Casting Away Disjunction and Negation under a Generalisation of Strong Equivalence with Projection*

Jörg Pührer and Hans Tompits

Institut für Informationssysteme 184/3, Technische Universität Wien,
Favoritenstraße 9–11, A–1040 Vienna, Austria
{puehrer,tompits}@kr.tuwien.ac.at

Abstract. In answer-set programming (ASP), many notions of program equivalence have been introduced and formally analysed. A particular line of research in this direction aims at studying conditions under which certain syntactic constructs can be eliminated from programs preserving some given equivalence relation. In this paper, we continue this endeavour introducing novel conditions under which disjunction and negation can be eliminated from answer-set programs under *relativised strong equivalence with projection*. This notion is a generalisation of the usual strong-equivalence relation, as introduced by Lifschitz, Pearce, and Valverde, by allowing parametrisable context and output alphabets, which is an important feature in view of practical programming techniques like the use of local variables and modules. We provide model-theoretic conditions that hold for a disjunctive logic program P precisely when there is a program Q, equivalent to P under our considered notion, such that Q is either *positive*, *normal*, or *Horn*, respectively. Moreover, we outline how such a Q, called a *casting of* P, can be obtained, and consider complexity issues.

1 Introduction

An important area of research in answer-set programming (ASP) is devoted to the study of different notions of program equivalence. This particular field emerged with the seminal paper on *strong equivalence* by Lifschitz, Pearce, and Valverde [1]. In contrast to *ordinary equivalence*, which holds whenever two programs have the same answer sets, strong equivalence holds whenever two programs are ordinarily equivalent in *every context*. More formally, two programs P and Q are strongly equivalent iff, for every context program R, $P \cup R$ and $Q \cup R$ are ordinarily equivalent.

Strong equivalence circumvents a particular weakness of ordinary equivalence, viz. that the latter fails to yield a replacement property similar to the one of classical logic. That is to say, under ordinary equivalence, given a program P, replacing some subprogram $Q \subseteq P$ by an equivalent program R may yield an overall program $(P \setminus Q) \cup R$ which is not equivalent to P. Clearly, this is undesirable as far as modular programming and program optimisation is concerned. Strong equivalence does allow subprogram replacements, basically by definition, yet it is too restrictive for some purposes. In particular, strong equivalence does not take standard programming techniques like the use

* This work was partially supported by the Austrian Science Fund (FWF) under projects P18019 and P21698.

of local variables into account, which are ignored in the final output. Thus, it does not admit the *projection* of answer sets to a set of designated output letters.

A generalisation of strong equivalence taking this aspect into account is *relativised strong equivalence with projection*, defined as a special instance of a general framework for defining parameterised program-correspondence notions in ASP [2]. Relativised strong equivalence with projection extends the usual strong-equivalence relation via two parameters: one parameter specifies the alphabet of the context set and the other the alphabet of the output atoms. Thus, it is possible to specify an *input alphabet* and an *output alphabet*, allowing to view programs as black boxes computing some task with respect to a defined input/output behaviour. We note that if no projection is performed (i.e., if the output alphabet coincides with the overall program alphabet), then we arrive at the notion of *relativised strong equivalence*, first studied by Woltran [3].

In this paper, we are interested in the question whether a given disjunctive logic program P can be replaced by a program Q that is from a syntactically simpler class than P preserving relativised strong equivalence with projection (we refer to Q as a *casting* of P). In particular, we are interested in the questions whether a given program can be casted (i) to a program without disjunctions, (ii) to a program without negations, and (iii) to a program without both disjunctions and negations. In other words, we consider the question whether a program can be replaced (preserving relativised strong equivalence with projection) by a *normal*, *positive*, or *Horn program*, respectively. Our results follow a line of research dealing with analogous questions studied previously for ordinary, strong, and uniform equivalence [4, 5] as well as for hyperequivalence [6], but they are the first in this direction to take the issue of projection into account.

The main results of our paper are the following. First of all, we introduce model-theoretic conditions which are necessary and sufficient for having positive answers of our casting questions. For each casting question, we actually provide two different conditions: one in terms of *minimal certificates* [2], and one in terms of *relativised SE-models* [3]. These concepts are the model-theoretic structures underlying relativised strong equivalence with and without projection, respectively, i.e., two programs are equivalent in one of these senses iff they have the same associated structures. Interestingly, our characterisations show that (i) in case the elimination of disjunction is possible, there is always a casting of the given program P just over the atoms $atm(P)$ of P, (ii) in case the elimination of both disjunction and negation is possible, a casting just over $atm(P)$ intersected with the input and output alphabet exists, while (iii) in case the elimination of negation is possible, a casting may introduce new atoms. Secondly, we provide upper complexity bounds for checking our casting questions. It turns out that the complexity of these tasks is not higher than the complexity of checking relativised strong equivalence with projection, which lies on the fourth level of the polynomial hierarchy. Thirdly, we outline how a casting can be obtained, in case a casting exists.

2 Preliminaries

Syntax and Semantics of Answer-Set Programs. We deal with finite propositional disjunctive logic programs containing rules (over a set At of atoms) of form

$$a_1 \vee \cdots \vee a_l \leftarrow b_1, \ldots, b_m, not\, b_{m+1}, \ldots, not\, b_n,$$

where $l \geq 0$, $n \geq m \geq 0$, all a_i, b_j are from At, and not denotes default negation. A rule r as above is *normal*, if $l \leq 1$; *positive*, if $m = n$; a *constraint*, if $l = 0$ and $m + n > 0$; and *Horn* if it is positive and normal. We define the *head* of r as $H(r)$ = $\{a_1, \ldots, a_l\}$ and the *body* of r as $B(r)$ = $\{b_1, \ldots, b_m, not\, b_{m+1}, \ldots, not\, b_n\}$. Furthermore, we also define $B^+(r)$ = $\{b_1, \ldots, b_m\}$ and $B^-(r)$ = $\{b_{m+1}, \ldots, b_n\}$. A *disjunctive logic program* (*over* At), or simply a *program*, is a finite set of rules (over At). A program P is normal (resp., positive, Horn) if every rule in P is normal (resp., positive, Horn). We use N, P, and H to refer to the classes of normal, positive, and Horn programs, respectively. Finally, $atm(P)$ denotes the set of all atoms occurring in P.

Let I be an *interpretation*, i.e., a set of atoms. I *satisfies* a rule r, symbolically $I \models r$, iff $I \cap H(r) \neq \emptyset$ whenever $B^+(r) \subseteq I$ and $I \cap B^-(r) = \emptyset$ jointly hold. I is a *model* of a program P, symbolically $I \models P$, iff $I \models r$, for all $r \in P$. I is an *answer set* [7] of P iff I is a minimal model of P^I, where

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

is the *reduct of* P *relative to* I. The set of all answer sets of P is denoted by $\mathcal{AS}(P)$.

