INFSYS
RESEARCH
REPORT

INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG WISSENSBASIERTE SYSTEME

ON ELIMINATING DISJUNCTIONS IN STABLE LOGIC PROGRAMMING

Thomas Eiter   Michael Fink   Hans Tompits
Stefan Woltran

INFSYS RESEARCH REPORT 1843-03-12
NOVEMBER 2003

Institut für Informationssysteme
Abtg. Wissensbasierte Systeme
Technische Universität Wien
Favoritenstraße 9-11
A-1040 Wien, Austria
Tel: +43-1-58801-18405
Fax: +43-1-58801-18493
sek@kr.tuwien.ac.at
www.kr.tuwien.ac.at

TECHNISCHE UNIVERSITÄT WIEN

INFSYS RESEARCH REPORT
INFSYS RESEARCH REPORT 1843-03-12, NOVEMBER 2003

ON ELIMINATING DISJUNCTIONS IN STABLE LOGIC PROGRAMMING

Thomas Eiter[1] and Michael Fink[2] and Hans Tompits[3] and Stefan Woltran[4]

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 does there exist an equivalent normal (i.e., disjunction-free) logic program P'. 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.

Keywords: uniform equivalence, strong equivalence, stable models, answer set semantics, computational complexity, program optimization

[1]Institut of Information Systems, Knowledge Bases Systems Group, TU Vienna, Favoritenstraße 9-11, A-1040 Vienna, Austria. E-mail: eiter@kr.tuwien.ac.at.
[2]Institut of Information Systems, Knowledge Bases Systems Group, TU Vienna, Favoritenstraße 9-11, A-1040 Vienna, Austria. E-mail: michael@kr.tuwien.ac.at.
[3]Institut of Information Systems, Knowledge Bases Systems Group, TU Vienna, Favoritenstraße 9-11, A-1040 Vienna, Austria. E-mail: tompits@kr.tuwien.ac.at.
[4]Institut of Information Systems, Knowledge Bases Systems Group, TU Vienna, Favoritenstraße 9-11, A-1040 Vienna, Austria. E-mail: stefan@kr.tuwien.ac.at.

Acknowledgements: This work was partially supported by the Austrian Science Fund (FWF) under projects Z29-N04 and P15068-INF, as well as by the European Commission under projects FET-2001-37004 WASP and IST-2001-33570 INFOMIX.

Contents

1 Introduction **1**

2 Preliminaries **2**

3 Equivalence Results **4**
3.1 Strong Equivalence 4
3.2 Uniform Equivalence 8
3.3 Ordinary Equivalence 10

4 Complexity Issues **11**
4.1 Checking Here-Intersection 11
4.2 Expressibility of DLPs closed under Here-Intersection 13
4.3 Succinctness of DLPs 15

5 Conclusion **16**

1 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 [10], while normal logic programs (NLPs) can express only properties at the first level of the polynomial hierarchy [28]. However, disjunction is often not really needed, in that an equivalent normal logic program (i.e., without disjunction) can be given. For example, [9] 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 [5], Smodels [29], ASSAT [17], or GnT [13], 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 [20]; 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 [2, 12, 19].

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* [15, 30, 31, 24, 16, 4] and *uniform equivalence* [6, 7, 25] 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 [31, 22, 7], 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 helpful for optimizing components of a more complex logic program. Note that, as recently discussed in [25], uniform and strong equivalence are essentially the only concepts of equivalence obtained by varying the logical form of the program extensions.

A natural issue in this context is the expressiveness of disjunctions in rule heads, i.e., whether it really adds 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. [31]). 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.

(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 a NLP must lead in general to an exponential blowup, 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 recent results on simplifying logic programs under different notions of equivalence [22, 31, 7]. They might be used for deciding whether a given disjunctive problem representation for a system such as DLV [5] or GnT [13] 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 [3, 2, 12], showing that DLPs are more succinct than NLPs under different notions of equivalence.

2 Preliminaries

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

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

$n \geq m \geq l \geq 0$, where all a_i are atoms from a finite set of propositional atoms, $\mathcal{A}$, 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 $\{not\, a_1, \ldots, not\, a_n\}$.

A rule r is *normal*, if $l \leq 1$; and *(proper) disjunctive*, otherwise; *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 $\perp$ otherwise.

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 [11, 26]. 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 $\{r\}^I$. (Note that an empty program has any interpretation as its model.) 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 $\mathcal{SM}(P)$.

Lemma 2.1 *Let P be a DLP and $X \subseteq Y' \subseteq Y$. Then, $X \models P^{Y'}$ implies $X \models P^Y$.*

The result is seen by the observation that $Y' \subseteq Y$ implies $P^Y \subseteq P^{Y'}$. Thus, $X \models P^{Y'}$ implies $X \models P^Y$. In particular, for $X = Y'$, $X \models P$ implies $X \models P^Y$, for any $X \subseteq Y$.

