for the stable model semantics, such as DLV (Eiter *et al.* 2000), Smodels (Simons, Niemelä, & Soininen 2002), ASSAT (Lin & Zhao 2002), or GnT (Janhunen *et al.* 2000), which utilize efficient algorithms and methods, the approach to encode solutions of a problem in terms of the stable models resp. answer sets of a logic program (known as *stable logic programming* or *answer set programming*), has become a popular paradigm for solving KR problems in different areas, like, e.g., planning, inheritance reasoning, and diagnosis, to mention just a few. This raised interest in the expressiveness of logic programs in terms of the whole collection of their stable models (or answer sets) rather than their intersection or union as in cautious and brave reasoning, respectively (Marek & Remmel 2003). Related to this is preliminary work on the expressiveness of other well-known KR formalisms, such as default logic and circumscription, in terms of their extensions and models (Cadoli *et al.* 2000a; Gogic *et al.* 1995; Marek, Treur, & Truszczyński 1997).

Recently, different notions of equivalence between logic programs have been studied. Besides the usual equivalence between programs, i.e., checking whether two programs have the same stable models, the more refined notions of *strong equivalence* (Lifschitz, Pearce, & Valverde 2001; Turner 2001; 2003; Pearce, Tompits, & Woltran 2001; Lin 2002; de Jongh & Hendriks 2003) and *uniform equivalence* (Eiter & Fink 2003; Pearce & Valverde 2003; Eiter *et al.* 2004) have been investigated. Formally, two DLPs P_1 and P_2 are strongly equivalent (resp., uniformly equivalent), if, for any set R of rules (resp., atoms), the programs $P_1 \cup R$ and $P_2 \cup R$ are equivalent in the usual sense.

Strong and uniform equivalence can be utilized for program optimization (Turner 2003; Osorio, Navarro, & Arrazola 2001; Eiter *et al.* 2004), taking a possible incompleteness of a program into account, where not all rules are known at optimization time, and for varying input data given by atomic facts, respectively. This is particularly help-

*This work was partially supported by the Austrian Science Fund (FWF) under grants Z29-N04 and P15068-INF, as well as by the European Commission under projects FET-2001-37004 WASP, IST-2001-33570 INFOMIX, and the IST-2001-33123 CologNeT Network of Excellence.
Copyright © 2004, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

ful for optimizing components of a more complex logic program. Note that, as recently discussed by Pearce & Valverde (2003), uniform and strong equivalence are essentially the only concepts of equivalence obtained by varying the syntactic form of the program extensions.

A natural issue in this context is the expressiveness of disjunctions in rule heads, i.e., whether they really add expressive power. This is indeed the case, as can be seen by the simple example of the program $P = \{a \vee b \leftarrow\}$: This program is not strongly equivalent to any normal logic program P' (cf. (Turner 2003)). However, as easily seen, P is equivalent to the NLP $P' = \{a \leftarrow not\, b,\ b \leftarrow not\, a\}$, since for both the stable models are $X_1 = \{a\}$ and $X_2 = \{b\}$, and furthermore P is also uniformly equivalent to P'.

This raises the question of a criterion which determines when disjunctions can be eliminated, and a method for deciding, given a DLP P, whether an equivalent NLP P' exists. We study this issue for propositional programs, on which we focus here, and make the following contributions:

(i) We present a simple, appealing semantic characterization of the programs from which all disjunctions can be eliminated under strong equivalence. In particular, the characterization is based on the condition that, for each classical model Y of a program P, the Gelfond-Lifschitz reduct P^Y of P is semantically Horn if models of P^Y not contained in Y are disregarded, i.e., $X, X' \subseteq Y$ being a model of P^Y implies that $X \cap X'$ is also a model of P^Y.

(ii) We further show that under ordinary and uniform equivalence, this elimination is always possible. In all three cases, we obtain a constructive method to rewrite a DLP to an equivalent normal logic program, by stepwise eliminating disjunctions.

(iii) We show that testing whether for a given propositional DLP a strongly equivalent normal program exists is coNP-complete, and, moreover, that the class of programs possessing a strongly equivalent normal program has the same complexity as general disjunctive logic programs.

(iv) Finally, we show that any equivalence-preserving rewriting of a DLP to an NLP must lead in general to an exponential blow-up, providing the polynomial hierarchy does not collapse. Thus, replacing a DLP by an equivalent NLP, which is, in some sense, "easier" to evaluate (viz., with NP or coNP complexity vs. Σ_2^P or Π_2^P) comes at a price. However, there are classes of programs for which rewriting is efficiently possible.

Our results extend and complement recent results on simplifying logic programs under different notions of equivalence (Osorio, Navarro, & Arrazola 2001; Turner 2003; Eiter *et al.* 2004). They might be used for deciding whether a given disjunctive problem representation for a system such as DLV (Eiter *et al.* 2000) or GnT (Janhunen *et al.* 2000) can, in principle, be replaced by an equivalent non-disjunctive representation, and in particular for (automated) rewriting. Furthermore, they contribute to the comparative linguistics of KR formalisms in the sense of (Cadoli *et al.* 2000b; 2000a; Gogic *et al.* 1995), showing that DLPs are more succinct than NLPs under different notions of equivalence.

Preliminaries

We deal with propositional disjunctive logic programs, containing rules r of form

$$a_1 \vee \cdots \vee a_l \leftarrow a_{l+1}, \ldots, a_m, not\, a_{m+1}, \ldots, not\, a_n, \quad (1)$$

$n \geq m \geq l \geq 0$, where all a_i are atoms from a finite set of propositional atoms, $\mathcal{A}$, such that $a_1, \ldots, a_l$ are pairwise distinct, and not denotes default negation. The *head* of r is the set $H(r) = \{a_1, \ldots, a_l\}$, and the *body* of r is $B(r) = \{a_{l+1}, \ldots, a_m, not\, a_{m+1}, \ldots, not\, a_n\}$. We also define $B^+(r) = \{a_{l+1}, \ldots, a_m\}$ and $B^-(r) = \{a_{m+1}, \ldots, a_n\}$. Moreover, for a set of atoms $A = \{a_1, \ldots, a_n\}$, $not\, A$ denotes the set $\{not\, a_1, \ldots, not\, a_n\}$.

We call rule r *normal*, if $l \leq 1$; (*proper*) *disjunctive*, if $l > 1$; *positive*, if $n = m$; and *Horn*, if it is normal and positive. If $H(r) = \emptyset$ and $B(r) \neq \emptyset$, then r is a *constraint*; if $B(r) = \emptyset$, r is a *fact*, written as $a_1 \vee \cdots \vee a_l$ if $l > 0$, and as $\bot$ otherwise.

With some abuse of notation, we identify rules of form (1) also by $H(r) \leftarrow B^+(r), not\, B^-(r)$.

A *disjunctive logic program* (DLP), P, is a finite set of rules. P is called a *normal logic program* (NLP) (resp., *positive program*, *Horn program*), if every rule in P is normal (resp., positive, Horn). We use $\mathcal{DLP}$ and $\mathcal{NLP}$ to denote the classes of DLPs and NLPs, respectively.

We recall the stable model semantics for DLPs (Gelfond & Lifschitz 1991; Przymusinski 1991). Let I be an interpretation, i.e., a subset of $\mathcal{A}$. Then, an atom a is *true under* I, symbolically $I \models a$, iff $a \in I$, and *false under* I otherwise. For a rule r, $I \models H(r)$ iff some $a \in H(r)$ is true under I, and $I \models B(r)$ iff (i) each $a \in B^+(r)$ is true under I, and (ii) each $a \in B^-(r)$ is false under I. I *satisfies* r, denoted $I \models r$, iff $I \models H(r)$ whenever $I \models B(r)$. Furthermore, I is a *model* of a program P, denoted $I \models P$, iff $I \models r$, for all $r \in P$. As usual, $P \models r$ iff $I \models r$, for each model I of P.

The *Gelfond-Lifschitz reduct* of a program P *relative to* a set of atoms I is the positive program

$$P^I = \{H(r) \leftarrow B^+(r) \mid r \in P,\ B^-(r) \cap I = \emptyset\}.$$

For a single rule r, we write r^I instead of $\{r\}^I$. An interpretation I is a *stable model* of a program P iff I is a minimal model (under set inclusion) of P^I. The set of all stable models of P is denoted by $\mathcal{SM}(P)$. Note that an empty program has any interpretation as its model.

The following property will be required later on.

Proposition 1 *Let P be a DLP and $X, Y \subseteq Z$ interpretations. Then, $X \models P^Y$ implies $X \models P^Z$.*

The result is seen by the observation that $Y \subseteq Z$ implies $P^Z \subseteq P^Y$. Thus, $X \models P^Y$ implies $X \models P^Z$. In particular, for $X = Y$, $X \models P^X$ iff $X \models P$, and thus $X \models P$ implies $X \models P^Z$, for any $X \subseteq Z$.

Several notions of equivalence between logic programs have been considered in the literature (cf., e.g., (Lifschitz, Pearce, & Valverde 2001; Maher 1988; Sagiv 1988)). Under stable semantics, two DLPs P and Q are regarded as equivalent, denoted $P \equiv Q$, iff $\mathcal{SM}(P) = \mathcal{SM}(Q)$. The more restrictive forms of *strong equivalence* and *uniform equivalence* are as follows:

Definition 1 *Let P and Q be two DLPs. Then,*

1. *P and Q are* strongly equivalent*, or s-*equivalent*, denoted $P \equiv_s Q$, iff, for any set R of rules, the programs $P \cup R$ and $Q \cup R$ are equivalent, i.e., $P \cup R \equiv Q \cup R$; and*
2. *P and Q are* uniformly equivalent*, or u-*equivalent*, denoted $P \equiv_u Q$, iff, for any set F of normal facts, $P \cup F$ and $Q \cup F$ are equivalent, i.e., $P \cup F \equiv Q \cup F$.*

Obviously, $P \equiv_s Q$ implies $P \equiv_u Q$, but the converse does not always hold. Both notions of equivalence enjoy interesting semantical characterizations (Lifschitz, Pearce, & Valverde 2001; Turner 2001; 2003; Eiter & Fink 2003). As shown by Lifschitz, Pearce, & Valverde (2001), strong equivalence is closely related to the non-classical logic of here-and-there, which was adapted to logic-programming terms by Turner (2001; 2003):

Definition 2 *A pair (X, Y) with $X, Y \subseteq \mathcal{A}$ and $X \subseteq Y$ is called an* SE-interpretation (over $\mathcal{A}$). *By $INT_{\mathcal{A}}$ we denote the set of all SE-interpretations over $\mathcal{A}$. Furthermore, $(X, Y) \in INT_{\mathcal{A}}$ is an* SE-model (*over $\mathcal{A}$*) *of a DLP P, if $Y \models P$ and $X \models P^Y$. The set of all SE-models of P is denoted by $M_s^{\mathcal{A}}(P)$ (or simply by $M_s(P)$ if $\mathcal{A}$ is fixed).*

Proposition 2 (Turner 2001; 2003) *For every DLP P and Q, $P \equiv_s Q$ iff $M_s(P) = M_s(Q)$.*

SE-models can also be used to determine the stable models of a program (Pearce 1997; Lifschitz, Pearce, & Valverde 2001).