Strong Equivalence and its Generalisations. Next, we introduce basic equivalence notions relevant for our purposes and provide model-theoretic characterisations for them. All of the equivalence notions discussed are special instances of a general framework for specifying parameterised program correspondence relations between answer-set programs [2].

We begin with *strong equivalence*, originally introduced by Lifschitz, Pearce, and Valverde [1]: Two programs, P and Q, are strongly equivalent, symbolically $P \equiv_s Q$, iff $\mathcal{AS}(P \cup R) = \mathcal{AS}(Q \cup R)$, for any program R (which is also referred to as a *context program*). Interestingly, strong equivalence corresponds to equivalence in the logic of here-and-there [8], HT, which, from a semantical point of view, is intuitionistic logic restricted to two worlds, "here" and "there". More specifically, two programs, viewed as logical theories, are strongly equivalent iff they have the same models in HT [1]. Emerging from this observation, Turner [9] characterised strong equivalence in terms of *SE-models* which directly correspond to models in HT: By an *SE-interpretation* we understand a pair (X, Y) of interpretations $X, Y \subseteq At$ such that $X \subseteq Y$. If $X = Y$, then (X, Y) is *total*, otherwise (X, Y) is *non-total*. In view of HT, the first component of an SE-interpretation is identified with the world "here", whilst the second component refers to the world "there". An SE-interpretation (X, Y) is an *SE-model* of a program P over At if $Y \models P$ and $X \models P^Y$. The set of all SE-models of P is denoted by $SE(P)$. It then holds that two programs P and Q are strongly equivalent iff $SE(P) = SE(Q)$. By limiting the context programs to be defined over particular alphabets only, we arrive at the notion of *relativised strong equivalence* [3]. Formally, given an alphabet $A \subseteq At$, programs P and Q over At are strongly equivalent *relative to* A, symbolically $P \equiv_s^A Q$, iff $\mathcal{AS}(P \cup R) = \mathcal{AS}(Q \cup R)$, for any program R over A. Model-theoretically, strong equivalence relative to A is captured in terms of A-*SE-models*: an SE-interpretation (X, Y) is an A-*SE-model* of a program P over At if (i) $Y \models P$, (ii) for all $Y' \subset Y$ with $Y' \cap A = Y \cap A$, it holds that $Y' \not\models P^Y$, and (iii) if $X \subset Y$, there is an $X' \subset Y$ with $X' \cap A = X$ such that $X' \models P^Y$. The set of all A-SE-models of P is denoted by $SE^A(P)$. For programs P and Q, it holds that $P \equiv_s^A Q$ iff $SE^A(P) = SE^A(Q)$.

A set $\mathcal{S}$ of SE-interpretations is *A-complete* if, for all $(X,Y) \in \mathcal{S}$ with $X \subset Y$, it holds that $(Y,Y) \in \mathcal{S}$ and $X \subset (Y \cap A)$, and, for all $(X,Y), (Z,Z) \in \mathcal{S}$ with $Y \subset Z$, it holds that $(X \cap A, Z) \in \mathcal{S}$. Note that $SE^A(P)$ is A-complete for every program P and, conversely, for every A-complete set $\mathcal{S}$ of SE-interpretations, there exists a program Q with $SE^A(Q) = \mathcal{S}$.

A further relaxation of strong equivalence is *relativised strong equivalence with projection* [2]: Given sets $A, O \subseteq At$, two programs P and Q over At are *strongly equivalent relative to A under projection to O*, or *$\langle A, O\rangle$-equivalent*, symbolically $P \equiv^A_{|_O} Q$, iff, for any program R over A, it holds that $\{I \cap O \mid I \in \mathcal{AS}(P \cup R)\} = \{I \cap O \mid I \in \mathcal{AS}(Q \cup R)\}$. For better readability, let us write $I|_O$ for $I \cap O$, for an interpretation I and an alphabet O, and define $S|_O = \{I|_O \mid I \in S\}$ for a set of interpretations. Then, we have that $P \equiv^A_{|_O} Q$ iff $\mathcal{AS}(P \cup R)|_O = \mathcal{AS}(Q \cup R)|_O$, for any program R over A.

Clearly, relativised strong equivalence with projection includes strong equivalence and relativised strong equivalence as special cases. Indeed, for all programs P, Q over At, $P \equiv_s Q$ iff $P \equiv^{At}_{|_{At}} Q$, and $P \equiv^A_s Q$ iff $P \equiv^A_{|_{At}} Q$. Certainly, we also have $P \equiv_s Q$ iff $P \equiv^{At}_s Q$.

In the spirit of the model-theoretic characterisations above, $\langle A, O\rangle$-equivalence can be characterised as follows [2]: Let $A, O \subseteq At$ be sets of atoms. A *certificate structure* is a pair (Ξ, Y), where Ξ is a set of interpretations and Y is an interpretation. For a program P over At, a certificate structure (Ξ, Y) is an *$\langle A, O\rangle$-certificate* of P if there is some $(Y', Y') \in SE^A(P)$ with $Y = Y'|_{A \cup O}$ and $\Xi = \{X \mid (X, Y') \in SE^A(P), X \subset Y'\}$. An $\langle A, O\rangle$-certificate (Ξ, Y) of P is *minimal* if, for any $\langle A, O\rangle$-certificate (Ξ', Y) of P, $\Xi' \subseteq \Xi$ implies $\Xi' = \Xi$. By $\mathscr{C}_{A,O}(P)$ we denote the set of all $\langle A, O\rangle$-certificates of P, and $\mathscr{C}^m_{A,O}(P)$ stands for the set of all minimal $\langle A, O\rangle$-certificates of P. Then, for two programs P and Q, $P \equiv^A_{|_O} Q$ holds iff $\mathscr{C}^m_{A,O}(P) = \mathscr{C}^m_{A,O}(Q)$.