Several notions of equivalence between logic programs have been considered in the literature (cf., e.g., [15, 18, 27]). 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,*

(*i*) *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., $P \cup R \equiv Q \cup R$;*

(*ii*) *P and Q are* uniformly equivalent, *or u-*equivalent, *denoted $P \equiv_u Q$, iff, for any set F of non-disjunctive 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 not vice versa. Both notions of equivalence enjoy interesting semantical characterizations [15, 30, 31]. As shown in [15], strong equivalence is closely related to the non-classical logic of here-and-there, which was adapted to logic-programming terms by Turner [30, 31]:

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 2.1 ([30, 31]) *For any DLP P and Q, $P \equiv_s Q$ iff $M_s(P) = M_s(Q)$.*

SE-models also can be used to determine the stable models of a program.

Proposition 2.2 ([23, 15]) *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 [6].

Definition 3 *Let P be a DLP and (X, Y) an SE-model of P. Then, (X, Y) is an* 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 2.3 ([6]) *For any DLP P and Q, $P \equiv_u Q$ iff $M_u(P) = M_u(Q)$.*

This test can be reformulated as follows.

Proposition 2.4 *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 2.3, $P \equiv_u Q$ 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 if direction, 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 $(X, Y) \notin M_u(P)$. For (i), we have two cases, by definition of UE-models. 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 $X \subset X' \subset Y$, 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. The argumentation for (ii) is analogous. □

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

Proposition 2.5 *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 $Y \models H(r)$, $Y \not\models B^+(r)$, or $Y \cap B^-(r) \neq \emptyset$. The latter holds iff $X \models H(r)$, $X \not\models B^+(r)$, or $Y \cap B^-(r) \neq \emptyset$. Hence, $(X, Y) \in M_s(r)$ iff $Y \cap B^-(r) \neq \emptyset$ or

$$\Big(Y \models H(r) \ \vee \ Y \not\models B^+(r)\Big) \wedge \Big(X \models H(r) \ \vee \ X \not\models B^+(r)\Big). \tag{1}$$

Clearly, $Y \not\models B^+(r)$ implies $X \not\models B^+(r)$ and, furthermore, $X \models H(r)$ implies $Y \models H(r)$. From this, it is easily verified that (X, Y) satisfies (1) iff either $Y \not\models B^+(r)$, $X \models H(r)$ (i.e., (i)), or jointly $X \not\models B^+(r)$ and $Y \models H(r)$ (i.e., (iii)), 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. □

3 Equivalence Results

In what follows, we provide general characterizations under which disjunctions can be eliminated 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.

3.1 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:

$$\begin{array}{ll}
P_1 = \{a \vee b \leftarrow\}; & \\
P_2 = \{a \vee b \leftarrow; a \leftarrow\}; & P_3 = \{a \vee b \leftarrow; a \leftarrow b\}; \\
P_4 = \{a \vee b \leftarrow; a \leftarrow; \leftarrow \mathit{not}\, b\}; & P_5 = \{a \vee b \leftarrow; a \leftarrow b; \leftarrow \mathit{not}\, b\}; \\
P_6 = \{a \vee b \leftarrow; a \leftarrow; b \leftarrow\}; & P_7 = \{a \vee b \leftarrow; a \leftarrow b; b \leftarrow a\}; \\
P_8 = \{a \vee b \leftarrow; \leftarrow a, b\}; & P_9 = \{a \vee b \leftarrow; \leftarrow \mathit{not}\, a; \leftarrow \mathit{not}\, b\}.
\end{array}$$

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

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

[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 $a \leftarrow \mathit{not}\, b$ and $b \leftarrow \mathit{not}\, a$, i.e., by applying the usual shifting technique. 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 shifting rules.

Proposition 3.1 *For a disjunctive rule, r, define*

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

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

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

Clearly, we have $S_r \subseteq M_s(r^{\rightarrow})$ and $M_s(r) \subseteq M_s(r^{\rightarrow})$. The former relation can be seen by as follows, for each $r_p \in r^{\rightarrow}$: If $(X,Y) \in S_r$, then $|Y \cap H(r)| > 1$ and thus, by definition, $Y \not\models B^-(r_p)$, yielding $(X,Y) \in M_s(r^{\rightarrow})$. For the latter relation, $M_s(r) \subseteq M_s(r^{\rightarrow})$, suppose $(X,Y) \in M_s(r)$. 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 an $p \in Y$ 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 $M_s(r^{\rightarrow}) \subseteq M_s(r) \cup S_r$. Therefore, let $(X,Y) \in M_s(r^{\rightarrow})$ and suppose $(X,Y) \notin M_s(r)$. We show that $(X,Y) \in S_r$. Towards a contradiction, suppose $(X,Y) \notin S_r$. Since $(X,Y) \notin M_s(r)$, we get by Proposition 2.5 that $X \not\models H(r)$, $Y \models B(r)$, and either $X \models B^+(r)$ or $Y \not\models H(r)$.