Proposition 3 *Let P be a DLP. Then, $Y \in \mathcal{SM}(P)$ iff $(Y, Y) \in M_s(P)$ and, for each $X \subset Y$, $(X, Y) \notin M_s(P)$.*

Recently, the following pendant to SE-models, characterizing uniform equivalence for (finite) logic programs, has been defined (Eiter & Fink 2003).

Definition 3 *Let P be a DLP and (X, Y) an SE-model of P. Then, (X, Y) is a* UE-model *of P iff, for every $(X', Y) \in M_s(P)$, it holds that $X \subset X'$ implies $X' = Y$. By $M_u(P)$ we denote the set of all UE-models of P.*

Proposition 4 (Eiter & Fink 2003) *For every DLP P and Q, $P \equiv_u Q$ iff $M_u(P) = M_u(Q)$.*

This test can be reformulated as follows.

Proposition 5 *For DLPs P and Q, $P \equiv_u Q$ iff $M_u(P) \subseteq M_s(Q)$ and $M_u(Q) \subseteq M_s(P)$.*

Proof. From Proposition 4, $P \equiv_u Q$ iff both $M_u(P) \subseteq M_u(Q)$ and $M_u(Q) \subseteq M_u(P)$ hold. Clearly, $M_u(R) \subseteq M_s(R)$ holds for any DLP R, which immediately gives the only-if direction. For the converse, suppose $P \not\equiv_u Q$. Hence, there exists an SE-interpretation (X, Y) such that either (i) $(X, Y) \in M_u(P)$ and $(X, Y) \notin M_u(Q)$, or (ii) $(X, Y) \in M_u(Q)$ and $(X, Y) \notin M_u(P)$. We only deal with Case (i); the second case proceeds analogously. Assume therefore that (i) holds. Then, by the definition of UE-models, there are two subcases to consider. First, $(X, Y) \notin M_s(Q)$. But then, $M_u(P) \subseteq M_s(Q)$ cannot hold. Second, there exists a set X' with $X \subset X' \subset Y$ and such that $(X', Y) \in M_u(Q)$. But $(X', Y) \notin M_s(P)$ since $(X, Y) \in M_u(P)$, hence $M_u(Q) \subseteq M_s(P)$ cannot hold. □

In the sequel, we shall write $M_\alpha(r)$ instead of $M_\alpha(\{r\})$, for a rule r and $\alpha \in \{s, u\}$.

As a final result here, we characterize the set of SE-models of a disjunctive rule.

Proposition 6 *Let r be a disjunctive rule and (X, Y) an SE-interpretation. Then, $(X, Y) \in M_s(r)$ iff one of the following conditions is satisfied: (i) $X \models H(r)$; (ii) $Y \not\models B(r)$; or (iii) $X \not\models B^+(r)$ and $Y \models H(r)$.*

Proof. By definition, $(X, Y) \in M_s(r)$ iff $Y \models r$ and $X \models r^Y$. The former holds iff either of $Y \models H(r)$, $Y \not\models B^+(r)$, or $Y \cap B^-(r) \neq \emptyset$ is satisfied. $X \models r^Y$ holds iff either of $X \models H(r)$, $X \not\models B^+(r)$, or $Y \cap B^-(r) \neq \emptyset$ holds. Hence, $(X, Y) \in M_s(r)$ iff $Y \cap B^-(r) \neq \emptyset$, or

$$Y \models H(r) \text{ or } Y \not\models B^+(r), \quad \text{and} \tag{2}$$

$$X \models H(r) \text{ or } X \not\models B^+(r) \tag{3}$$

jointly hold. Clearly, $Y \not\models B^+(r)$ implies $X \not\models B^+(r)$, and $X \models H(r)$ implies $Y \models H(r)$. From this, it is easily verified that (X, Y) satisfies (2) and (3) iff either $Y \not\models B^+(r)$ and $X \models H(r)$, or jointly $X \not\models B^+(r)$ and $Y \models H(r)$ holds. Hence, we have that $(X, Y) \in M_s(P)$ iff either $Y \cap B^-(r) \neq \emptyset$, $Y \not\models B^+(r)$, (i), or (iii) holds. Finally, $Y \cap B^-(r) \neq \emptyset$ or $Y \not\models B^+(r)$ holds exactly if $Y \not\models B(r)$ (i.e., (ii)) holds. □

Equivalence Results

In what follows, we provide general characterizations for eliminating disjunctions from programs. We first deal with a criterion for eliminating disjunctions under strong equivalence, and afterwards we show that such an elimination is always possible under uniform and ordinary equivalence.

Strong Equivalence

We start our analysis with some informal discussion. Consider the following logic programs, each of them having $r = a \vee b \leftarrow$ as its only disjunctive rule:

$$P_1 = \{a \vee b \leftarrow\};$$
$$P_2 = \{a \vee b \leftarrow; a \leftarrow\};$$
$$P_3 = \{a \vee b \leftarrow; a \leftarrow b\};$$
$$P_4 = \{a \vee b \leftarrow; a \leftarrow; \leftarrow not\, b\};$$
$$P_5 = \{a \vee b \leftarrow; a \leftarrow b; \leftarrow not\, b\};$$
$$P_6 = \{a \vee b \leftarrow; a \leftarrow; b \leftarrow\};$$
$$P_7 = \{a \vee b \leftarrow; a \leftarrow b; b \leftarrow a\};$$
$$P_8 = \{a \vee b \leftarrow; \leftarrow a, b\};$$
$$P_9 = \{a \vee b \leftarrow; \leftarrow not\, a; \leftarrow not\, b\}.$$

Let us first compute the SE-models (over $\mathcal{A} = \{a, b\}$) of these programs:[1]

$M_s(P_1) = \{\ (ab, ab),\ (a, ab),\ (b, ab),\ (a, a),\ (b, b)\ \};$
$M_s(P_2) = M_s(P_3) = \{\ (ab, ab),\ (a, ab),\ (a, a)\ \};$
$M_s(P_4) = M_s(P_5) = \{\ (ab, ab),\ (a, ab)\ \};$
$M_s(P_6) = M_s(P_7) = \{\ (ab, ab)\ \};$
$M_s(P_8) = \{\ (a, a),\ (b, b)\ \};$
$M_s(P_9) = \{\ (ab, ab)\ (a, ab),\ (b, ab)\ \}.$

[1] We write ab instead of $\{a, b\}$, and a instead of $\{a\}$, etc.

A good approximation for obtaining strongly equivalent normal logic programs is to replace $a \vee b \leftarrow$ by the two rules $a \leftarrow not\, b$ and $b \leftarrow not\, a$, i.e., by applying the usual shifting technique (Gelfond *et al.* 1991; Dix, Gottlob, & Marek 1996). It is easy to see that this replacement works for P_2, P_4, P_6, and P_8, but not for P_1, P_3, P_5, P_7, and P_9. As a matter of fact, for the latter programs, this replacement yields an additional SE-model, $(\emptyset, ab)$.

In general, we have the following relation between the SE-models of a disjunctive rule and its corresponding shifted rules.

Proposition 7 *For a disjunctive rule, r, define*

$$\begin{aligned} r^{\rightarrow} &= \{p \leftarrow B(r), not\,(H(r) \setminus \{p\}) \mid p \in H(r)\}\text{; and} \\ S_r &= \{(X,Y) \in INT_{\mathcal{A}} \mid X \not\models H(r),\ X \models B^+(r), \\ &\qquad Y \cap B^-(r) = \emptyset,\ |Y \cap H(r)| > 1\}. \end{aligned}$$

Then, $M_s(r^{\rightarrow}) = M_s(r) \cup S_r$.

Proof. Let r_p denote that rule in $r^{\rightarrow}$ with $H(r_p) = p$.

We first show that $S_r \subseteq M_s(r^{\rightarrow})$ and $M_s(r) \subseteq M_s(r^{\rightarrow})$. The former relation is easily seen, for each $r_p \in r^{\rightarrow}$: If $(X,Y) \in S_r$, then $|Y \cap H(r)| > 1$. Thus, $Y \cap B^-(r_p) \neq \emptyset$, yielding $(X,Y) \in M_s(r^{\rightarrow})$. For the latter relation, let $(X,Y) \in M_s(r)$. If $Y \not\models B(r)$ then $Y \not\models B(r_p)$, for each $r_p \in r^{\rightarrow}$. Hence, $(X,Y) \in M_s(r^{\rightarrow})$. So assume $Y \models B(r)$, and thus $Y \models H(r)$. Then, there exists some $p \in Y$ such that $p \in H(r)$. Hence, $Y \models r_p$. $(X,Y) \in M_s(r_p)$ is easily seen by similar arguments. Moreover, both $X \models (r^{\rightarrow})^Y$ and $Y \models r^{\rightarrow}$ hold, since for each $r_q \in r^{\rightarrow}$ with $p \neq q$, $Y \cap B^-(r_q) \neq \emptyset$ by definition.