For our later purposes, we need to characterise minimal $\langle A, O\rangle$-certificates directly in terms of A-SE-models. To this end, we introduce the following notion:

Definition 1. *Let $A, O \subseteq At$ be sets of atoms, $\mathcal{S}$ a set of SE-interpretations over At, and $(X,Y) \in \mathcal{S}$. Then, (X,Y) is* $\langle A, O\rangle$-optimal *in $\mathcal{S}$ if there is no $(Y', Y') \in \mathcal{S}$ with $Y|_{A \cup O} = Y'|_{A \cup O}$ and $\{U \mid (U, Y') \in \mathcal{S}, U \subset Y'\} \subset \{U \mid (U, Y) \in \mathcal{S}, U \subset Y\}$.*

We then have the following property:

Theorem 1. *Let $A, O \subseteq At$ be sets of atoms and P a program over At. Then, $(\Xi, Y) \in \mathscr{C}^m_{A,O}(P)$ iff there is some (Y', Y') that is $\langle A, O\rangle$-optimal in $SE^A(P)$ with $Y'|_{A \cup O} = Y$ and $\Xi = \{X \mid (X, Y') \in SE^A(P), X \subset Y'\}$.*

3 Main Results

In this section, we present necessary and sufficient conditions such that for a given disjunctive logic program P over At and given sets $A, O \subseteq At$, there is a program Q which is $\langle A, O\rangle$-equivalent to P, where Q is either normal, positive, or Horn. We call such a Q, if it exists, an *$\langle A, O\rangle$-$\mathcal{C}$-casting of P*, for $\mathcal{C} \in \{\mathrm{N}, \mathrm{P}, \mathrm{H}\}$, referring to the normal, positive, or Horn case, respectively, or simply a *casting of P* if no ambiguity arises.

$r_1: b \leftarrow p, not\ c$	$r_5: p \vee j \leftarrow b, not\ d$	$r_9: v \leftarrow j, not\ c, not\ d$
$r_2: b \leftarrow p, not\ d$	$r_6: p \leftarrow j, not\ c, not\ d$	$r_{10}: c \leftarrow j, not\ d, not\ v$
$r_3: b \leftarrow j, not\ c$	$r_7: p \leftarrow v$	$r_{11}: d \leftarrow j, not\ c, not\ v$
$r_4: b \leftarrow j, not\ d$	$r_8: \leftarrow p, d$	

Fig. 1. Program P_{ex}.

In each case, we present two kinds of model-theoretic conditions—one based on certificate structures and one on SE-interpretations. As well, we approach our conditions by first addressing the case of relativised strong equivalence *without projection*, which can be directly obtained from previous results about casting under hyperequivalence [6].

We start with presenting a typical scenario for our casting questions, serving as a running example for our subsequent elaborations.

Example 1. Consider a party-attendance problem for determining who will attend a party based on given preferences and constraints of potential party guests. Assume the following circumstances: Our friend Betty only attends if Peter or Mary-Jane does and if there is no need to dance or no cheesy music playing. If Betty comes, in case there is no dancing, she will bring Peter or Mary-Jane along. As Peter needs to talk to Mary-Jane, he will definitely attend if Mary-Jane comes and there is neither dancing nor cheesy music playing. He also comes if there is only vegetarian food, but as he hates dancing he is not coming if dancing is required. Mary-Jane is a party-tiger and a die-hard vegetarian, well-known to force people to either dance, listen to cheesy music, or eat only vegetarian meals.

Program P_{ex} in Fig. 1 is an encoding of this information, where atoms b, p, and j represent the attendance of Betty, Peter, or Mary-Jane, respectively, and d, v, and c indicate dancing, vegetarian food, or cheesy music at the party. Note that P_{ex} is understood as representing partial information only, since we expect further preferences and constraints from Betty, Peter, or Mary-Jane concerning their attendance. Thus, P_{ex} will later be joined with further rules containing atoms from $A = \{b, p, j\}$.

As discussed below, no program Q exists not involving disjunction or negation that is strongly equivalent to P_{ex}, even when context programs are built from atoms in A only. However, assume we are only interested in who attends the party but we do not care about which party activities take place. Then, we do not mind if the answer sets of a casting disagree with those of P_{ex} on atoms in $\{d, v, c\}$. Hence, we are interested whether there is a casting Q of P_{ex} under relativised strong equivalence under projection. Our model-theoretic properties presented later on allow to answer this question.

3.1 Normal Logic Programs

For getting an intuition of the mechanisms that underlie our characterisations, it is helpful to deliberate how the syntactic class of a program influences its SE-models. Consider a normal program P. Then, for any interpretation Y, the reduct P^Y is a Horn program. Since the models of a Horn program are closed under intersection, it follows that for all SE-models $(X_1, Y), (X_2, Y)$ of P, $(X_1 \cap X_2, Y)$ must also be an SE-model of P, as both $X_1 \models P^Y$ and $X_2 \models P^Y$ holds. Following Eiter et al. [5], let us call a collection $\mathcal{S}$

of SE-interpretations *closed under here-intersection* if $(X_1, Y) \in \mathcal{S}$ and $(X_2, Y) \in \mathcal{S}$ implies $(X_1 \cap X_2, Y) \in \mathcal{S}$. So, our above argument shows that the set of all SE-models of a normal program is closed under here-intersection. Consequently, if, for a DLP P, there is some strongly equivalent normal program Q, then $SE(P)$ must be closed under here-intersection. However, as shown by Eiter et al. [5], closure under here-intersection is also a *sufficient* condition to guarantee the existence of a normal program Q being strongly equivalent to a given DLP P. Interestingly, this characterisation holds for strong equivalence relative to A as well, but using A-SE-models instead of SE-models.

Proposition 1 ([6]). *Given a set $A \subseteq At$ of atoms and a program P over At, a normal program Q over At exists with $P \equiv_s^A Q$ iff $SE^A(P)$ is closed under here-intersection.*

Example 2. Consider the program P_{ex} from Example 1 and $A = \{b, p, j\}$. Then,[1]

$$\begin{aligned} SE^A(P_{ex}) = \{&(\emptyset, \emptyset), (\emptyset, bp), (bp, bp), (\emptyset, bjc), (bjc, bjc), (\emptyset, bjd), (b, bjd),\\ &(bjd, bjd), (\emptyset, bpjv), (bp, bpjv), (bpjv, bpjv), (\emptyset, bpjc),\\ &(bp, bpjc), (bj, bpjc), (bpjc, bpjc)\}. \end{aligned}$$

As $(bp, bpjc), (bj, bpjc) \in SE^A(P_{ex})$ but $(b, bpjc) \notin SE^A(P_{ex})$, $SE^A(P_{ex})$ is not closed under here-intersection. Hence, there is no normal program Q with $P_{ex} \equiv_s^A Q$.

Turning to the case of relativised strong equivalence with projection, we now define our key properties for casting to normal programs—first for certificate structures and then for SE-interpretations.