We have two cases. First, if $Y \not\models H(r)$, i.e., $|Y \cap H(r)| = 0$, this clearly contradicts $(X,Y) \in M_s(r^{\rightarrow})$. 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 $p \notin X$, and thus contradicts $(X,Y) \in M_s(r^{\rightarrow})$. □

Observe that $M_s(r)$ and S_r are disjoint, in view of Proposition 2.5. Therefore, $S_r \cap M_s(P \setminus \{r\})$ are exactly the potential additional SE-models which may occur after the shifting process. In the above examples, we have $r = a \vee b \leftarrow$, $r^{\rightarrow} = \{a \leftarrow \mathit{not}\, b;\ b \leftarrow \mathit{not}\, a\}$, 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 $(\emptyset, ab) \notin M_s(a \leftarrow)$, 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 property 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 3.1 *Each normal LP 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: $X \models P'$ and $X' \models P'$ implies $X \cap X' \models P'$. □

Hence, we can state the following result:

Theorem 3.1 *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,*

$$P_{r,s} = P_r^- \cup r^{\rightarrow} \cup \hat{r}_{P_s}, \quad \textit{where} \quad \hat{r}_{P_s} = \bigcup_{(X,Y,Z) \in S_r^{\uparrow}(P)} r_{X,Y,Z},$$

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

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 accordingly to the elements of $S_r^{\uparrow}(P)$ behave as follows:

Proposition 3.2 *For any sets, $X,Y,Z \subseteq \mathcal{A}$, $(X',Y') \in INT_{\mathcal{A}}$ is 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. Let $p \in Y$ and r_p the corresponding rule in $r_{X,Y,Z}$ with $H(r_p) = p$. By Proposition 2.5, $(X',Y') \in M_s(r_p)$ iff one of the following conditions hold: (i) $X' \models H(r_p)$ (i.e., $p \in X'$); (ii) $Y' \not\models B(r_p)$ (i.e., either $X \not\subseteq Y'$ or $Y' \not\subseteq Z$); or (iii) $X' \not\models B^+(r_p)$ and $Y' \models H(r_p)$ (i.e., $X \not\subseteq X'$ and $p \in Y'$). For any two rules $r_p, r_q \in r_{X,Y,Z}$, we have $B(r_p) = B(r_q)$. Thus, $(X',Y') \in M_s(r_{X,Y,Z})$ iff either (ii), or, for any $r_p \in r_{X,Y,Z}$, (i) or (iii) holds. For the latter relations consider two cases. If $X \subseteq X'$, the relation holds iff $p \in X'$ for each $r_p \in r_{X,Y,Z}$, i.e., for each $p \in Y$. Hence, the relation holds iff $Y \subseteq X'$. If $X \not\subseteq X'$ and $Y \not\subseteq X'$ then from (iii) $p \in Y'$ for each $p \in Y$. Hence, $Y \subseteq Y'$ has then to hold. □

[2] The term "here-intersection" derives from the logical underpinning of strong equivalence [15], given by the logic of here-and-there [23], in which the first component of an SE-interpretation refers to the world "here".

Note that $r_{X,Y,Z}$ may contain redundant rules, for instance 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 3.2 *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 3.1, we have

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

The strategy for the remainder of the proof is as follows. We first show that for the latter set

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

This leads us then to $M_s(P_{r,s}) = M_s(P) \cap M_s(\hat{r}_{P_s})$, and thus to show $M_s(P) = M_s(P) \cap M_s(\hat{r}_{P_s})$, 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 immediately have $(X,Z) \notin T$. Otherwise, $(X,Z) \in M_s(P_r^-)$. Then there exists a triple $(X,Y,Z) \in S_r^{\uparrow}(P)$ with $X \subset Y \subseteq Z$, and thus, $r_{X,Y,Z} \subseteq \hat{r}_{P_s}$. We now show that $(X,Z) \notin M_s(r_{X,Y,Z})$. Assume the contrary, i.e., $(X,Z) \in M_s(r_{X,Y,Z})$. Then, by Proposition 3.2 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 $Y \subseteq Z$. (i) does not hold since we have $X \subset Y$; (ii) does not hold since we get $X \subset Z$ from $X \subset Y \subseteq Z$; (iii+iv) do not hold trivially. Contradiction.