It remains to show that $M_s(r^{\rightarrow}) \subseteq M_s(r) \cup S_r$. Therefore, consider some $(X,Y) \in M_s(r^{\rightarrow})$, and suppose that $(X,Y) \notin M_s(r)$. We show that $(X,Y) \in S_r$. Towards a contradiction, suppose that $(X,Y) \notin S_r$. Since $(X,Y) \notin M_s(r)$, we get by Proposition 6 that $X \not\models H(r)$, $Y \models B(r)$, and either $X \models B^+(r)$ or $Y \not\models H(r)$.

We consider two cases. First, if $Y \not\models H(r)$, i.e., $|Y \cap H(r)| = 0$, then we have a contradiction to the assumption that $(X,Y) \in M_s(r^{\rightarrow})$. Otherwise, if $Y \models H(r)$, we have $X \models B^+(r)$, and we get $|Y \cap H(r)| = 1$, otherwise $(X,Y) \in S_r$. Let $Y \cap H(r) = \{p\}$, and consider the rule r_p. Obviously, $Y \cap B^-(r_p) = \emptyset$. Moreover, we have $X \models B^+(r_p) = B^+(r)$. But then, $X \not\models H(r)$ yields $p \notin X$, which in turn contradicts $(X,Y) \in M_s(r^{\rightarrow})$. □

Observe that $M_s(r)$ and S_r are disjoint, in view of Proposition 6. Therefore, $S_r \cap M_s(P \setminus \{r\})$ is the (possibly empty) set of additional SE-models generated by the shifting process. In the above examples, we have $r = a \vee b \leftarrow$, $r^{\rightarrow} = \{a \leftarrow not\, b;\ b \leftarrow not\, a\}$, and $S_r = \{(\emptyset, ab)\}$.

The question is how to eliminate these additional SE-models. A possible method is to add suitable rules to the programs resulting from replacing disjunctive rules by shifted ones. For the above examples, it can be shown that adding $a \leftarrow$ to P_3, P_5, and P_7 does the job, since $(\emptyset, ab)$ is not an SE-model of the rule $a \leftarrow$, and, for each $(X,Y) \in M_s(P_i)$, $(X,Y) \in M_s(a \leftarrow)$, for $i \in \{3,5,7\}$. However, for P_1 and P_9, this does not work. The problem here is that both (a, ab) and (b, ab) are contained in $M_s(P_1)$ and $M_s(P_9)$, and it is not possible to "delete" the undesired interpretation $(\emptyset, ab)$ without "deleting" one of the necessary models (a, ab) or (b, ab) as well. Indeed, there is no normal logic program which is strongly equivalent to P_1 or to P_9.

Now, what is the distinguishing feature in these examples? In all of the above programs, except for P_1 or P_9, we have the property that with (X,Y) and (X',Y) being SE-models, $(X \cap X', Y)$ is an SE-model too. To wit, in the case of programs P_1 and P_9, the pair of SE-models (a, ab) and (b, ab) violates this property, since $(\emptyset, ab)$ is not contained in $M_s(P_1)$, resp. $M_s(P_9)$. This basic observation motivates the following definition:

Definition 4 *Let P be a DLP over $\mathcal{A}$. We say that P is* closed under here-intersection[2] *iff, for any pair (X,Y) and (X',Y) of SE-models of P, $(X \cap X', Y)$ is also an SE-model of P. We call $(X \cap X', Y)$ the* here-intersection of (X,Y) *and (X',Y), for any pair $(X,Y), (X',Y) \in INT_{\mathcal{A}}$.*

Hence, P_2–P_8 are closed under here-intersection, whilst P_1 and P_9 are not.

Lemma 1 *Each normal logic program is closed under here-intersection.*

Proof. Since P is normal, P^Y is Horn. Then, $X \models P^Y$ and $X' \models P^Y$ immediately implies $X \cap X' \models P^Y$, since each Horn program P' satisfies the following intersection property: if $X \models P'$ and $X' \models P'$, then $X \cap X' \models P'$. □

Hence, we can state the following result:

Theorem 1 *Let P be a DLP. If there exists a normal logic program Q such that P and Q are strongly equivalent, then P is closed under here-intersection.*

As discussed next, it turns out that closure under here-intersection is also a *sufficient* condition for the existence of a strongly equivalent normal program, given an arbitrary DLP. To this end, we introduce the following objects:

Definition 5 *Let P be a DLP over $\mathcal{A}$, let $r \in P$ be a disjunctive rule, and let $P_r^- = P \setminus \{r\}$. Then,*

$$P_{r,s} = P_r^- \cup r^{\rightarrow} \cup \hat{r}_{P_s},$$

where

$$\begin{aligned} \hat{r}_{P_s} &= \bigcup_{(X,Y,Z) \in S_r^{\uparrow}(P)} r_{X,Y,Z}; \\ r_{X,Y,Z} &= \{p \leftarrow X, not\,(\mathcal{A} \setminus Z) \mid p \in Y\}\text{; and} \\ S_r^{\uparrow}(P) &= \{(X,Y,Z) \mid (X,Z) \in S_r \cap M_s(P_r^-), \\ &\qquad X \subseteq Y,\ (Y,Z) \in M_s(P), \\ &\qquad \forall Y' : X \subseteq Y' \subset Y \Rightarrow (Y',Z) \notin M_s(P)\}. \end{aligned}$$

[2]The term "here-intersection" derives from the logical underpinning of strong equivalence (Lifschitz, Pearce, & Valverde 2001), given by the logic of here-and-there (Pearce 1997), in which the first component of an SE-interpretation refers to the world "here".

Intuitively, $S_r^{\uparrow}(P)$ collects, for each $(X,Z) \in S_r$ which is also an SE-model of P_r^-, the minimal SE-models (Y,Z) of P "above" X (with fixed Z). Note that, by definition of S_r, for any $(X,Z) \in INT_{\mathcal{A}}$, $(X,X,Z) \notin S_r^{\uparrow}(P)$, but $(X,Z) \in S_r$ implies the existence of an interpretation Y with $X \subseteq Y \subseteq Z$ such that $(X,Y,Z) \in S_r^{\uparrow}(P)$, since, at least for $Y = Z$, $(Y,Z) \in M_s(P)$ holds (again by definition of S_r).

The rules $r_{X,Y,Z}$, added in accord to the elements of $S_r^{\uparrow}(P)$, behave as follows:

Proposition 8 *For any sets $X,Y,Z \subseteq \mathcal{A}$, we have that $(X',Y') \in INT_{\mathcal{A}}$ is an SE-model of $r_{X,Y,Z}$ iff one of the following conditions holds: (i) $Y \subseteq X'$; (ii) $X \not\subseteq Y'$; (iii) $Y' \not\subseteq Z$; or (iv) $X \not\subseteq X'$ and $Y \subseteq Y'$.*

Proof. For $p \in Y$, let r_p denote the corresponding rule in $r_{X,Y,Z}$ with $H(r_p) = p$. Note that all rules in $r_{X,Y,Z}$ have the same body, $B = (X, not\ (\mathcal{A} \setminus Z))$. Clearly, $(X',Y') \in M_s(r_{X,Y,Z})$ iff $(X',Y') \in M_s(r_p)$, for each $p \in Y$. Applying Proposition 6 to each r_p yields $(X',Y') \in M_s(r_{X,Y,Z})$ iff (1) $Y' \not\models B$, or (2) for each $p \in Y$, $X' \models H(r_p)$ or jointly $X' \not\models B^+(r_p)$ and $Y' \models H(r_p)$ holds. Observe that (1) holds exactly if $X \not\subseteq Y'$ (i.e., Condition (ii)), or $Y' \cap (\mathcal{A} \setminus Z) \neq \emptyset$ (i.e., Condition (iii)) holds. It suffices to show that (2) is satisfied iff (i) or (iv) holds. Since $B^+(r_p) = X$, for each $p \in Y$, we proceed as follows. Assume $X' \not\models X$, i.e., $X \not\subseteq X'$. Then, either $X' \models H(r_p)$ or $Y' \models H(r_p)$, for each $p \in Y$. However, $Y' \models H(r_p)$ holds whenever $X' \models H(r_p)$. Hence, in this case, $Y' \models H(r_p)$ holds for all $p \in Y$. We thus arrive at $X \not\subseteq X'$ and $Y \subseteq Y'$, i.e., the properties of (iv). Otherwise, if $X \subseteq X'$, $X' \models H(r_p)$ must hold, for all $p \in Y$. Thus, $Y \subseteq X'$, i.e., Condition (i) holds. □

Note that $r_{X,Y,Z}$ may contain redundant rules, e.g., if we have $X \subseteq Y$. It can be shown that then $r_{X,Y,Z} \equiv_s r_{X,Y \setminus X,Z}$, which reduces the number of rules. However, for technical reasons we subsequently do not pay attention to this potential optimization.

Lemma 2 *If P is a DLP closed under here-intersection, then $M_s(P) = M_s(P_{r,s})$, for any disjunctive rule $r \in P$.*

Proof. First observe that, by Proposition 7, we have

$$\begin{aligned} M_s(P_{r,s}) &= M_s(P_r^- \cup r^{\rightarrow} \cup \hat{r}_{P_s}) = \\ &= M_s(P_r^-) \cap M_s(r^{\rightarrow}) \cap M_s(\hat{r}_{P_s}) \\ &= M_s(P_r^-) \cap (M_s(r) \cup S_r) \cap M_s(\hat{r}_{P_s}) \\ &= \big(M_s(P_r^-) \cap M_s(r) \cap M_s(\hat{r}_{P_s})\big) \cup \\ &\quad \big(M_s(P_r^-) \cap S_r \cap M_s(\hat{r}_{P_s})\big) \\ &= \big(M_s(P) \cap M_s(\hat{r}_{P_s})\big) \cup \\ &\quad \big(M_s(P_r^-) \cap S_r \cap M_s(\hat{r}_{P_s})\big). \end{aligned}$$

The strategy for the remainder of the proof is as follows. We first show that $T = M_s(P_r^-) \cap S_r \cap M_s(\hat{r}_{P_s}) = \emptyset$. This leads us to $M_s(P_{r,s}) = M_s(P) \cap M_s(\hat{r}_{P_s})$, and thus it remains to show that $M_s(P) = M_s(P) \cap M_s(\hat{r}_{P_s})$, i.e., that $M_s(P) \subseteq M_s(\hat{r}_{P_s})$ holds.