Definition 2. *Let $A, O \subseteq At$ be sets of atoms. Then, a set $\mathscr{S}$ of certificate structures is $\langle A, O\rangle_{\mathsf{N,c}}$-*compliant *if, for every $(\Xi, Y) \in \mathscr{S}$, Ξ is closed under intersection.*

Definition 3. *Let $A, O \subseteq At$ be sets of atoms. Then, a set $\mathcal{S}$ of SE-interpretations is $\langle A, O\rangle_{\mathsf{N,s}}$-*compliant *if the set of $\langle A, O\rangle$-optimal SE-interpretations in $\mathcal{S}$ is closed under here-intersection.*

Consider a normal program P. Since, in view of Proposition 1, for every (X_1, Y), $(X_2, Y) \in SE^A(P)$, we have that $(X_1 \cap X_2, Y) \in SE^A(P)$, and, by Definition 1, either all or none of (X_1, Y), (X_2, Y), and $(X_1 \cap X_2, Y)$ are $\langle A, O\rangle$-optimal in $SE^A(P)$, it follows that $SE^A(P)$ is $\langle A, O\rangle_{\mathsf{N,s}}$-compliant. Hence, if a DLP P is $\langle A, O\rangle$-equivalent to a normal program Q, $SE^A(P)$ must be $\langle A, O\rangle_{\mathsf{N,s}}$-compliant. As the next result shows, the converse also holds, as well as similar relations for $\langle A, O\rangle_{\mathsf{N,c}}$-compliance.

Theorem 2. *Let $A, O \subseteq At$ be sets of atoms and P a program over At. Then, the following statements are equivalent:*

1. *$\mathscr{C}^m_{A,O}(P)$ is $\langle A, O\rangle_{\mathsf{N,c}}$-compliant;*
2. *$SE^A(P)$ is $\langle A, O\rangle_{\mathsf{N,s}}$-compliant;*
3. *a normal program Q over $At \cup At'$ exists for some universe At' with $P \equiv^A_{|O} Q$;*
4. *a normal program Q over $atm(P)$ exists such that $P \equiv^A_{|O} Q$.*

[1] For brevity, in what follows, we omit braces and commas for interpretations, i.e., we write, e.g., bp instead of $\{b, p\}$.

Note that if a casting exists, then there is a casting Q that is built just of atoms from the input program P. Indeed, it is sufficient to remove all A-SE-models from $SE^A(P)$ that are not $\langle A, O\rangle$-optimal in order to construct the set $\mathcal{S}$ of A-SE-models of Q. We will see later on that for other casting questions a solution can be found only at the expense of introducing new atoms into the casting.

Example 3. The minimal $\langle A, O\rangle$-certificates of our running example P_{ex} for $A = O = \{b, p, j\}$ are given by $\mathscr{C}^m_{A,O}(P_{ex}) = \{(\emptyset, \emptyset), (\{\emptyset\}, bp), (\{\emptyset\}, bj), (\{\emptyset, bp\}, bpj)\}$. As $\emptyset$, $\{\emptyset\}$, and $\{\emptyset, bp\}$ are closed under intersection, $\mathscr{C}^m_{A,O}(P_{ex})$ is $\langle A, O\rangle_{\mathsf{N,c}}$-compliant. Hence, there exists a normal program being $\langle A, O\rangle$-equivalent to P_{ex}. For instance, the following program is one in question:[2]

$$Q = \{b \leftarrow p; \quad b \leftarrow j; \quad j \leftarrow c; \quad p \leftarrow b, not\ c; \quad p \leftarrow b, not\ j; \quad p \leftarrow j, not\ c;$$
$$c \leftarrow b, not\ p; \quad c \leftarrow j, not\ p; \quad v \leftarrow j, not\ c\}.$$

3.2 Positive Logic Programs

A positive program P satisfies $P = P^I$, for any interpretation I. Hence, if $(X, Y) \in SE(P)$, then $(X, X) \in SE(P)$ as well, since $X \models P$ trivially implies $X \models P^I$. This motivates the following definition: A set $\mathcal{S}$ of SE-interpretations for which $(X, Y) \in \mathcal{S}$ implies $(X, X) \in \mathcal{S}$ is called *here-total*. Considering strong equivalence relative to A, here-totality of A-SE-models does not make sense, as the here-component of an A-SE-model is not a model of the program but only a *projection* of a model. Hence, we need to adapt the property as follows: A set $\mathcal{S}$ of SE-interpretations is *A-here-total* if, for all $(X, Y) \in \mathcal{S}$ with $X \subset Y$, some $(X', X') \in \mathcal{S}$ exists with $X'|_A = X$ and $X' \subset Y$.

Proposition 2 ([6]). *Let $A \subseteq At$ be a set of atoms and P a program over At. Then, there is a positive program Q over At such that $P \equiv^A_s Q$ iff $SE^A(P)$ is A-here-total.*

Example 4. Consider again our running example. Since $(b, bjd) \in SE^A(P_{ex})$, for $A = \{b, p, j\}$, but there is no $(X', X') \in SE^A(P_{ex})$ with $X'|_A = \{b\}$, $SE^A(P_{ex})$ is not A-here-total. Hence, there is no positive program Q with $P_{ex} \equiv^A_s Q$.

We continue with the characterising conditions for casting to positive programs.

Definition 4. *Let $A, O \subseteq At$ be sets of atoms. Then, a set $\mathscr{S}$ of certificate structures is $\langle A, O\rangle_{\mathsf{P,c}}$-*compliant *if, for every $X \in \Xi$ where $(\Xi, Y) \in \mathscr{C}^m_{A,O}(P)$, there is some $(\Xi', X') \in \mathscr{C}^m_{A,O}(P)$ such that $X'|_O \subseteq Y|_O$, $X'|_A = X$, and $\Xi' \subseteq \Xi$.*

Definition 5. *Let $A, O \subseteq At$ be sets of atoms. Then, a set $\mathcal{S}$ of SE-interpretations is $\langle A, O\rangle_{\mathsf{P,s}}$-*compliant *if, for every (X, Y) which is $\langle A, O\rangle$-optimal in $\mathcal{S}$ with $X \subset Y$, there is some $(X', X') \in \mathcal{S}$ with $X'|_O \subseteq Y|_O$, $X'|_A = X$, and $(V, X') \in \mathcal{S}$ with $V \subset X'$ implies $(V, Y) \in \mathcal{S}$.*

Intuitively, the implication in the last part of the above condition—which does not have a pendant in the definition of A-here-totality—ensures A-completeness of some A-here-total variant $\mathcal{S}'$ of $\mathcal{S}$ that amounts to the set of A-SE-models of the desired casting.

[2] Section 4 contains a description how to construct a casting.

Theorem 3. *Let $A, O \subseteq At$ be sets of atoms and P a program over At. Then, the following statements are equivalent:*

1. *$\mathscr{C}^m_{A,O}(P)$ is $\langle A, O\rangle_{\mathsf{P,c}}$-compliant;*
2. *$SE^A(P)$ is $\langle A, O\rangle_{\mathsf{P,s}}$-compliant;*
3. *a positive program Q over $At \cup At'$ exists with $P \equiv^A_{|O} Q$, for some universe At'.*

Example 5. It can be checked that $\mathscr{C}^m_{A,O}(P_{ex})$ is $\langle A, O\rangle_{\mathsf{P,c}}$-compliant for $A = O = \{b, p, j\}$ and hence there is a positive program being $\langle A, O\rangle$-equivalent to P_{ex}, e.g., $Q = \{p \vee c \leftarrow b;\ b \leftarrow c;\ b \leftarrow p;\ b \leftarrow j;\ j \leftarrow c\}$.