We 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 $\hat{r}_{P_s}$. Hence, suppose $\hat{r}_{P_s} \neq \emptyset$. We show that, for each $r_{X,Y,Z} \subseteq \hat{r}_{P_s}$ and each $(X',Z') \in M_s(P)$, $(X',Z') \in M_s(r_{X,Y,Z})$ holds. Towards a contradiction, let $r_{X,Y,Z} \subseteq \hat{r}_{P_s}$, $(X',Z') \in M_s(P)$, and suppose $(X',Z') \notin M_s(r_{X,Y,Z})$. 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 3.2, (X',Z') satisfies the following conditions for being a counter SE-model of $r_{X,Y,Z}$: (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 Lemma 2.1, we get $X' \models P^Z$ and $Z' \models P^Z$, and from (b) we get $(Y,Z) \in M_s(P)$, and thus $Z \models P$. Hence, $(X',Z) \in M_s(P)$ 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.

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

$X \not\subseteq X'$: We have a similar argumentation. By (4), $Y \not\subseteq Z'$ holds, yielding $(Y \cap Z') \subset Y$. Moreover, by (2) $X \subseteq Z'$ and by (c) $X \subset Y$ hold. Thus, $X \subseteq (Y \cap Z')$. Again, we have $X \subseteq (Y \cap Z') \subset Y$, and by (a) and $(Y \cap Z',Z) \in M_s(P)$, this contradicts $(X,Y,Z) \in S_r^{\uparrow}(P)$. $\square$

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 3.2 *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 3.1, 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 and $r = a \vee b \leftarrow$. 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. In the end, our general transformation goes well conform with our informal analysis in the beginning of the section.

3.2 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 with $r = a \vee b \leftarrow$ replaced 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 $r_{X,Y,Z}$ as in Definition 5, and define*

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

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

Lemma 3.3 *Given a DLP P with $r \in P$ disjunctive, it holds that $M_u(P) = M_u(P_{r,u})$.*

Proof. To start with, observe that analogously to the proof of Lemma 3.2, we get

$$M_s(P_{r,u}) = \Big(M_s(P) \cap M_s(\hat{r}_{P_u})\Big) \cup \Big(M_s(P_r^-) \cap S_r \cap M_s(\hat{r}_{P_u})\Big). \tag{2}$$

We show $M_u(P) = M_u(P_{r,u})$. By Proposition 2.4, 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 $M_u(P) \subseteq M_s(P_{r,u})$ immediately. 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 that, for each $r_{X,Y,Y} \subseteq \hat{r}_{P_u}$ and each $(X', Y') \in M_s(P)$, $(X', Y') \in M_s(r_{X,Y,Y})$ holds. Towards a contradiction, let $r_{X,Y,Y} \subseteq \hat{r}_{P_u}$, $(X', Y') \in M_s(P)$, and suppose $(X', Y') \notin M_s(r_{X,Y,Y})$. 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 $Z = Y$. On the other hand, (X', Y') satisfies the following conditions for being a counter SE-model of $r_{X,Y,Y}$, by Proposition 3.2: (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) $Y' \subseteq Y$ hold. We get $X \subseteq X' \subset Y$. Moreover, $X' \models P^{Y'}$ holds, since $(X', Y') \in M_s(P)$. By Lemma 2.1, 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), then $Y \not\subseteq Y'$. Together with (iii) we have $Y' \subset Y$. Moreover, $X \subseteq Y'$ holds by (ii). Since $(X', Y') \in M_s(P)$, $Y' \models P^{Y'}$ holds. Lemma 2.1 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 contradiction to (c).

It remains to show $M_u(P_{r,u}) \subseteq M_s(P)$. 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 (2), it can easily 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 shows 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, for any $(X, Y) \in S_r \cap M_s(P_r^-)$, that either $(X, Y) \notin M_u(P_{r,u})$ or $(X, Y) \notin M_s(\hat{r}_{P_u})$. So, fix some $(X, Y) \in S_r \cap M_s(P_r^-)$. 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 $M_s(P) \subseteq M_s(P_{r,u})$, 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 from Proposition 3.2 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+ii) this is seen by the fact that $(X, Y) \in S_r$ and thus $X \subset Y$. (iii+iv) trivially fail. Hence, $(X, Y) \notin M_s(\hat{r}_{P_u})$. □

Theorem 3.3 *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 has $\hat{r}_{P_u} \neq \emptyset$, 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:

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

Indeed, P is not closed under here-intersection. S_r is given by $\{(\emptyset, ab), (\emptyset, abc), (c, abc)\}$ and as can be verified, $S_r \cap M_s(P_r^-) = S_r$. Moreover,

$$S_r^{\uparrow}(P) = \{(\emptyset, a, ab), (\emptyset, b, ab), (\emptyset, a, abc), (\emptyset, b, abc), (c, abc, abc)\}.$$

Only the last triple, (c, abc, abc), is applied in the construction of $\hat{r}_{P_u}$. In fact, we have to add $r_{c,abc,abc}$ (which is given by $\{a \leftarrow c; b \leftarrow c; c \leftarrow c\}$) to $P_r^- \cup r^{\rightarrow}$. For the resulting program P_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 UE-model of $P_{r,u}$.