We show $T = \emptyset$. Let $(X,Z) \in S_r$. If $(X,Z) \notin M_s(P_r^-)$, we have $(X,Z) \notin T$. So suppose $(X,Z) \in M_s(P_r^-)$. Then, there exists a triple $(X,Y,Z) \in S_r^{\uparrow}(P)$. Hence, we can assume $r_{X,Y,Z} \subseteq \hat{r}_{P_s}$. Moreover, we have $X \subset Y \subseteq Z$. We now show that $(X,Z) \notin M_s(r_{X,Y,Z})$. Assume $(X,Z) \in M_s(r_{X,Y,Z})$. Then, by Proposition 8, one of the following conditions has to hold: (i) $Y \subseteq X$, (ii) $X \not\subseteq Z$, (iii) $Z \not\subseteq Z$, or (iv) $X \not\subseteq X$ and $Y \subseteq Z$. Condition (i) does not hold since we have $X \subset Y$; (ii) does not hold since we get $X \subset Z$ from $X \subset Y \subseteq Z$; and (iii) and (iv) do not hold trivially. We arrive at a contradiction, and we get $T = \emptyset$.

We now show that $M_s(P) \subseteq M_s(\hat{r}_{P_s})$ holds. Clearly, if $\hat{r}_{P_s}$ is empty, we are done, since then each SE-interpretation is also an SE-model of $\hat{r}_{P_s}$. So, suppose $\hat{r}_{P_s} \neq \emptyset$. We show $(X',Z') \in M_s(r_{X,Y,Z})$, for each $(X',Z') \in M_s(P)$ and each $r_{X,Y,Z} \subseteq \hat{r}_{P_s}$. Towards a contradiction, consider some $r_{X,Y,Z} \subseteq \hat{r}_{P_s}$ and some $(X',Z') \in M_s(P)$ such that $(X',Z') \notin M_s(r_{X,Y,Z})$. On the one hand, from $r_{X,Y,Z} \subseteq \hat{r}_{P_s}$, we have $(X,Y,Z) \in S_r^{\uparrow}(P)$, which implies that (a) $(X,Z) \in S_r \cap M_s(P_r^-)$; (b) $(Y,Z) \in M_s(P)$; and (c) $X \subset Y \subseteq Z$. On the other hand, by Proposition 8, we know from $(X',Z') \notin M_s(r_{X,Y,Z})$ that (1) $Y \not\subseteq X'$, (2) $X \subseteq Z'$, (3) $Z' \subseteq Z$, and (4) $X \subseteq X'$ or $Y \not\subseteq Z'$.

By assumption, $(X',Z') \in M_s(P)$. Hence, $X' \models P^{Z'}$ and $Z' \models P^{Z'}$. Moreover, by (3), $Z' \subseteq Z$ holds. By Proposition 1, we get $X' \models P^{Z}$ and $Z' \models P^{Z}$, and from (b) we get $Z \models P$. Therefore, $(X',Z) \in M_s(P)$ and $(Z',Z) \in M_s(P)$. Now, P is closed under here-intersection, yielding $(Y \cap X', Z) \in M_s(P)$ and $(Y \cap Z', Z) \in M_s(P)$. We use (4) to distinguish between the following two cases:

First, assume $X \subseteq X'$. By (c), $X \subset Y$, and thus $X \subseteq (Y \cap X')$. From (1), we have that $Y \not\subseteq X'$. This implies $(Y \cap X') \subset Y$. Hence, $X \subseteq (Y \cap X') \subset Y$ follows. Together with (a) and $(Y \cap X', Z) \in M_s(P)$, we obtain $(X, Y \cap X', Z) \in S_r^{\uparrow}(P)$, contradicting $(X,Y,Z) \in S_r^{\uparrow}(P)$.

Now assume $X \not\subseteq X'$. Similarly, from (4), we get $Y \not\subseteq Z'$, yielding $(Y \cap Z') \subset Y$. Moreover, by (2), $X \subseteq Z'$, and by (c), $X \subset Y$. Thus, $X \subseteq (Y \cap Z')$. Again, we have $X \subseteq (Y \cap Z') \subset Y$, and by (a) and $(Y \cap Z', Z) \in M_s(P)$, we arrive at a contradiction to $(X,Y,Z) \in S_r^{\uparrow}(P)$. □

Hence, by applying the above transformation successively for all disjunctive rules in a given DLP P, we eventually obtain a normal logic program strongly equivalent to P.

Theorem 2 *Let P be a DLP over $\mathcal{A}$. If P is closed under here-intersection, then there exists a normal logic program Q over $\mathcal{A}$ such that P and Q are strongly equivalent.*

Therefore, in view of Theorem 1, a DLP P possesses a strongly equivalent NLP *precisely if* P is closed under here-intersection.

For illustration, let us apply the construction of $P_{r,s}$ to the examples P_2 and P_3 from the above, with $r = a \vee b \leftarrow$. Consider $S_r^{\uparrow}(P_2)$. Clearly, $S_r \cap M_s((P_2)_r^-) = \emptyset$, and r is just replaced by $r^{\rightarrow}$. But $S_r \cap M_s((P_3)_r^-) = \{(\emptyset, ab)\}$, and by the SE-models of P_3, we get $S_r^{\uparrow}(P_3) = \{(\emptyset, a, ab)\}$. Hence, we exchange r in P_3 by $r^{\rightarrow} \cup r_{\emptyset,a,ab}$, where $r_{\emptyset,a,ab} = \{a \leftarrow\}$ (under the assumption that $\mathcal{A} = \{a,b\}$). For the other programs, the construction is similar.

Uniform Equivalence

If we change from strong to uniform equivalence, and retaining our overall strategy, the intuitive problems are very similar to those observed in the case of strong equivalence. But now an SE-model (X, Y) from S_r comes into play only if it is also a UE-model of the remaining program. Thus, if we want to eliminate such an SE-model, the problem of eliminating further SE-models, which should be retained, is less complicated compared to the case of strong equivalence. Roughly speaking, because of this difference, we are *always* able to construct a uniformly equivalent normal program. For instance, all our example programs P_1–P_9 except P_7 are uniformly equivalent to the program resulting from P_i by replacing $r = a \vee b \leftarrow$ by its shifting $r^{\rightarrow}$. For the program $P_7 = \{r; a \leftarrow b; b \leftarrow a\}$, however, we obtain $(\emptyset, ab)$ as additional SE- *and* UE-model of $(P_7 \setminus \{a \vee b \leftarrow\}) \cup \{a \leftarrow not\, b; b \leftarrow not\, a\}$. Adding rules $a \leftarrow$ or $b \leftarrow$ (or both of them) circumvents this problem. Hence, in some cases, but in fewer than for strong equivalence, we again have to add further rules to achieve our goal. To wit, only rules $r_{X,Y,Z}$, for $(X, Y, Z) \in S_r^{\uparrow}(P)$, are used where $Y = Z$.

Definition 6 *Let P, r, P_r^-, $S_r^{\uparrow}(P)$, and $r_{X,Y,Z}$ be as in Definition 5, and define*

$$P_{r,u} = P_r^- \cup r^{\rightarrow} \cup \hat{r}_{P_u}, \text{ for } \hat{r}_{P_u} = \bigcup_{(X,Z,Z) \in S_r^{\uparrow}(P)} r_{X,Z,Z}.$$

In contrast to the case of strong equivalence, this transformation is always applicable in order to retain uniform equivalence.

Lemma 3 *Given a DLP P with $r \in P$ disjunctive, it holds that $M_u(P) = M_u(P_{r,u})$.*

Proof. First of all, observe that, analogously to the proof of Lemma 2, we get that $M_s(P_{r,u})$ is given by

$$\big(M_s(P) \cap M_s(\hat{r}_{P_u})\big) \cup \big(M_s(P_r^-) \cap S_r \cap M_s(\hat{r}_{P_u})\big). \quad (4)$$

We show $M_u(P) = M_u(P_{r,u})$. By Proposition 5, this holds iff both $M_u(P) \subseteq M_s(P_{r,u})$ and $M_u(P_{r,u}) \subseteq M_s(P)$ hold.

We first show $M_s(P) \subseteq M_s(P_{r,u})$, which clearly implies $M_u(P) \subseteq M_s(P_{r,u})$. Note that if $\hat{r}_{P_u}$ is empty, we are done, since then $M_s(P) \subseteq M_s(P) \cap M_s(\hat{r}_{P_u})$ holds trivially. So consider $\hat{r}_{P_u} \neq \emptyset$. We show $(X', Y') \in M_s(r_{X,Y,Y})$, for each $(X', Y') \in M_s(P)$ and each $r_{X,Y,Y} \subseteq \hat{r}_{P_u}$. Towards a contradiction, consider $r_{X,Y,Y} \subseteq \hat{r}_{P_u}$ and $(X', Y') \in M_s(P)$ such that $(X', Y') \notin M_s(r_{X,Y,Y})$. On the one hand, since $r_{X,Y,Y} \subseteq \hat{r}_{P_u}$, we have (a) $(X, Y) \in S_r \cap M_s(P_r^-)$, (b) $X \subset Y$, and (c) for each SE-interpretation (Z, Y) with $X \subseteq Z$, $(Z, Y) \in M_s(P)$ implies $Z = Y$. By Proposition 8, on the other hand, $(X', Y') \notin M_s(r_{X,Y,Y})$ yields (i) $Y \not\subseteq X'$, (ii) $X \subseteq Y'$, (iii) $Y' \subseteq Y$, and (iv) $X \subseteq X'$ or $Y \not\subseteq Y'$. We use (iv) for distinguishing between the following two cases:

First, assume $X \subseteq X'$. Clearly, $X' \subseteq Y'$. By (i), $X' \neq Y$, and by (iii), $Y' \subseteq Y$. We thus get $X \subseteq X' \subset Y$. Moreover, $X' \models P^{Y'}$ holds, since $(X', Y') \in M_s(P)$. By Proposition 1, we get $X' \models P^Y$. Furthermore, $Y \models P$ holds by (a). Hence $(X', Y) \in M_s(P)$, which clearly is in contradiction to (c).