In contrast to Example 5, where Q contains only atoms from P_{ex}, unlike in the case of normal programs, building an $\langle A, O\rangle$-equivalent positive program might require atoms not occurring in the original program. The following program illustrates this point.

Example 6. Consider program P over $At = \{a, b, c, h\}$, given by the following rules:

$$\begin{array}{lll} a \leftarrow not\ b, not\ c; & a \leftarrow not\ h; & h \leftarrow not\ a, not\ c; \\ b \leftarrow not\ a, not\ c; & b \leftarrow not\ h; & h \leftarrow not\ b, not\ c. \end{array}$$

For $A = \{a, b, c\}$ and $O = \emptyset$, we obtain $\mathscr{C}^m_{A,O}(P) = \{(\emptyset, a), (\emptyset, b), (\emptyset, ab), (\{a, ab\}, abc)\}$. Although, $\mathscr{C}^m_{A,O}(P)$ is $\langle A, O\rangle_{\mathsf{P,c}}$-compliant, it can be shown that no $\langle A, O\rangle$-equivalent positive program exists containing only atoms from At.

3.3 Horn Programs

Horn programs are both normal and positive. However, it is not sufficient to combine the criteria of casting to normal and positive programs in order to obtain a characterisation for Horn programs. The reason is that the elimination of disjunction may introduce negation and vice versa. As stated earlier, the classical models of Horn theories are intersection closed. In terms of SE-models of such a program, this means that the there-components occurring in the SE-models, being the models of the program, are closed under intersection [4]. In analogy to closure under here-intersection, this property is called *closure under there-intersection*. More formally, a set $\mathcal{S}$ of SE-interpretations is closed under there-intersection iff, whenever $(X, X) \in \mathcal{S}$ and $(Y, Y) \in \mathcal{S}$, then $(X \cap Y, X \cap Y) \in \mathcal{S}$. For a program P, a strongly equivalent Horn program Q exists iff $SE(P)$ is here-total and closed under there-intersection [4]. Note that closure under here-intersection follows automatically from these two conditions. For relativised strong equivalence, similar to here-totality for the positive case, closure under there-intersection has to be adapted with respect to the context alphabet: A set $\mathcal{S}$ of SE-interpretations is *A-closed under there-intersection* if, for all $(X, X), (Y, Y) \in \mathcal{S}$, there is some $(Z, Z) \in \mathcal{S}$ with $Z \subseteq (X \cap Y)$ and $Z|_A = (X \cap Y)|_A$.

Proposition 3 ([6]). *Let $A \subseteq At$ be a set of atoms and P a program over At. Then, there exists a Horn program Q over At such that $P \equiv^A_s Q$ iff $SE^A(P)$ is A-here-total and A-closed under there-intersection.*

The restriction of A-closure under there-intersection in interplay with A-completeness imposes an interesting side effect on the A-SE-models of Horn programs.

Theorem 4. *Let $A, O \subseteq At$ be sets of atoms and Q a Horn program over At. Then, $(Z_1, Z_1), (Z_2, Z_2) \in SE^A(Q)$ with $Z_1|_A = Z_2|_A$ only if $Z_1 = Z_2$.*

As a consequence of that, all A-SE-models of Q are $\langle A, O\rangle$-optimal ones, since for some $(X, Y) \in SE^A(Q)$ it holds that $(Y, Y) \in SE^A(Q)$, and hence, by Theorem 4, there cannot be any $(Y', Y') \in SE^A(Q)$ with $Y' \neq Y$ and $Y'|_{A\cup O} = Y|_{A\cup O}$, which is a requirement for violation of $\langle A, O\rangle$-optimality. Due to this restriction on the A-SE-models, in contrast to the case of positive programs, we do not need additional atoms for building a Horn casting Q, if one exists for some given program P. Indeed, Q can be built from atoms in $atm(P)|_{A\cup O}$ only.

We obtain the following conditions for casting to Horn programs:

Definition 6. *Let $A, O \subseteq At$ be sets of atoms. Then, a set $\mathscr{S}$ of certificate structures is* $\langle A, O\rangle_{\mathsf{H,c}}$-compliant *if*

(*i*) *for all $(\Xi, Y) \in \mathscr{S}$ and all $X \in \Xi$ there is some $(\Xi', X') \in \mathscr{S}$ such that $X' \subset Y$ and $X'|_A = X$,*
(*ii*) *for all $(\Xi_1, Y_1), (\Xi_2, Y_2) \in \mathscr{S}$ there is some $(\Xi', Z) \in \mathscr{S}$ with $Z \subseteq (Y_1 \cap Y_2)$ and $Z|_A = (Y_1 \cap Y_2)|_A$, and*
(*iii*) *for all $(\Xi_Y, Y), (\Xi_Z, Z) \in \mathscr{S}$ with $Y \subseteq Z$ it holds that $\Xi_Y \subseteq \Xi_Z$. Moreover, if $Y \neq Z$, then $Y|_A \in \Xi_Z$.*

Definition 7. *Let $A, O \subseteq At$ be sets of atoms. Then, a set $\mathcal{S}$ of SE-interpretations is* $\langle A, O\rangle_{\mathsf{H,s}}$-compliant *if*

(*i*) *for every (X, Y) which is $\langle A, O\rangle$-optimal in $\mathcal{S}$ with $X \subset Y$, some $(X', X') \in \mathcal{S}$ exists such that $X'|_O \subseteq Y|_O$ and $X'|_A = X$,*
(*ii*) *for all $(Y_1, Y_1), (Y_2, Y_2)$ which are $\langle A, O\rangle$-optimal in $\mathcal{S}$, some $(Z, Z) \in \mathcal{S}$ exists with $Z|_O \subseteq (Y_1 \cap Y_2)|_O$ and $Z|_A = (Y_1 \cap Y_2)|_A$, and*
(*iii*) *for all $(X, Y), (Z, Z)$ which are are $\langle A, O\rangle$-optimal in $\mathcal{S}$ such that $X \subset Y$ and $Y|_{A\cup O} \subseteq Z_{A\cup O}$, $(X, Z) \in \mathcal{S}$ holds. Moreover, if $Y \neq Z$, then $(Y|_A, Z) \in \mathcal{S}$.*

The individual Subproperties (i), (ii), and (iii) directly correspond to the equally labelled ones from Definition 6. In both definitions, (i) and (ii) are the pendants of A-here-totality and A-closure under there-intersection for $\langle A, O\rangle$-equivalence, respectively. Note that (i) differs from $\langle A, O\rangle_{\mathsf{P,s}}$-compliance in not having a subcondition for A-completeness. Instead, Subproperty (iii) corresponds to A-completeness of the $\langle A, O\rangle$-optimal SE-interpretations in $\mathcal{S}$.