3.3 Ordinary Equivalence

Finally, we discuss the case of ordinary equivalence. Since uniform equivalence implies ordinary equivalence, in view of Theorem 3.3, 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:

Corollary 3.1 *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 3.4 *Let P be 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 start with, we state the following result which is easily seen from Proposition 3.2.

$$\text{For any } Y \subseteq \mathcal{A}, \quad M_s(r_{\emptyset,Y,Y}) = \{(X',Y') \in INT_{\mathcal{A}} \mid Y \subseteq X' \text{ or } Y' \not\subseteq Y\}. \tag{3}$$

Let $Y \in \mathcal{SM}(P)$. Then, for each $Y' \in \mathcal{SM}(P)$, either $Y \not\subseteq Y'$ or $Y = Y'$. By (3), $(Y,Y) \in M_s(\tilde{P})$. Towards a contradiction, suppose $Y \notin \mathcal{SM}(\tilde{P})$. By Proposition 2.2, 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 (3), it is easily seen that this is contradicted by $X \subset Y$. Thus, $Y \in \mathcal{SM}(\tilde{P})$.

Conversely, suppose $Y \in \mathcal{SM}(\tilde{P})$ and $Y \notin \mathcal{SM}(P)$. That is, $(X,Y) \in M_s(P)$ for some $X \subset Y$. Then, $Y \not\subseteq Y'$ for each $Y' \in \mathcal{SM}(P)$. We have $(X,Y) \in M_s(r_{\emptyset,Y',Y'})$, for each $Y' \in \mathcal{SM}(P)$ by (3). Thus, $(X,Y) \in M_s(\tilde{P})$ with $X \subset Y$. But this contradicts $Y \in \mathcal{SM}(\tilde{P})$. □

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. for ordinary equivalence.

Lemma 3.4 *Let P, r, P_r^-, $S_r^{\uparrow}(P)$, and $r_{X,Y,Z}$ be as in Definition 5, and define*

$$P_{r,e} = P_r^- \cup r^{\rightarrow} \cup \hat{r}_{P_e}, \quad \textit{where} \quad \hat{r}_{P_e} = \bigcup_{(X,Z,Z) \in S_r^{\uparrow}(P), Z \in \mathcal{SM}(P)} r_{X,Z,Z}.$$

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

Proof. We first show that,

$$\text{for each } Y \subseteq \mathcal{A}, \quad Y \models P \quad \text{iff} \quad Y \models P_{r,e}. \tag{4}$$

From Lemma 3.3, we have $M_u(P) = M_u(P_{r,u})$ and thus $Y \models P$ iff $Y \models P_{r,u}$. By definition, $P_{r,e} \subseteq P_{r,u}$, which shows the only-if direction. For 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, let $Y \in \mathcal{SM}(P)$. Then, $Y \models P$ and by (4), $Y \models P_{r,e}$. It remains to show that no $X \subset Y$, yields an SE-model (X,Y) of $P_{r,e}$. Towards a contradiction, suppose some $X \subset Y$ 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 3.1, $(X,Y) \in S_r$. We have $(X,Y) \in S_r \cap M_s(P_r^-)$ and $Y \in \mathcal{SM}(P)$, and thus get $r_{X,Y,Y} \in P_{r,e}$ by construction. By Proposition 3.2, $(X,Y) \in M_s(r_{X,Y,Y})$ only if $X = Y$. Thus $(X,Y) \notin M_s(P_{r,e})$. Contradiction.

Conversely, let $Y \in \mathcal{SM}(P_{r,e})$; by (4), we get $Y \models P$. Towards a contradiction, suppose $Y \notin \mathcal{SM}(P)$, i.e., there exists a set $X \subset Y$, such that $(X, Y) \in M_s(P)$. By Proposition 3.1, $(X, Y) \in M_s(P_r^- \cup r^\rightarrow)$. Since $Y \models P$, and thus $Y \models P^Y$, we get by Lemma 2.1 that $Y \models P^{Y'}$ holds, for each Y' with $Y \subseteq Y'$. Hence, $Y \not\subseteq Y'$, for each $Y' \in \mathcal{SM}(P)$. By Proposition 3.2, $(X, Y) \in M_s(r_{X,Y',Y'})$, for each Y' with $Y \not\subseteq Y'$. Hence, $(X, Y) \in M_s(P_{r,e})$. By Proposition 2.2, this contradicts $Y \in \mathcal{SM}(P_{r,e})$. □