Second, assume $X \not\subseteq X'$. By (iv), $Y \not\subseteq Y'$. Together with (iii), we thus have $Y' \subset Y$. Moreover, $X \subseteq Y'$ holds by (ii). Since $(X', Y') \in M_s(P)$, $Y' \models P^{Y'}$ holds. Proposition 1 yields $Y' \models P^Y$, and since $Y \models P$, we have $(Y', Y) \in M_s(P)$ with $X \subseteq Y' \subset Y$. Again, this is in violation to (c).

It remains to show that $M_u(P_{r,u}) \subseteq M_s(P)$. In particular, we show that $M_u(P_{r,u}) \cap T = \emptyset$ holds, where

$$T = M_s(P_r^-) \cap S_r \cap M_s(\hat{r}_{P_u}).$$

By inspecting (4), it can be seen that $M_u(P_{r,u}) \cap T = \emptyset$ implies $M_u(P_{r,u}) \subseteq M_s(P) \cap M_s(\hat{r}_{P_u})$, which proves the claim since $M_s(P) \cap M_s(\hat{r}_{P_u}) \subseteq M_s(P)$ holds trivially.

To derive $M_u(P_{r,u}) \cap T = \emptyset$, we show that for any $(X, Y) \in S_r \cap M_s(P_r^-)$, either $(X, Y) \notin M_u(P_{r,u})$ or $(X, Y) \notin M_s(\hat{r}_{P_u})$ holds. Fix a $(X, Y) \in S_r \cap M_s(P_r^-)$. We consider two cases.

Assume $(X, Y, Y) \notin S_r^{\uparrow}(P)$. Hence, there exists a set $X \subseteq X' \subset Y$ such that $(X', Y) \in M_s(P)$. We know that $(X, X, Y) \notin S_r^{\uparrow}(P)$. Thus, $X \subset X'$. We already have shown that $M_s(P) \subseteq M_s(P_{r,u})$, yielding $(X', Y) \in M_s(P_{r,u})$. But then, $(X, Y) \notin M_u(P_{r,u})$, since $X \subset X' \subset Y$.

So assume $(X, Y, Y) \in S_r^{\uparrow}(P)$, and thus $r_{X,Y,Y} \subseteq \hat{r}_{P_u}$. However, we have $(X, Y) \notin M_s(r_{X,Y,Y})$, since none of the following conditions, which hold by Proposition 8, is satisfied: (i) $Y \subseteq X$, (ii) $X \not\subseteq Y$, (iii) $Y \not\subseteq Y$, or (iv) $X \not\subseteq X$ and $Y \subseteq Y$. For (i) and (ii), this is seen by the fact that $(X, Y) \in S_r$, and thus $X \subset Y$; and (iii) and (iv) fail trivially. Hence, $(X, Y) \notin M_s(\hat{r}_{P_u})$. □

Theorem 3 *For each DLP P, there exists a normal program Q such that $P \equiv_u Q$.*

As already discussed above, the only program from our examples P_1–P_9 which is not uniformly equivalent after replacing $r = a \vee b \leftarrow$ by $r^{\rightarrow}$ is P_7. However, since P_7 is closed under here-intersection, we already know how to derive a strongly (and thus uniformly) equivalent normal logic program. In fact, one can verify that $(P_7)_{r,s} = (P_7)_{r,u}$. For an example program P which is not closed under here-intersection and such that $\hat{r}_{P_u} \neq \emptyset$, with $r \in P$ disjunctive, consider $P = \{a \vee b \leftarrow; a \leftarrow c, b; b \leftarrow c, a\}$ over $\mathcal{A} = \{a, b, c\}$. The SE-models of P are given as follows:

$$M_s(P) = \{(abc, abc), (ab, abc), (ab, ab), (a, abc), (a, ab), (a, a), (b, abc), (b, ab), (b, b)\}.$$

Indeed, P is not closed under here-intersection. S_r is given by $\{(\emptyset, ab), (\emptyset, abc), (c, abc)\}$, and $S_r \cap M_s(P_r^-) = S_r$ holds. Observe that in contrast to (c, abc), the remaining elements in S_r, i.e., $(\emptyset, ab)$ and $(\emptyset, abc)$, are not problematic, since adding them to $M_s(P)$ does not change the set of UE-models. $S_r^{\uparrow}(P)$ is given by the set

$$\{(\emptyset, a, ab), (\emptyset, b, ab), (\emptyset, a, abc), (\emptyset, b, abc), (c, abc, abc)\},$$

but only the last triple, (c, abc, abc), is applied in the construction of $\hat{r}_{P_u}$. In fact, we have to add

$$r_{c,abc,abc} = \{a \leftarrow c;\ b \leftarrow c;\ c \leftarrow c\}$$

to $P_r^- \cup r^{\rightarrow}$. For the resulting normal program $P_{r,u}$, we then have $M_s(P_{r,u}) = M_s(P) \cup \{(\emptyset, ab), (\emptyset, abc)\}$, but the "critical" SE-interpretation, (c, abc), has been eliminated. In fact, $M_u(P) = M_u(P_{r,u})$ holds, since neither $(\emptyset, ab)$ nor $(\emptyset, abc)$ is a UE-model of $P_{r,u}$.

Ordinary Equivalence

Finally, we discuss the case of ordinary equivalence. Since uniform equivalence implies ordinary equivalence, in view of Theorem 3, for any DLP P, there always exists an NLP Q such that P and Q are equivalent. Moreover, the normal program obtained by successive applications of transformation $P_{r,u}$ clearly does the job. Hence:

Theorem 4 *For each DLP P, there exists a normal program Q such that $P \equiv Q$.*

In fact, this result is also obtained by an enumeration of stable models.

Theorem 5 *Let P be a DLP and $r_{X,Y,Z}$ as in Definition 5. Then, $\mathcal{SM}(P) = \mathcal{SM}(\tilde{P})$, with $\tilde{P} = \bigcup_{Y \in \mathcal{SM}(P)} r_{\emptyset,Y,Y}$.*

Proof. To begin with, we note the following property, which is a simple consequence of Proposition 8:

($*$) For any $Y \subseteq \mathcal{A}$, $M_s(r_{\emptyset,Y,Y})$ is given by $\{(X', Y') \in INT_{\mathcal{A}} \mid Y \subseteq X' \text{ or } Y' \not\subseteq Y\}$.

Consider $Y \in \mathcal{SM}(P)$. Then, for each $Y' \in \mathcal{SM}(P)$, either $Y \not\subseteq Y'$ or $Y = Y'$. By ($*$), $(Y, Y) \in M_s(\tilde{P})$. Towards a contradiction, suppose $Y \notin \mathcal{SM}(\tilde{P})$. By Proposition 3, there exists a $X \subset Y$ such that $(X, Y) \in M_s(\tilde{P})$. In particular, we must have $(X, Y) \in M_s(r_{\emptyset,Y,Y})$. But by ($*$), this is impossible in view of $X \subset Y$. Therefore, $Y \in \mathcal{SM}(\tilde{P})$. This proves $\mathcal{SM}(P) \subseteq \mathcal{SM}(\tilde{P})$.

Suppose there is some $Y \in \mathcal{SM}(\tilde{P})$ such that $Y \notin \mathcal{SM}(P)$. That is, $(X, Y) \in M_s(P)$ for some $X \subset Y$. By Proposition 1, we get $(X, Z) \in M_s(P)$ for each $Y \subseteq Z$. Thus, for each $Y' \in \mathcal{SM}(P)$, $Y \not\subseteq Y'$ holds. Hence, we have $(X, Y) \in M_s(r_{\emptyset,Y',Y'})$, for each $Y' \in \mathcal{SM}(P)$, by ($*$). Thus, $(X, Y) \in M_s(\tilde{P})$ with $X \subset Y$ But this contradicts $Y \in \mathcal{SM}(\tilde{P})$, and thus $\mathcal{SM}(P) = \mathcal{SM}(\tilde{P})$ must hold. □

Finally, we also note the following transformation for ordinary equivalence, following the line of $P_{r,s}$ and $P_{r,u}$, but being more compact with respect to ordinary equivalence.

Lemma 4 *Let P, r, P_r^-, $S_r^{\uparrow}(P)$, and $r_{X,Y,Z}$ be as in Definition 5, and define*

$$P_{r,e} = P_r^- \cup r^{\rightarrow} \cup \hat{r}_{P_e},$$

where

$$\hat{r}_{P_e} = \bigcup_{\substack{(X,Z,Z) \in S_r^{\uparrow}(P),\\ Z \in \mathcal{SM}(P)}} r_{X,Z,Z}.$$

Then, $\mathcal{SM}(P) = \mathcal{SM}(P_{r,e})$.

Proof. We first show that,

($*$) for each $Y \subseteq \mathcal{A}$, $Y \models P$ iff $Y \models P_{r,e}$.

Indeed, from Lemma 3, we have $M_u(P) = M_u(P_{r,u})$, and thus $Y \models P$ iff $Y \models P_{r,u}$. By definition, $P_{r,e} \subseteq P_{r,u}$, from which the only-if direction is an immediate consequence. For the proof of the if-direction, assume $Y \models P_{r,e}$. Then, $Y \models P_r^- \cup r^{\rightarrow}$. By classical logic, this clearly implies $Y \models P$.

We proceed with the proof of the lemma. First, fix some $Y \in \mathcal{SM}(P)$. Then, $Y \models P$, and by ($*$), $Y \models P_{r,e}$. It remains to show that no $X \subset Y$ yields an SE-model (X, Y) of $P_{r,e}$. Towards a contradiction, suppose some $X \subset Y$ exists such that $(X, Y) \in M_s(P_{r,e})$. Clearly, $(X, Y) \in M_s(P_r^-)$, hence, since $Y \in \mathcal{SM}(P)$, $(X, Y) \notin M_s(r)$ must hold. Then, by Proposition 7, $(X, Y) \in S_r$. Therefore, $(X, Y) \in S_r \cap M_s(P_r^-)$ and $Y \in \mathcal{SM}(P)$, and we thus get $r_{X,Y,Y} \in P_{r,e}$ by construction. By Proposition 8, $(X, Y) \in M_s(r_{X,Y,Y})$ only if $X = Y$. Thus $(X, Y) \notin M_s(P_{r,e})$, which is a contradiction. Hence, $Y \in \mathcal{SM}(P_{r,e})$, which proves $\mathcal{SM}(P) \subseteq \mathcal{SM}(P_{r,e})$.

