Equivalences in Answer-Set Programming by Countermodels in the Logic of Here-and-There

Michael Fink

Institut für Informationssysteme, Technische Universität Wien,
Vorortstraße 9-11, A-1040 Vienna, Austria
fink@kr.tuwien.ac.at

Abstract. In Answer-Set Programming different notions of equivalence, such as the prominent notions of strong and uniform equivalence, have been studied and characterized by various selections of models in the logic of Here-and-There (HT). For uniform equivalence however, correct characterizations in terms of HT-models can only be obtained for finite theories, respectively programs. In this paper, we show that a selection of countermodels in HT captures uniform equivalence also for infinite theories. This result is turned into coherent characterizations of the different notions of equivalence by countermodels, as well as by a mixture of HT-models and countermodels (so-called equivalence interpretations), which are lifted to first-order theories under a very general semantics given in terms of a quantified version of HT. We show that countermodels exhibit expedient properties like a simplified treatment of extended signatures, and provide further results for non-ground logic programs. In particular, uniform equivalence coincides under open and ordinary answer-set semantics, and for finite non-ground programs under these semantics, also the usual characterization of uniform equivalence in terms of maximal and total HT-models of the grounding is correct, even for infinite domains, when corresponding ground programs are infinite.

Keywords: answer-set programming, uniform equivalence, knowledge representation, program optimization.

1 Introduction

Logic programming under the answer-set semantics, called Answer-Set Programming (ASP), is a fundamental paradigm for nonmonotonic knowledge representation [1]. It is distinguished by a purely declarative semantics and efficient solvers [2,4,5]. Initially providing a semantics for rules with default negation in the body, the answer-set semantics [6] has been continually extended in terms of expressiveness, and recently the formalism has been lifted to a general answer-set semantics for first-order theories [7].

In a different line of research, the restriction to Herbrand domains for programs with variables, i.e., non-ground programs, has been relaxed in order to cope with open domains [8]. The open answer-set semantics has been further generalized by dropping the unique names assumption [9] for application settings where it does not apply, for instance, when combining ontologies with nonmonotonic rules [10].

As for a logical characterization of the answer-set semantics, the logic of Here-and-There (HT), a nonclassical logic extending intuitionistic logic, served as a basis.
Equilibrium Logic selects certain minimal HT-models for characterizing the answer-set semantics for propositional theories and programs. It has recently been extended to Quantified Equilibrium Logic (QEL) for first-order theories on the basis of a quantified version of Here-and-There (QHT) [11]. Equilibrium Logic serves as a viable formalism for the study of semantic comparisons of theories and programs, like different notions of equivalence [12,13,14,15,16]. The practical relevance of this research originates in program optimization tasks that rely on modifications that preserve certain properties [17,18,19].

In this paper, we contribute by tackling an open problem concerning uniform equivalence of propositional theories and programs. Intuitively, two propositional logic programs are uniformly equivalent if they have the same answer sets under the addition of an arbitrary set of atoms to both programs. As has been shown in [20], so-called UE-models, a selection of HT-models based on a maximality criterion, do not characterize uniform equivalence for infinite propositional programs. Moreover, uniform equivalence of infinite programs cannot be captured by any selection of HT-models [20], as this is the case, e.g., for strong equivalence.

While the problem might seem esoteric at a first glance, since infinite propositional programs are rarely dealt with in practice, it is relevant when turning to the non-ground setting, respectively first-order theories, where infinite domains, such as the natural numbers, are encountered in many application domains.

The main contributions can be summarized as follows:

– We show that uniform equivalence of possibly infinite propositional theories, and thus programs, can be characterized by certain countermodels in HT. However, HT is not ‘dual’ (wrt. the characterization of countermodels) in the following sense: The countermodels of a theory $\Gamma$ cannot be characterized by the models of a theory $\Gamma'$. Therefore, we also study equivalence interpretations, a mixture of models and countermodels of a theory, that can be characterized by a transformation of the theory if it is finite. We characterize classical equivalence, answer-set equivalence, strong equivalence, and uniform equivalence by appropriate selections of countermodels and equivalence interpretations.

– We lift these results to first-order theories by means of QHT, essentially introducing uniform equivalence for first-order theories under the most general form of answer-set semantics currently considered. We prove that, compared to QHT-models, countermodels allow for a simplified treatment of extended signatures.

– Finally, we show that the notion generalizes uniform equivalence for logic programs, and prove that it coincides for open and ordinary answer-set semantics. For finite non-ground programs under both ordinary and open answer-set semantics, we establish that uniform equivalence can be handled by the usual characterization in terms of HT-models of the grounding also for infinite domains.

Our results provide an elegant, uniform model-theoretic characterization of the different notions of equivalence considered in ASP. They generalize to first-order theories without finiteness restrictions, and are relevant for practical ASP systems that handle finite non-ground programs over infinite domains. For the sake of presentation, the technical content is split into two parts, discussing the propositional case first (Sections 2 and 3), and addressing first order theories and nonground programs in Sections 4 and 5.