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, whenever P is closed under here-intersection, a program $P_{r,s}$ which is strongly equivalent to P. All these programs are of the form

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

Recall that an application of these rule-wise transformations successively for all disjunctive rules in a given program leads to a normal logic program. Hence, our method can be seen as a uniform framework to obtain 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 disjunctions, since the outcome of these methods coincides with the presented 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 free programs, cf. [1]) which allows for shifting. Moreover, in terms of uniform equivalence, we generalized on an observation made in [6] (see Theorem 4.3).

However, the size of the outcoming programs 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.

4 Complexity Issues

This sections 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.

4.1 Checking Here-Intersection

We can express the test for a DLP P of being closed under here-intersection via the following normal logic program, which is linear in the size of P. Therefore, 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 and let $\bar{V}$, V'_1, $\bar{V}'_1$, V'_2, $\bar{V}'_2$, V'_3, and u be disjoint new atoms. Define*

$$
\begin{aligned}
P_Q \;&=\; \{v \leftarrow not\, \bar{v};\; \bar{v} \leftarrow not\, v \mid v \in V\} && (5)\\
&\cup\; \{v'_i \leftarrow v, not\, \bar{v}'_i;\; \bar{v}'_i \leftarrow not\, v'_i \mid v \in V, i \in \{1, 2\}\} && (6)\\
&\cup\; \{\leftarrow B(r), not\, H(r) \mid r \in P\} && (7)
\end{aligned}
$$

$$
\begin{aligned}
&\cup \quad \{\leftarrow B^{+}(r_i'), not\, B^{-}(r), not\, H(r_i') \mid r \in P, i \in \{1,2\}\} && (8)\\
&\cup \quad \{v_3' \leftarrow v_1', v_2' \mid v \in V\} && (9)\\
&\cup \quad \{u \leftarrow B^{+}(r_3'), not\, B^{-}(r), not\, H(r_3') \mid r \in P\} && (10)\\
&\cup \quad \{\leftarrow not\, u\}. && (11)
\end{aligned}
$$

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

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

Theorem 4.1 *A DLP P is closed under here-intersection iff $\mathcal{SM}(P_Q) = \emptyset$.*

Based on this, we derive the following complexity result.

Theorem 4.2 *Checking whether, for a given DLP P, there exists a normal program Q such that P and Q are strongly equivalent, is* coNP-*complete.*

Proof. By Theorem 4.1, and the linear encoding from Definition 7 we get that closedness under here-intersection is in coNP. As an alternative argumentation, we note that checking whether a given DLP P is not closed under here-intersection can obviously be done in nondeterministic polynomial time as follows: Guess SE-interpretations (X, Y) and (X', Y), and check in polynomial time whether $(X, Y) \in M_s(P)$, $(X', Y) \in M_s(P)$, and $(X \cap X', Y) \notin M_s(P)$.

We show coNP-hardness of this problem by a reduction to the coNP-complete problem of deciding whether $\mathcal{A}$ is the unique model of a positive DLP as follows:

Let P be a positive DLP over the alphabet $\mathcal{A}$, let v, v' be new atoms, and consider the program

$$Q = P \cup \{v \vee v' \leftarrow;\ v \leftarrow \mathcal{A};\ v' \leftarrow \mathcal{A}\}.$$

We prove that Q is closed under here-intersection iff $\mathcal{A}$ is the unique model of P.

First, if $\mathcal{A}$ is the unique model of P then, by construction, $\mathcal{A} \cup \{v, v'\}$ 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 Q is closed under here-intersection and, towards a proof by contradiction, suppose there exists a model $M \subset \mathcal{A}$ of P. Then both, $M \cup \{v\}$ and $M \cup \{v'\}$ are models of Q and thus also both, $(M \cup \{v\}, \mathcal{A} \cup \{v, v'\})$ and $(M \cup \{v'\}, \mathcal{A} \cup \{v, v'\})$ are SE-models of Q. However, $(M, \mathcal{A} \cup \{v, v'\})$ is not an SE-model of Q, since $M \not\models v \vee v' \leftarrow$. This contradicts the assumption that Q is closed under here-intersection, hence $\mathcal{A}$ is the unique model of P.

We have shown coNP-hardness of deciding whether a DLP P is closed under here-intersection which immediately implies coNP-hardness of checking whether there exists a normal logic program Q strongly equivalent to P by Theorem 3.2. □

4.2 Expressibility of DLPs closed under Here-Intersection

We now consider the expressibility 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 in [8], for DLPs 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.