Now consider some $Y \in \mathcal{SM}(P_{r,e})$. By ($*$), we get $Y \models P$. Assume $Y \notin \mathcal{SM}(P)$, i.e., there exists a set $X \subset Y$ such that $(X, Y) \in M_s(P)$. By Proposition 7, $(X, Y) \in M_s(P_r^- \cup r^{\rightarrow})$. Since $Y \models P$, and thus $Y \models P^Y$, we get by Proposition 1 that $Y \models P^{Y'}$, for each Y' with $Y \subseteq Y'$. Hence, $Y \not\subseteq Y'$, for each $Y' \in \mathcal{SM}(P)$. By Proposition 8, $(X, Y) \in M_s(r_{X,Y',Y'})$, for each Y' with $Y \not\subseteq Y'$. Hence, $(X, Y) \in M_s(P_{r,e})$. By Proposition 3, this contradicts $Y \in \mathcal{SM}(P_{r,e})$. Thus $Y \in \mathcal{SM}(P)$ must hold, which proves $\mathcal{SM}(P_{r,e}) \subseteq \mathcal{SM}(P)$. □

To summarize, given a DLP P with $r \in P$ disjunctive, we are able to construct (via a replacement of r by normal rules)

- a logic program $P_{r,e}$ which is ordinary equivalent to P;
- a program $P_{r,u}$ which is uniformly equivalent to P; and
- a program $P_{r,s}$ which is strongly equivalent to P, whenever P is closed under here-intersection.

All these programs are of the form

$$P_{r,\alpha} = P_r^- \cup r^{\rightarrow} \cup \hat{r}_{P_\alpha}, \text{ for } \alpha \in \{e, u, s\},$$

and satisfy

$$P_{r,e} \subseteq P_{r,u} \subseteq P_{r,s}.$$

Recall that successive applications of these rulewise transformations, for all disjunctive rules in a given program, leads to a normal logic program. Hence, our method can be seen as a uniform framework for obtaining normal logic programs from disjunctive logic programs with respect to all important notions of equivalence. Moreover, our results extend and generalize methods based on shifting techniques, since the outcome of these methods coincides with the present rewriting $P_{r,\alpha}$, whenever $\hat{r}_{P_\alpha}$ is empty. In particular, concerning equivalence in terms of stable models, we present a semantic criterion (in contrast to the syntactic criterion of head-cycle freeness as discussed by Ben-Eliyahu & Dechter (1994)) which allows for shifting. Moreover, in terms of uniform

equivalence, we generalized also an observation made by Eiter & Fink (2003) (cf. Theorem 4.3 of their paper).

We note that the size of the outcoming programs in our method is in general exponential in the size of the input program. However, as we discuss in the next section, this exponential increase is, in a certain sense, unavoidable from a complexity-theoretic point of view.

Complexity Issues

This section deals with complexity issues related to the results discussed above. We first analyse the complexity of checking closure of a DLP under here-intersection. Afterwards, we investigate the expressiveness of the class of DLPs closed under here-intersection. As we show, this class resides at the same level of the polynomial hierarchy (PH) as the class of arbitrary DLPs. Finally, we show that the exponential increase of program size in the worst case of our general rewriting method is unavoidable, providing the PH does not collapse.

Checking Closure under Here-Intersection

We can express testing a DLP P for being closed under here-intersection via the following normal logic program, which is linear in the size of P. To this end, for any rule r, let r'_i denote the rule obtained from r by replacing each occurrence of an atom p_i in r by p'_i.

Definition 7 *Let P be a DLP over atoms V. For each atom $v \in V$, let $\bar{v}$, v'_1, $\bar{v}'_1$, v'_2, $\bar{v}'_2$, v'_3 be pairwise distinct new atoms, and let u be a new atom. Define P_Q as the program containing the following items:*

1. *for each $v \in V$ and each $i \in \{1, 2\}$,*

$$v \leftarrow not\ \bar{v}; \qquad \bar{v} \leftarrow not\ v; \tag{5}$$

$$v'_i \leftarrow v, not\ \bar{v}'_i; \quad \bar{v}'_i \leftarrow not\ v'_i; \tag{6}$$

2. *for each $r \in P$ and each $i \in \{1, 2\}$,*

$$\leftarrow B(r), not\ H(r); \tag{7}$$

$$\leftarrow B^+(r'_i), not\ B^-(r), not\ H(r'_i); \tag{8}$$

3. *for each $v \in V$,*

$$v'_3 \leftarrow v'_1, v'_2; \tag{9}$$

4. *for each $r \in P$,*

$$u \leftarrow B^+(r'_3), not\ B^-(r), not\ H(r'_3); \tag{10}$$

and

5. *the constraint*

$$\leftarrow not\ u. \tag{11}$$

Intuitively, the program P_Q works as follows. Rules of form (5) guess an interpretation Y of P, and rules of form (7) check that Y is a model of P. Similarly, rules of form (6) guess subsets X_1 and X_2 of Y such that both are models of P^Y, which is enforced by the constraints of form (8). Hence, the program consisting of all rules of form (5)–(8) "computes" all pairs of SE-models (X_1, Y) and (X_2, Y) of P.

Now, the rules of form (9) compute the intersection $X_1 \cap X_2$, and via the rules of form (10) the new atom u can be derived iff the intersection does not satisfy P^Y, i.e., iff $(X_1 \cap X_2, Y)$ is no SE-model of P. The constraint (11) eliminates all models of P_Q for which this is not the case, i.e., for which $(X_1 \cap X_2, Y)$ is an SE-model of P. Thus, items (9)–(11) ensure that P_Q has no stable model iff P is closed under here-intersection. Formally, we have:

Lemma 5 *A DLP P is closed under here-intersection iff $\mathcal{SM}(P_Q) = \emptyset$.*

Based on this, we derive the following complexity result.

Theorem 6 *Checking closure under here-intersection, for a given DLP, is* coNP*-complete.*

Proof. By Lemma 5 and the linear encoding from Definition 7, we get that closure under here-intersection is in coNP.

We show coNP-hardness by a reduction to the coNP-complete problem of deciding whether a given interpretation is the unique model of a positive DLP as follows:

Let P be a positive DLP over atoms $V = \{v_1, \ldots, v_n\}$, let q, q' be new atoms, and consider the program

$$Q = P \cup \{q \vee q' \leftarrow;\ q \leftarrow v_1, \ldots, v_n;\ q' \leftarrow v_1, \ldots, v_n\}.$$

We show that Q is closed under here-intersection iff V is the unique model of P.

First, if V is the unique model of P then, by construction, $V \cup \{q, q'\}$ is the unique model of Q, and since Q is positive—and hence constant under reduction—Q is trivially closed under here-intersection.

Second, for the only-if direction, assume that Q is closed under here-intersection. Towards a contradiction, assume that there is a model $M \subset V$ of P. Then, both $M \cup \{q\}$ and $M \cup \{q'\}$ are models of Q, and thus also both $(M \cup \{q\}, V \cup \{q, q'\})$ and $(M \cup \{q'\}, V \cup \{q, q'\})$ are SE-models of Q. However, $(M, V \cup \{q, q'\})$ is not an SE-model of Q, since $M \not\models q \vee q' \leftarrow$. This contradicts our assumption that Q is closed under here-intersection. Hence, V is the unique model of P. □

In view of Theorem 2, the above result immediately implies the following property:

Corollary 1 *Checking whether, for a given DLP P, there exists a normal program Q such that P and Q are strongly equivalent, is* coNP*-complete.*

Expressiveness of Here-Intersection closed DLPs

We now consider the expressiveness of the class of DLPs closed under here-intersection. We show that this class of programs possesses the same worst-case complexity as arbitrary DLPs. More specifically, the relevant reasoning tasks in the context of DLPs are

- checking the existence of a stable model of a given DLP ("consistency problem");
- checking whether a given atom belongs to at least one stable model of a given DLP ("brave reasoning"); and

- checking whether a given atom belongs to all stable models of a given DLP ("cautious reasoning").

As shown by Eiter & Gottlob (1995), for arbitrary disjunctive programs, the consistency problem and brave reasoning are Σ_2^P-complete, whilst cautious reasoning is Π_2^P-complete. Moreover, the respective hardness results for these problems hold even for a quite restricted class of DLPs. This class comprises DLPs where each disjunctive rule is a fact and where each stable model contains an *exact hitting set*[3] for the collection of disjunctive facts in P.

We define the following kinds of programs:

Definition 8 *A DLP P is called*

1. *a* disjunctive-fact program (*DFP*) *iff each disjunctive rule in P has an empty body, i.e., for each $r \in P$, $|H(r)| > 1$ implies $B(r) = \emptyset$; and*
2. *a* hitting-set program *iff, for each stable model $I \in \mathcal{SM}(P)$, $|H(r) \cap I| = 1$, for each disjunctive rule $r \in P$.*

DLPs satisfying both conditions are called hitting-set DFPs (*HDFPs*).

Observe that HDFPs are in general *not* closed under here-intersection, as seen by the program $\{a \vee b \leftarrow\}$. However, we construct a polynomial-time translation mapping each DFP P into a DLP Q (over an extended alphabet) such that (i) Q is closed under here-intersection and (ii) there is a one-to-one correspondence (over the original alphabet) between the stable models of P and the stable models of Q, whenever P is also a HDFP. From this, we derive the same lower complexity bounds for DLPs closed under here-intersection as for arbitrary DLPs.

We employ the following notation: For a given alphabet $\mathcal{A}$, $\overline{\mathcal{A}} = \{\overline{p} \mid p \in \mathcal{A}\}$ is a set of globally new disjoint atoms. Accordingly, for a rule r, $\overline{r}$ is the rule resulting from r by replacing each atom p in r by $\overline{p}$. Finally, let $\mathcal{A}^* = \mathcal{A} \cup \overline{\mathcal{A}}$.