Theorem 5. *Let $A, O \subseteq At$ be sets of atoms and P a program over At. Then, the following statements are equivalent:*

1. *$\mathscr{C}^m_{A,O}(P)$ is $\langle A, O\rangle_{\mathsf{H,c}}$-compliant;*
2. *$SE^A(P)$ is $\langle A, O\rangle_{\mathsf{H,s}}$-compliant;*
3. *a Horn program Q over $At \cup At'$ exists, for some universe At', with $P \equiv^A_{|O} Q$;*
4. *a Horn program Q over $atm(P)|_{A\cup O}$ exists such that $P \equiv^A_{|O} Q$.*

Example 7. As we have seen, $\mathscr{C}^m_{A,O}(P_{ex})$ is $\langle A,O\rangle_{\mathsf{N,c}}$-compliant and $\langle A,O\rangle_{\mathsf{P,c}}$-compliant for $A = O = \{b,p,j\}$. However, it turns out that $\mathscr{C}^m_{A,O}(P_{ex})$ is not $\langle A,O\rangle_{\mathsf{H,c}}$-compliant. Indeed, Condition (ii) of Definition 6 is violated as $(\{\emptyset\}, bp), (\{\emptyset\}, bj) \in \mathscr{C}^m_{A,O}(P_{ex})$ but there is no $(\Xi, Z) \in \mathscr{C}^m_{A,O}(P_{ex})$ with $Z \subseteq \{b\}$ and $Z|_A = \{b\}$. This means that, for preserving $\langle A,O\rangle$-equivalence, we can remove disjunctions from P_{ex} only by introducing negation and vice versa.

However, if we are not interested in who attends the party but whether we have to bear vegetarian food, dancing, or cheesy music, we can obtain a Horn program corresponding to P_{ex}: If we set $A' = O' = \{v,d,c\}$, we expect only input programs that mention party activities and we are only interested in output related to them. The minimal $\langle A',O'\rangle$-certificates of P_{ex} are given by $\mathscr{C}^m_{A',O'}(P_{ex}) = \{(\emptyset,\emptyset), (\{\emptyset\}, d), (\{\emptyset\}, v), (\{\emptyset\}, c), (\{\emptyset, v, c\}, vc), (\{\emptyset, d, c\}, dc)\}$. All conditions of Definition 6 apply, therefore $\mathscr{C}^m_{A',O'}(P_{ex})$ is $\langle A',O'\rangle_{\mathsf{H,c}}$-compliant. An example of a Horn program $\langle A',O'\rangle$-equivalent to P_{ex} is $Q = \{c \leftarrow v, d; \leftarrow d, v, c\}$.

3.4 Computational Complexity

We finally discuss the complexity of our casting questions. While checking strong equivalence is coNP-complete [10], deciding relativised strong equivalence and hyperequivalence is already Π^P_2-complete [11, 12]. Checking whether, for a given program P, a hyperequivalent casting exists is Π^P_2-complete for normal castings and Π^P_3-complete for positive castings. Furthermore, for the case of Horn programs, the problem lies in Π^P_3 [6]. Although testing relativised strong equivalence with projection is presumably harder, viz. Π^P_4-complete [2], the next result shows that casting under this equivalence notion does not yield an additional source of complexity:

Theorem 6. *Given a program P over At and $A, O \subseteq At$, deciding whether $SE^A(P)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{s}}$-compliant, for $\mathcal{C} \in \{\mathsf{N},\mathsf{P},\mathsf{H}\}$, is in Π^P_4, as is deciding whether $\mathscr{C}^m_{A,O}(P)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{c}}$-compliant.*

The major source of complexity is checking $\langle A,O\rangle$-optimality. Important for establishing the upper complexity bound is the observation that checking $\langle A,O\rangle$-optimality is not required in the consequent of any of the (sub)conditions of $\langle A,O\rangle_{\mathcal{C},\mathsf{s}}$-compliance.

4 Proof Outline of Main Results

We briefly outline how Theorems 2, 3, and 5 can be shown, thereby providing methods for constructing castings. First of all, from the definitions of an A-SE-model, a minimal $\langle A,O\rangle$-certificate, and $\langle A,O\rangle$-optimality, the following lemma can be shown:

Lemma 1. *For any program P over At, any $A, O \subseteq At$, and any $\mathcal{C} \in \{\mathsf{N},\mathsf{P},\mathsf{H}\}$, $\mathscr{C}^m_{A,O}(P)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{c}}$-compliant iff $SE^A(P)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{s}}$-compliant.*

The general proof schema, then, proceeds by showing, for a given program P over At and $\mathcal{C} \in \{\mathsf{N},\mathsf{P},\mathsf{H}\}$ that

1. if P has a casting $Q \in \mathcal{C}$ over some $At' \supseteq At$, then $\mathscr{C}^m_{A,O}(P)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{c}}$-compliant, and

2. if $\mathscr{C}^m_{A,O}(P)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{c}}$-compliant, then there exists a casting $Q \in \mathcal{C}$ over some At'', where (i) $At'' = atm(P)$ for $\mathcal{C} = \mathsf{N}$, (ii) $At'' \supseteq At$ for $\mathcal{C} = \mathsf{P}$, and (iii) $At'' = atm(P)|_{A\cup O}$ for $\mathcal{C} = \mathsf{H}$.

Showing Item 1 can be reduced to showing that for every program $Q' \in \mathcal{C}$ it holds that $\mathscr{C}^m_{A,O}(Q')$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{c}}$-compliant. Hence, given a casting $Q \in \mathcal{C}$ of P, $\mathscr{C}^m_{A,O}(Q)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{c}}$-compliant and, in view of $\mathscr{C}^m_{A,O}(P) = \mathscr{C}^m_{A,O}(Q)$, so is $\mathscr{C}^m_{A,O}(P)$.

For proving Item 2, we devised algorithms that allow for obtaining a casting $Q \in \mathcal{C}$ over At'' whenever $\mathscr{C}^m_{A,O}(P)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{c}}$-compliant. These involve three transformations, which have the following properties:

(i) $\mathfrak{f}^{\mathcal{C}}_{A,O}$ maps a program P over At to $\mathfrak{f}^{\mathcal{C}}_{A,O}(P) = SE^A(Q)$, where Q is a program over At'' such that $Q \in \mathcal{C}$, providing $SE^A(P)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{s}}$-compliant. Furthermore, (X,Y) is $\langle A,O\rangle$-optimal in $SE^A(P)$ iff (X,Y) is $\langle A,O\rangle$-optimal in $\mathfrak{f}^{\mathcal{C}}_{A,O}(P)$.