Definition 8 *A DLP P is called*

(*i*) *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*

(*ii*) *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 (i) and (ii) 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 a in r by $\overline{a}$. 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 $(\overline{r})^{\rightarrow}$ and $\{\leftarrow \overline{p}, \overline{q} \mid p, q \in H(r), p \neq q\}$.*

Sublemma 4.1 *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:*

(*i*) *$(X, Y) \in M_s^{\mathcal{A}^*}(P)$, $X \cap \overline{\mathcal{A}} \subseteq \overline{(X \cap \mathcal{A})}$, $Y \cap \overline{\mathcal{A}} \subseteq \overline{(Y \cap \mathcal{A})}$; and*

(*ii*) *for each $r \in P$ with $|H(r)| > 1$, $X \cap H(\overline{r}) = Y \cap H(\overline{r}) = \{\overline{p}\}$.*

[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 [14].

Proof. It is easy to see that $M_s^{\mathcal{A}^*}(P \cup \{p \leftarrow \overline{p} \mid p \in \mathcal{A}\})$ is given by those $(X, Y) \in INT_{\mathcal{A}^*}$ which satisfy (i). Furthermore, consider a disjunctive fact $r \in P$. Applying Proposition 3.1, we get

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

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 (ii). □

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

$$P_1^+ = \{a \vee b \leftarrow;\ a \leftarrow \overline{a};\ b \leftarrow \overline{b};\ \overline{a} \leftarrow not\, \overline{b};\ \overline{b} \leftarrow not\, \overline{a};\ \leftarrow \overline{a}, \overline{b}\}.$$

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 4.1 *For each DFP P, P^+ is closed under here-intersection.*

Proof. If P is a NLP, then P^+ is clearly closed under here-intersection. So suppose that $P \in \mathcal{DLP} \backslash \mathcal{NLP}$, 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 2.5, 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 $P_2 \subseteq P$ by definition). By Sublemma 4.1, 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 $Z \cap \overline{\mathcal{A}} \subseteq \overline{(Z \cap \mathcal{A})}$. Since $p \in H(r)$, 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 4.2 *Let P be a HDFP over $\mathcal{A}$. Then,*

(*i*) *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*

(*ii*) *if $I \in \mathcal{SM}(P^+)$, then $I \cap \mathcal{A}$ is a stable model of P.*

Proof. To show (i), 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 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 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 (ii), 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 by Lemma 4.1, $I \cap \overline{\mathcal{A}} \subseteq \overline{(I \cap \mathcal{A})}$. Now let $\overline{K} = \overline{J} \cap (I \cap \overline{\mathcal{A}})$. Consequently, $(J \cup \overline{K}) \subset I$, and moreover, we have $J \cup \overline{K} \models \{p \leftarrow \overline{p} \mid p \in \mathcal{A}\}^I$. Moreover, for each disjunctive fact $r \in P$, we have $|J \cap H(r)| \geq 1$ (otherwise $J \not\models P^I$) and $|I \cap H(\overline{r})| = 1$ (otherwise $I \not\models P^+$). Since $J \subset I$ and $I \cap \overline{\mathcal{A}} \subseteq \overline{(I \cap \mathcal{A})}$ holds by the fact that $I \in \mathcal{SM}(P^+)$, we finally get $J \cup \overline{K} \models R^I$. Hence, $J \cup \overline{K} \models (P^+)^I$. □

The relations from Lemmas 4.1 and 4.2 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 in [8] 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 4.3 *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.*

4.3 Succinctness of DLPs

Theorem 4.4 *There is no polynomial-size rewriting $f : \mathcal{DLP} \rightarrow \mathcal{NLP}$ such that $P \equiv_\alpha f(P)$, for every $P \in \mathcal{DLP}$, with $\alpha \in \{u, s\}$, unless the PH collapses.*

Proof. Towards a contradiction, 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 every stable model of P [8].

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 [6], 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 easily shown from known results [3, 2].

Proposition 4.1 *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 4.4 is implied by Proposition 4.1, 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 4.1 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 4.1 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 in [2] 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 [12], 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.

5 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

[1] R. Ben-Eliyahu and R. Dechter. Propositional semantics for disjunctive logic programs. *Annals of Mathematics & Artificial Intelligence*, 12:53–87, 1994.

[2] M. Cadoli, F. Donini, P. Liberatore, and M. Schaerf. Space efficiency of propositional knowledge representation formalisms. *Journal of Artificial Intelligence Research*, 13:1–31, 2000.

[3] M. Cadoli, F. Donini, M. Schaerf, and R. Silvestri. On compact representations of propositional circumscription. *Theoretical Computer Science*, 182(1-2):183–202, 1997.