Definition 9 *For a DLP P over $\mathcal{A}$, let P^+ be the program resulting from P by adding $\{p \leftarrow \overline{p} \mid p \in \mathcal{A}\}$ and, for each r with $|H(r)| > 1$, adding $(\overline{r})^{\rightarrow} \cup \{\leftarrow \overline{p}, \overline{q} \mid p, q \in H(r), p \neq q\}$.*

Lemma 6 *For a DFP P, we have that $M_s^{\mathcal{A}^*}(P^+)$ is given by those $(X, Y) \in INT_{\mathcal{A}^*}$ which satisfy the following conditions:*

1. *$(X, Y) \in M_s^{\mathcal{A}^*}(P)$;*
2. *$X \cap \overline{\mathcal{A}} \subseteq \overline{(X \cap \mathcal{A})}$, $Y \cap \overline{\mathcal{A}} \subseteq \overline{(Y \cap \mathcal{A})}$; and*
3. *for each $r \in P$ with $|H(r)| > 1$,*

$$X \cap H(\overline{r}) = Y \cap H(\overline{r}) = \{\overline{p}\}.$$

Proof. First, $M_s^{\mathcal{A}^*}(P \cup \{p \leftarrow \overline{p} \mid p \in \mathcal{A}\})$ is clearly given by those $(X, Y) \in INT_{\mathcal{A}^*}$ which satisfy Conditions 1 and 2. Furthermore, consider a disjunctive fact $r \in P$. Applying Proposition 7, we get $M_s^{\mathcal{A}^*}((\overline{r})^{\rightarrow}) = M_s^{\mathcal{A}^*}(\overline{r}) \cup S_{\overline{r}}$, which in turn is given by

$$\{(X, Y) \in INT_{\mathcal{A}^*} \mid H(\overline{r}) \cap X \neq \emptyset \text{ or } H(\overline{r}) \cap X = \emptyset; |Y \cap H(\overline{r})| > 1\}.$$

In the construction of P^+, it remains to consider the rules $R = \{\leftarrow \overline{p}, \overline{q} \mid p, q \in H(r), p \neq q\}$ having as its SE-models (over $\mathcal{A}^*$)

$$\{(X, Y) \in INT_{\mathcal{A}^*} \mid |Y \cap H(\overline{r})| \leq 1\}.$$

Putting things together, for each disjunctive fact $r \in P$, we get as SE-models (over $\mathcal{A}^*$) of rules $\overline{r}^{\rightarrow} \cup R$ exactly those $(X, Y) \in INT_{\mathcal{A}^*}$ which satisfy Condition 3. □

For illustration, reconsider $P_1 = \{a \vee b \leftarrow\}$ over $\mathcal{A} = \{a, b\}$. We get P_1^+ given by

$$\{a \vee b \leftarrow;\ a \leftarrow \overline{a};\ b \leftarrow \overline{b};\ \overline{a} \leftarrow not\ \overline{b};\ \overline{b} \leftarrow not\ \overline{a};\ \leftarrow \overline{a}, \overline{b}\}.$$

The set of SE-models (over $\{a, b, \overline{a}, \overline{b}\}$) of P_1^+ is then given by

$$\{(a\overline{a}, a\overline{a}),\ (b\overline{b}, b\overline{b}),\ (a\overline{a}, ab\overline{a}),\ (b\overline{b}, ab\overline{b}),\ (ab\overline{a}, ab\overline{a}),\ (ab\overline{b}, ab\overline{b})\}.$$

Observe that P_1^+ is closed under here-intersection, whilst P_1 is not. This example mirrors the following general property:

Lemma 7 *For any DFP P, P^+ is closed under here-intersection.*

Proof. If P is an NLP, then P^+ is clearly closed under here-intersection. So suppose that $P \in \mathcal{DLP} \setminus \mathcal{NLP}$, and assume that P^+ is not closed under here-intersection. Then, there exist SE-models (X, Y) and (Z, Y) of P^+ such that $(X \cap Z, Y) \notin M_s(P^+)$. Let P_1 consist of all normal rules in P^+ and P_2 of all disjunctive rules in P^+. We know that P_1 is closed under here-intersection, i.e., $(X \cap Z, Y) \in M_s(P_1)$. Moreover, by Proposition 6, for all $r \in P_2$, we have that $X \models H(r)$ and $Z \models H(r)$, since each body $B(r)$ is empty. Fix such an r (observe that $P_2 \neq \emptyset$ by hypothesis and moreover $P_2 \subseteq P$ by definition). By Lemma 6, we get

$$X \cap H(\overline{r}) = Z \cap H(\overline{r}) = Y \cap H(\overline{r}) = \{\overline{p}\},$$

$X \cap \overline{\mathcal{A}} \subseteq \overline{(X \cap \mathcal{A})}$, and $Z \cap \overline{\mathcal{A}} \subseteq \overline{(Z \cap \mathcal{A})}$. By $p \in H(r)$, we obtain $p \in X \cap Z$, and thus $X \cap Z \models H(r)$. Consequently, $(X \cap Z, Y) \in M_s(r)$. Since this holds for all $r \in P_2$, we end up with $(X \cap Z, Y) \in M_s(P_2)$. We already know that $(X \cap Z, Y) \in M_s(P_1)$, and so $(X \cap Z, Y) \in M_s(P^+)$. This, however, is a contradiction to $(X \cap Z, Y) \notin M_s(P^+)$. Hence, P^+ must be closed under here-intersection. □

Lemma 8 *Let P be a HDFP over $\mathcal{A}$. Then,*

1. *if $I \in \mathcal{SM}(P)$, then there exists a $\overline{K} \subseteq \overline{\mathcal{A}}$ such that $I \cup \overline{K}$ is a stable model of P^+; and*
2. *if $I \in \mathcal{SM}(P^+)$, then $I \cap \mathcal{A}$ is a stable model of P.*

Proof. To show Part 1, let $I \in \mathcal{SM}(P)$ and let $\overline{K}$ be constructed by any $K \subseteq I$ satisfying $|K \cap H(r)| = 1$, for each disjunctive fact $r \in P$. Such $\overline{K}$ exists since P is an HDFP. Clearly, $I \cup \overline{K} \models P$ since no atoms from $\overline{\mathcal{A}}$ occur in P and $I \models P$. Moreover,

$$I \cup \overline{K} \models \{p \leftarrow \overline{p} \mid p \in \mathcal{A}\}$$

since $K \subseteq I$ by definition. Let R be the collection of the rules $(\overline{r})^{\rightarrow}$ and $\{\leftarrow \overline{p}, \overline{q} \mid p, q \in H(r), p \neq q\}$, for each

[3] The *exact hitting set* problem is as follows. Given a collection C of subsets of a set S, decide whether there exists a subset $S' \subseteq S$ such that $|S' \cap C'| = 1$, for each $C' \in C$. This problem is known to be NP-complete (Karp 1972).

disjunctive fact $r \in P$. Then, $I \cup \overline{K} \models R$, by the assumption that K satisfies $|K \cap H(r)| = 1$, for each $r \in P$ with $|H(r)| > 1$. Hence $I \cup \overline{K} \models P^+$.

It remains to show that no proper subset J of $I \cup \overline{K}$ is a model of $(P^+)^I$. Suppose $(J \cap \mathcal{A}) \subset (I \cap \mathcal{A})$. Then $J \models (P^+)^I$ would imply that $J \models P^I$, contradicting $I \in \mathcal{SM}(P)$. Now suppose $(J \cap \overline{\mathcal{A}}) \subset (I \cap \overline{\mathcal{A}})$. Then, $J \not\models R^I$, since we would have at least one rule from $(\overline{r})^{\rightarrow}$ which is not satisfied by J.

Concerning Part 2, assume $I \in \mathcal{SM}(P^+)$. Since $P \subseteq P^+$ and no atoms from $\mathcal{A}$ occur in P, we have $I \cap \mathcal{A} \models P$. Again, it remains to show that no $J \subset (I \cap \mathcal{A})$ exists, such that $J \models P^I$ holds. Towards a contradiction, let $J \subset (I \cap \mathcal{A})$ be a model of P^I. We show that then there exists a $\overline{K} \subseteq \overline{\mathcal{A}}$ such that both $(J \cup \overline{K}) \subset I$ and $J \cup \overline{K} \models (P^+)^I$ holds, thus contradicting $I \in \mathcal{SM}(P^+)$. First, since $I \in \mathcal{SM}(P^+)$, we have $(I, I) \in M_s^{\mathcal{A}^*}(P^+)$, and from Lemma 6 we get $I \cap \overline{\mathcal{A}} \subseteq \overline{(I \cap \mathcal{A})}$. Now let $\overline{K} = \overline{J} \cap (I \cap \overline{\mathcal{A}})$. Consequently, $(J \cup \overline{K}) \subset I$, and $J \cup \overline{K} \models \{p \leftarrow \overline{p} \mid p \in \mathcal{A}\}^I$. Moreover, for each disjunctive fact $r \in P$, we have $|J \cap H(r)| \geq 1$ (otherwise $J \not\models P^I$) as well as $|I \cap H(\overline{r})| = 1$ (otherwise $I \not\models P^+$). Since $J \subset I$ and $I \cap \overline{\mathcal{A}} \subseteq \overline{(I \cap \mathcal{A})}$ holds by $I \in \mathcal{SM}(P^+)$, we finally get $J \cup \overline{K} \models R^I$. Hence, we derive $J \cup \overline{K} \models (P^+)^I$. □

The relations from Lemmas 7 and 8 guarantee a faithful reduction (from arbitrary HDFPs to DLPs closed under here-intersection) of the relevant reasoning tasks in the context of logic programming. As already mentioned, the hardness results by Eiter & Gottlob (1995) carry over for HDFPs. We thus obtain our next result which shows that DLPs closed under here-intersection possess the same worst-case complexity as general DLPs.

Theorem 7 *Both the consistency problem and brave reasoning for DLPs closed under here-intersection is Σ_2^P-complete, and cautious reasoning for DLPs closed under here-intersection is Π_2^P-complete.*

Succinctness of DLPs

Finally, we discuss the size of the rewriting of a given DLP P into an equivalent NLP Q (if it exists).