(ii) $\mathfrak{c}^{\mathcal{C}}_A$ maps a set $\mathcal{S}$ of SE-interpretations over At'' to $\mathfrak{c}^{\mathcal{C}}_A(\mathcal{S}) = SE(Q)$, where Q is a program over At'' such that $Q \in \mathcal{C}$ and $SE^A(Q) = \mathcal{S}$. If such a Q does not exist, then $\mathfrak{c}^{\mathcal{C}}_A$ is undefined.

(iii) $\mathfrak{p}^{\mathcal{C}}$ maps a set $\mathcal{S}$ of SE-interpretations over At'' into $\mathfrak{p}^{\mathcal{C}}(\mathcal{S}) = Q$, where Q is a program over At'' such that $Q \in \mathcal{C}$ and $SE(Q) = \mathcal{S}$. If such a Q does not exist, then $\mathfrak{c}^{\mathcal{C}}_A$ is undefined.

Given a program P for which $SE^A(P)$ is $\langle A,O\rangle_{\mathcal{C},\mathsf{s}}$-compliant, a casting Q of P is given by $Q = \mathfrak{p}^{\mathcal{C}}(\mathfrak{c}^{\mathcal{C}}_A(\mathfrak{f}^{\mathcal{C}}_{A,O}(P)))$. Indeed, by the properties of $\mathfrak{f}^{\mathcal{C}}_{A,O}$, the set of $\langle A,O\rangle$-optimal A-SE-models of P coincides with the set of $\langle A,O\rangle$-optimal elements in $\mathfrak{f}^{\mathcal{C}}_{A,O}(P)$. But, again by construction of $\mathfrak{f}^{\mathcal{C}}_{A,O}$, there exists some $Q' \in \mathcal{C}$ such that $SE^A(Q') = \mathfrak{f}^{\mathcal{C}}_{A,O}(P)$. Hence, $\mathfrak{c}^{\mathcal{C}}_A(\mathfrak{f}^{\mathcal{C}}_{A,O}(P))$ is defined and $\mathfrak{c}^{\mathcal{C}}_A(\mathfrak{f}^{\mathcal{C}}_{A,O}(P)) = SE(Q'')$, for some $Q'' \in \mathcal{C}$ such that $SE^A(Q'') = \mathfrak{f}^{\mathcal{C}}_{A,O}(P)$. By the latter condition, we obtain in turn that $\mathfrak{p}^{\mathcal{C}}$ is defined and $\mathfrak{p}^{\mathcal{C}}(\mathfrak{c}^{\mathcal{C}}_A(\mathfrak{f}^{\mathcal{C}}_{A,O}(P))) = Q$, where Q is a program such that $Q \in \mathcal{C}$ and $SE(Q) = \mathfrak{c}^{\mathcal{C}}_A(\mathfrak{f}^{\mathcal{C}}_{A,O}(P))$. Now, since $\mathfrak{c}^{\mathcal{C}}_A(\mathfrak{f}^{\mathcal{C}}_{A,O}(P)) = SE(Q'')$, we have that $SE(Q) = SE(Q'')$, and thus $SE^A(Q) = SE^A(Q'')$. But $SE^A(Q'') = \mathfrak{f}^{\mathcal{C}}_{A,O}(P)$, so $SE^A(Q) = \mathfrak{f}^{\mathcal{C}}_{A,O}(P)$. Consequently, the set of $\langle A,O\rangle$-optimal A-SE-models of P coincides with the set of $\langle A,O\rangle$-optimal elements of Q. Therefore, by Theorem 1, it follows that P and Q have the same minimal $\langle A,O\rangle$-certificates, and thus $P \equiv^A_{|O} Q$.

The specific definitions of the above functions are as follows: $\mathfrak{c}^{\mathcal{C}}_A$ and $\mathfrak{p}^{\mathcal{C}}$ are functions defined in previous work about casting under hyperequivalence [6]. Specifically, $\mathfrak{c}^{\mathsf{N}}_A$, $\mathfrak{c}^{\mathsf{P}}_A$, and $\mathfrak{c}^{\mathsf{H}}_A$ are the *completion transformations* for normal, positive, and Horn programs, respectively [6, Definitions 4,7, and 9], while $\mathfrak{p}^{\mathsf{N}}$, $\mathfrak{p}^{\mathsf{P}}$, and $\mathfrak{p}^{\mathsf{H}}$ are the functions given by the adaptions in that work of the technique of *canonical programs* [2]. The functions $\mathfrak{f}^{\mathcal{C}}_{A,O}$, for $\mathcal{C} = \{\mathsf{N},\mathsf{P},\mathsf{H}\}$ are novel, however, and are briefly discussed in what follows. To begin with, $\mathfrak{f}^{\mathsf{N}}_{A,O}$ and $\mathfrak{f}^{\mathsf{H}}_{A,O}$ are defined as follows:

$$\mathfrak{f}^{\mathsf{N}}_{A,O}(P) = \{(X,Y) \mid (X,Y) \text{ is } \langle A,O\rangle\text{-optimal in } SE^A(P)\};$$
$$\mathfrak{f}^{\mathsf{H}}_{A,O}(P) = \{(X|_{A\cup O}, Y|_{A\cup O}) \mid (X,Y) \text{ is } \langle A,O\rangle\text{-optimal in } SE^A(P)\}.$$

The transformations reflect the fact that a normal casting can be build from atoms in P only and a Horn casting from atoms that occur in $atm(P)|_{A\cup O}$ only.

Obtaining the transformation $\mathfrak{f}^{\mathsf{P}}_{A,O}(P)$ for positive castings requires a more sophisticated approach, however, for which we first need to build a rooted forest $F_{A,O}(P)$ from the A-SE-models of P as an auxiliary structure. The nodes are triples (h, Y, Ξ), where h is an atom, unique for every node, that is from a set At' of auxiliary atoms not occurring in At, Y is an interpretation, and Ξ is a set of interpretations. Each node corresponds to an $\langle A, O\rangle$-certificate of the casting Q to build, and each root node to a minimal $\langle A, O\rangle$-certificate of P. The intuition is to preserve the minimal $\langle A, O\rangle$-certificates of P, for retaining $\langle A, O\rangle$-equivalence to P, and add missing (non-minimal) $\langle A, O\rangle$-certificates of Q that ensure A-here-totality of the A-SE-models of Q in order for Q to be a positive program. Roughly, for every $(X, Y) \in \mathfrak{f}^{\mathsf{P}}_{A,O}(P)$ with $X \subset Y$ that is built from a node N, a descendant of N guarantees that there is some $(X', X') \in \mathfrak{f}^{\mathsf{P}}_{A,O}(P)$ with $X'|_A = X$ and $X' \subset Y$, as required by A-here-totality. $F_{\langle A,O\rangle}(P)$ is obtained by the following algorithm, which starts with $F_{\langle A,O\rangle}(P)$ being empty:

1. For every (Y, Y) that is $\langle A, O\rangle$-optimal in $SE^A(P)$, add (h, Y, Ξ) as a root node of $F_{\langle A,O\rangle}(P)$, where $h \in At'$ is not occurring elsewhere in $F_{\langle A,O\rangle}(P)$ and $\Xi = \{X \mid (X, Y) \in SE^A(P), X \subset Y\}$.
2. While there is a leaf node (h, Y, Ξ) such that $\Xi \neq \emptyset$ do:
   - for each $X \in \Xi$, find some (U, U) that is $\langle A, O\rangle$-optimal in $SE^A(P)$,[3] where $U|_A = X$, $U|_O \subseteq Y|_O$, and $(V, U) \in SE^A(P)$ with $V \subset U$ implies $V \in \Xi$, and add $(h', U|_{A\cup O}, \Xi')$ as child node of (h, Y, Ξ), where $h \in At'$ does not occur elsewhere in $F_{\langle A,O\rangle}(P)$ and $\Xi' = \{V \mid (V, U) \in SE^A(P), V \subset U\}$.

Having $F_{\langle A,O\rangle}(P)$ built, we then obtain $\mathfrak{f}^{\mathsf{P}}_{\langle A,O\rangle}(P)$ as follows:

$$\begin{aligned}\mathfrak{f}^{\mathsf{P}}_{\langle A,O\rangle}(P) = \{(Y', Y') \mid & N = (h, Y, \Xi) \text{ is a node of } F_{\langle A,O\rangle}(P), Y' = Y \cup \{h\} \cup \\ & \{h' \mid (h', Z, \Xi') \text{ is ancestor or descendant of } N\}\} \cup \\ \{(X, Y') \mid & N = (h, Y, \Xi) \text{ is a node of } F_{\langle A,O\rangle}(P), X \in \Xi, Y' = Y \cup \\ & \{h\} \cup \{h' \mid (h', Z, \Xi') \text{ is ancestor or descendant of } N\}\}.\end{aligned}$$

When creating $\mathfrak{f}^{\mathsf{P}}_{A,O}(P)$ from $F_{A,O}(P)$, atoms from At' are added in a way such that A-completeness of the A-SE-models of Q is ensured, by making Y_1 and Y_2 incomparable for all $(X, Y_1), (X, Y_2) \in SE^A(Q)$ which are not associated to nodes on the same path in the forest. Nodes along the same path cannot violate A-completeness per se.

5 Discussion

We provided necessary and sufficient semantical conditions that hold for a program P and sets A, O of atoms iff there exists another program Q that is normal, positive, or Horn, respectively, such that P and Q are strongly equivalence relative to A under projection to O. These conditions are parameterised in A and O and defined on the minimal $\langle A, O\rangle$-certificates of P and, alternatively, on the A-SE-models of P. Furthermore, we showed that deciding whether a casting exists is not computationally harder than checking equivalence under the considered notion, and provided methods for constructing a

[3] The existence of such a (U, U) is guaranteed when $SE^A(P)$ is $\langle A, O\rangle_{\mathsf{P,s}}$-compliant.

casting whenever one exists. Our results contribute to the understanding of problem settings in logic programming in the sense that they show in what scenarios the usage of certain constructs are superfluous or not.

An interesting equivalence notion related to the one studied here is *modular equivalence*, introduced by Oikarinen and Janhunen [13] for the purpose of modular programming, accommodating the specification of input, output, and hidden atoms. Compared to the use of projection, where answer sets that coincide on the projected part are treated as identical, modular equivalence distinguishes between answer sets with differing hidden atoms and thereby enforces corresponding modules to have the same number of answer sets. *Update equivalence* [14], on the other hand, has no provision of projection but generalises relativised strong equivalence by allowing also the *deletion* of certain program parts during comparison.

A natural next step is to consider casting questions for nonground programs. Towards this end, recent work provides nonground versions of the kind of generalised equivalences studied here [15], yet relativisation quickly leads to undecidable instances.

References

1. Lifschitz, V., Pearce, D., Valverde, A.: Strongly equivalent logic programs. ACM Transactions on Computational Logic **2**(4) (2001) 526–541
2. Eiter, T., Tompits, H., Woltran, S.: On solution correspondences in answer set programming. In: IJCAI'05. (2005) 97–102
3. Woltran, S.: Characterizations for relativized notions of equivalence in answer set programming. In: JELIA'04. Volume 3229 of LNCS, Springer (2004) 161–173
4. Eiter, T., Fink, M., Tompits, H., Woltran, S.: Simplifying logic programs under uniform and strong equivalence. In: LPNMR'04. Volume 2923 of LNCS, Springer (2004) 87–99
5. Eiter, T., Fink, M., Tompits, H., Woltran, S.: On eliminating disjunctions in stable logic programming. In: KR'04, AAAI Press (2004) 447–458
6. Pührer, J., Tompits, H., Woltran, S.: Elimination of disjunction and negation in answer-set programs under hyperequivalence. In: ICLP'08. Volume 5366 of LNCS, Springer (2008) 561–575
7. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing **9** (1991) 365–385
8. Heyting, A.: Die formalen Regeln der intuitionistischen Logik. Sitzungsberichte, physikalisch-mathematische Klasse, preußische Akademie der Wissenschaften (1930)
9. Turner, H.: Strong equivalence made easy: Nested expressions and weight constraints. Theory and Practice of Logic Programming **3**(4-5) (2003) 602–622
10. Lin, F.: Reducing strong equivalence of logic programs to entailment in classical propositional logic. In: KR'02, Morgan Kaufmann (2002) 170–176
11. Eiter, T., Fink, M., Woltran, S.: Semantical characterizations and complexity of equivalences in answer set programming. ACM Transactions on Computational Logic **8**(3) (2007) 1–53
12. Woltran, S.: A common view on strong, uniform, and other notions of equivalence in answer-set programming. Theory and Practice of Logic Programming **8**(2) (2008) 217–234
13. Oikarinen, E., Janhunen, T.: Achieving compositionality of the stable model semantics for SMODELS programs. TPLP **8**(5-6) (2008) 717–761
14. Inoue, K., Sakama, C.: Equivalence of logic programs under updates. In: JELIA'04. Volume 3229 of LNCS, Springer (2004) 174–186
15. Oetsch, J., Tompits, H.: Program correspondence under the answer-set semantics: The nonground case. In: ICLP'08. Volume 5366 of LNCS, Springer (2008) 591–605