[4] D. J. de Jongh and L. Hendriks. Characterizations of strongly equivalent logic programs in intermediate logics. *Theory and Practice of Logic Programming*, 3(3):259–270, 2003.

[5] T. Eiter, W. Faber, N. Leone, and G. Pfeifer. Declarative problem-solving using the DLV system. In J. Minker, ed., *Logic-Based Artificial Intelligence*, pp. 79–103. Kluwer, 2000.

[6] T. Eiter and M. Fink. Uniform equivalence of logic programs under the stable model semantics. Tech. Rep. INFSYS RR-1843-03-08, Inst. für Informationssysteme, TU Wien, 2003. Preliminary Report. Short version in *Proc. ICLP-03*, to appear.

[7] T. Eiter, M. Fink, H. Tompits, and S. Woltran. Simplifying logic programs under uniform and strong equivalence. In *Proc. LPNMR-03*, to appear.

[8] T. Eiter and G. Gottlob. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics & Artificial Intelligence*, 15(3/4):289–323, 1995.

[9] T. Eiter and G. Gottlob. Expressiveness of stable model semantics for disjunctive logic programs with functions. *Journal of Logic Programming*, 33(2):167–178, 1997.

[10] T. Eiter, G. Gottlob, and H. Mannila. Disjunctive datalog. *ACM TODS*, 22(3):364-417, 1997.

[11] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[12] G. Gogic, H. Kautz, C. H. Papadimitriou, and B. Selman. The comparative linguistics of knowledge representation. In *Proc. IJCAI-95*, pp. 862–869. Morgan Kaufmann, 1995.

[13] T. Janhunen, I. Niemelä, P. Simons, and J.-H. You. Partiality and disjunctions in stable model semantics. In *Proc. KR-00*, pp. 411–419. Morgan Kaufmann, 2000.

[14] R. Karp. Reducibility among combinatorial problems, In *Complexity Of Computer Computations*, pp. 85–103, Plenum Press, 1972.

[15] V. Lifschitz, D. Pearce, and A. Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2(4):526–541, 2001.

[16] F. Lin. Reducing strong equivalence of logic programs to entailment in classical propositional logic. In *Proc. KR-02*, pp. 170–176. Morgan Kaufmann, 2002.

[17] F. Lin and Y. Zhao. ASSAT: Computing answer sets of a logic program by SAT solvers. In *Proc. AAAI-2002*, pp. 112–117. AAAI Press / MIT Press, 2002.

[18] M. J. Maher. Equivalences of logic programs. In Minker [21], pp. 627–658.

[19] V. W. Marek, J. Treur, and M. Truszczyński. Representation theory for default logic. *Annals of Mathematics & Artificial Intelligence*, 21(2–4):343–358, 1997.

[20] W. Marek and J. Remmel. On the expressibility of stable logic programming. *Theory and Practice of Logic Programming*, 3(4-5):551–567, 2003.

[21] J. Minker, editor. *Foundations of deductive databases and logic programming*. Morgan Kaufman, Washington DC, 1988.

[22] M. Osorio, J. Navarro, and J. Arrazola. Equivalence in answer set programming. In *Proc. LOPSTR-01*, LNCS 2372, pp. 57–75. Springer, 2001.

[23] D. Pearce. A new logical characterisation of stable models and answer sets. In *Non-Monotonic Extensions of Logic Programming (NMELP 1996)*, LNCS 1216, pp. 57–70. Springer, 1997.

[24] D. Pearce, H. Tompits, and S. Woltran. Encodings for equilibrium logic and logic programs with nested expressions. In *Proc. EPIA-01*, LNCS 2258, pp. 306–320. Springer, 2001.

[25] D. Pearce and A. Valverde. Some types of equivalence for logic programs and equilibrium logic. In *Proc. Joint Conf. on Declarative Programming (APPIA-GULP-PRODE)*, 2003.

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

[27] Y. Sagiv. Optimizing datalog programs. In Minker [21], pp. 659–698.

[28] J. Schlipf. The expressive powers of logic programming semantics. *Journal of Computer and System Sciences*, 51(1):64–86, 1995. Abstract in Proc. PODS 90, pp. 196–204.

[29] P. Simons, I. Niemelä, and T. Soininen. Extending and implementing the stable model semantics. *Artificial Intelligence*, 138:181–234, 2002.

[30] H. Turner. Strong equivalence for logic programs and default theories (made easy). In *Proc. LPNMR-01*, LNCS 2173, pp. 81–92. Springer, 2001.

[31] H. Turner. Strong equivalence made easy: nested expressions and weight constraints. *Theory and Practice of Logic Programming*, 3(4–5):609–622, 2003.