Theorem 8 *There is no rewriting $f : \mathcal{DLP} \rightarrow \mathcal{NLP}$ such that (i) $P \equiv_\alpha f(P)$, and (ii) $f(P)$ is polynomial in the size of P, for every $P \in \mathcal{DLP}$, with $\alpha \in \{u, s\}$, unless the PH collapses.*

Proof. Assume that a polynomial-size rewriting f of the described kind exists. Consider the Π_2^P-hard problem of checking whether, for a given positive DLP P and a given atom a, $not\ a$ is a cautious consequence of P, i.e., whether a is not contained in any stable model of P (Eiter & Gottlob 1995).

Define $P_1 = P^+$ if $\alpha = s$, and $P_1 = P$ if $\alpha = u$. Then, $not\ A$ is a cautious consequence of P iff it is a cautious consequence of P_1. By the existence of f, we can guess an NLP P' in nondeterministic polynomial time such that $P' \equiv_\alpha P_1$ ($\alpha \in \{u, s\}$). Checking $P' \equiv_\alpha P_1$ is in coNP (Eiter & Fink 2003), and checking whether $not\ a$ is a cautious consequence of P' is in coNP (since P' is normal). Thus, the Π_2^P-hard problem of deciding whether $not\ a$ is a cautious consequence of P is in Σ_2^P, which is a contradiction unless the PH collapses. □

Also for rewritings under ordinary equivalence we cannot avoid an exponential blowup unless the PH collapses, as shown from results by Cadoli *et al.* (2000a; 2000b).

Proposition 9 *There exists no polynomial-size rewriting $f : \mathcal{DLP} \rightarrow \mathcal{NLP}$ such that $P \equiv f(P)$, for every $P \in \mathcal{DLP}$, unless the PH collapses.*

Clearly Theorem 8 is implied by Proposition 9, but the proof of the latter refers to non-uniform complexity classes, while ours is from first principles. In particular, a direct proof of Proposition 9 would show that a polynomial-size rewriting $f : \mathcal{DLP} \rightarrow \mathcal{NLP}$ such that $P \equiv f(P)$ implies coNP $\subseteq$ P/poly (P/poly is the class of problems decidable in polynomial time with polynomial advice), which in turn implies a collapse of the PH. Furthermore, Proposition 9 remains true for generalized rewritings f which admit projective extra variables, i.e., $P \equiv f(P)|_{\mathcal{A}}$, where $f(P)$ is defined over atoms $\mathcal{A}' \supseteq \mathcal{A}$ and $f(P)|_{\mathcal{A}}$ denotes the restriction of the stable models of $f(P)$ to the original atoms $\mathcal{A}$. This is a consequence of combining results by Cadoli *et al.* (2000a) and the facts that (i) model checking for NLPs is polynomial and (ii) model checking for circumscription (which is hard for the non-uniform compilability variant of coNP) is a special case of model checking for DLPs (both rely on minimal model checking).

We remark that, in terms of (Gogic *et al.* 1995), DLPs are, because of the exponential blow up, a stronger KR formalism than NLPs, unless the PH collapses, regardless of the notion of equivalence considered.

Conclusion

In this paper, we derived new results concerning the elimination of disjunctions in logic programs under the stable model semantics with respect to strong, uniform, and ordinary equivalence. We showed that under uniform and ordinary equivalence, disjunctions can always be eliminated, whereas for strong equivalence, this is precisely possible in case a certain semantic criterion is satisfied, viz. that the given program is closed under here-intersection. We also provide an explicit, uniform method to rewrite a given DLP into an equivalent NLP (if such an NLP exists, in case of strong equivalence). Although the resultant NLPs are in general exponentially larger than the input DLPs, we showed that this increase is in some sense unavoidable, thus providing further insight on the *succinctness* of DLPs.

Our ongoing and future work concerns a closer investigation of the newly derived class of DLPs closed under here-intersection, as well as extending our results to the function-free first-order (datalog) case. Furthermore, it remains to explore how our results can be applied for optimizations of algorithms used in disjunctive logic programming engines such as DLV and GnT.

References

Ben-Eliyahu, R., and Dechter, R. 1994. Propositional Semantics for Disjunctive Logic Programs. *Annals of Mathematics and Artificial Intelligence* 12:53–87.

Cadoli, M.; Donini, F.; Liberatore, P.; and Schaerf, M. 2000a. Space Efficiency of Propositional Knowledge Representation Formalisms. *Journal of Artificial Intelligence Research* 13:1–31.

Cadoli, M.; Donini, F.; Schaerf, M.; and Silvestri, R. 2000b. On Compact Representations of Propositional Circumscription. *Theoretical Computer Science* 182(1-2):183–202.

de Jongh, D. J., and Hendriks, L. 2003. Characterizations of Strongly Equivalent Logic Programs in Intermediate Logics. *Theory and Practice of Logic Programming* 3(3):259–270.

Dix, J.; Gottlob, G.; and Marek, W. 1996. Reducing Disjunctive to Non-Disjunctive Semantics by Shift-Operations. *Fundamenta Informaticae* 28:87–100.

Eiter, T., and Fink, M. 2003. Uniform Equivalence of Logic Programs under the Stable Model Semantics. Technical Report INFSYS RR-1843-03-08, Institut für Informationssysteme, Technische Universität Wien, A-1040 Vienna, Austria. Preliminary Report. Short version in *Proc. ICLP-03*.

Eiter, T., and Gottlob, G. 1995. On the Computational Cost of Disjunctive Logic Programming: Propositional Case. *Annals of Mathematics and Artificial Intelligence* 15(3/4):289–323.

Eiter, T., and Gottlob, G. 1997. Expressiveness of Stable Model Semantics for Disjunctive Logic Programs with Functions. *Journal of Logic Programming* 33(2):167–178.

Eiter, T.; Faber, W.; Leone, N.; and Pfeifer, G. 2000. Declarative Problem-Solving Using the DLV System. In Minker, J., ed., *Logic-Based Artificial Intelligence*, 79–103. Kluwer Academic Publishers.

Eiter, T.; Fink, M.; Tompits, H.; and Woltran, S. 2004. Simplifying Logic Programs under Uniform and Strong Equivalence. In *Proc. LPNMR 2004*, number 2923 in LNCS, 87–99. Springer.

Eiter, T.; Gottlob, G.; and Mannila, H. 1997. Disjunctive Datalog. *ACM Transactions on Database Systems* 22(3):364–417.

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

Gelfond, M.; Przymusinska, H.; Lifschitz, V.; and Truszczyński, M. 1991. Disjunctive Defaults. In *Proc. KR-91*, 230–237.

Gogic, G.; Kautz, H.; Papadimitriou, C. H.; and Selman, B. 1995. The Comparative Linguistics of Knowledge Representation. In *Proc. IJCAI-95*, 862–869. Morgan Kaufmann.

Janhunen, T.; Niemelä, I.; Simons, P.; and You, J.-H. 2000. Partiality and Disjunctions in Stable Model Semantics. In *Proc. KR-00*, 411–419. Morgan Kaufmann.

Karp, R. 1972. Reducibility among Combinatorial Problems. In Miller, R. E.; and J. W. Thatcher, J. W., eds., *Complexity Of Computer Computations*, 85–103. Plenum Press.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly Equivalent Logic Programs. *ACM Trans. on Computational Logic* 2(4):526–541.

Lin, F., and Zhao, Y. 2002. ASSAT: Computing Answer Sets of a Logic Program by SAT Solvers. In *Proc. AAAI-02*. 112–117. AAAI Press / MIT Press.

Lin, F. 2002. Reducing Strong Equivalence of Logic Programs to Entailment in Classical Propositional Logic. In *Proc. KR-02*, 170–176. Morgan Kaufmann.

Maher, M. J. 1988. Equivalences of Logic Programs. In Minker (1988). 627–658.

Marek, W., and Remmel, J. 2003. On the Expressibility of Stable Logic Programming. *Theory and Practice of Logic Programming* 3(4-5):551–567.

Marek, V. W.; Treur, J.; and Truszczyński, M. 1997. Representation Theory for Default Logic. *Annals of Mathematics and Artificial Intelligence* 21(2–4):343–358.

Minker, J., ed. 1988. *Foundations of Deductive Databases and Logic Programming*. Washington DC: Morgan Kaufman.

Osorio, M.; Navarro, J.; and Arrazola, J. 2001. Equivalence in Answer Set Programming. In *Proc. LOPSTR-01*, number 2372 in LNCS, 57–75. Springer.

Pearce, D., and Valverde, A. 2003. Some Types of Equivalence for Logic Programs and Equilibrium Logic. In *Proc. APPIA-GULP-PRODE-03*, Informal Proceedings.

Pearce, D.; Tompits, H.; and Woltran, S. 2001. Encodings for Equilibrium Logic and Logic Programs with Nested Expressions. In *Proc. EPIA-01*, volume 2258 of LNCS, 306–320. Springer.

Pearce, D. 1997. A New Logical Characterisation of Stable Models and Answer Sets. In *Non-Monotonic Extensions of Logic Programming* (*NEMLP 1996*), volume 1216 of LNCS, 57–70. Springer.

Przymusinski, T. 1991. Stable Semantics for Disjunctive Programs. *New Generation Computing* 9:401–424.

Sagiv, Y. 1988. Optimizing Datalog Programs. In Minker (1988). 659–698.

Schlipf, J. 1995. The Expressive Powers of Logic Programming Semantics. *Journal of Computer and System Sciences* 51(1):64–86. Abstract in *Proc. PODS-90*, pp. 196–204.

Simons, P.; Niemelä, I.; and Soininen, T. 2002. Extending and Implementing the Stable Model Semantics. *Artificial Intelligence* 138:181–234.

Turner, H. 2001. Strong Equivalence for Logic Programs and Default Theories (Made Easy). In *Proc. LPNMR-01*, number 2173 in LNCS, 81–92. Springer.

Turner, H. 2003. Strong Equivalence Made Easy: Nested Expressions and Weight Constraints. *Theory and Practice of Logic Programming* 3(4–5):609–622.