Annals of Pure and Applied Logic 168 (2017) 234–253

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

Game semantics for non-monotonic intensional logic programming[☆], [☆☆], [☆☆☆]

Chrysida Galanaki[a,*], Christos Nomikos[b], Panos Rondogiannis[a]

[a] *Department of Informatics & Telecommunications, National and Kapodistrian University of Athens, Panepistimiopolis, 157 84 Athens, Greece*
[b] *Department of Computer Science and Engineering, University of Ioannina, P.O. Box 1186, 45 110 Ioannina, Greece*

ARTICLE INFO

Article history:
Available online 13 October 2016

MSC:
68N15
68N17
68Q55
91A05

Keywords:
Game semantics
Intensional logic programming
Non-monotonicity

ABSTRACT

Intensional logic programming is an extension of logic programming based on intensional logic, which includes as special cases both *temporal* and *modal* logic programming. In [13], M. Orgun and W.W. Wadge provided a general framework for capturing the semantics of intensional logic programming languages. They demonstrated that if the intensional operators of a language obey some simple semantic properties, then the programs of the language are guaranteed to have a *minimum model* semantics. One key property involved in the construction of [13] is the *monotonicity* of intensional operators. In this paper we consider intensional logic programming from a game-theoretic perspective. In particular we define a two-person game and demonstrate that it can be used in order to define a model for any given intensional program of the class introduced in [13]. Moreover, this model is shown to be identical to the minimum model constructed in [13]. More importantly, we demonstrate that the game is even applicable to intensional languages with *non-monotonic* operators. In this way we provide the first (to our knowledge) general framework for capturing the semantics of non-monotonic intensional logic programming.

[☆] This research was supported by the project "Handling Uncertainty in Data Intensive Applications", co-financed by the European Union (European Social Fund) and Greek national funds, through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) – Research Program: THALES, Investing in knowledge society through the European Social Fund.

[☆☆] The initial ideas of this paper were presented in GaLoP VII (Games for Logic and Programming Languages), Dubrovnik, Croatia, June 2012. A preliminary version of the paper has appeared as: *Ch. Galanaki, Ch. Nomikos and P. Rondogiannis: Game Semantics for Non-monotonic Intensional Logic Programming. LPNMR 2013: 329–341.*

[☆☆☆] Part of this work was conducted while the second author was visiting the National and Kapodistrian University of Athens.

[*] Corresponding author.
E-mail addresses: chrysida@di.uoa.gr (Ch. Galanaki), cnomikos@cs.uoi.gr (Ch. Nomikos), prondo@di.uoa.gr (P. Rondogiannis).

http://dx.doi.org/10.1016/j.apal.2016.10.005
0168-0072/© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Intensional Logic is an extension of classical logic that was introduced by R. Montague [9] in order to capture the semantics of natural languages. Roughly speaking, intensional logic was proposed as a formal system for understanding and reasoning about *context-dependent* properties of natural language expressions. In its initial form, intensional logic was a higher-order one, equipped with modal and temporal operators. Nowadays, however, the term "intensional logics" can also be used more loosely in order to describe a large class of logics for reasoning about context-dependent phenomena. In this sense, *temporal logics* and *modal logics* can be viewed as special cases of intensional logic.

Based on this broad interpretation of the term, M. Orgun and W.W. Wadge introduced in [13] the notion of *intensional logic programming*, which includes as special cases many non-classical extensions of logic programming (such as *temporal logic programming*, *modal logic programming*, and so on). As pointed out in [13], numerous logic programming languages that have been proposed in the literature can be characterized as "intensional" (such as Chronolog [13], Tempura [11], Molog [2], and so on). It was therefore natural to wonder whether there exists a common semantic framework for handling all these systems in a uniform way. As it was demonstrated in [13], if the intensional operators of the source intensional logic programming language obey some simple semantic properties, then the programs of the language are guaranteed to possess the *minimum model* property. However, all the intensional operators allowed in [13] are assumed to satisfy the *monotonicity* property (see Section 2 for a formal definition of this notion), and this excludes many interesting applications that involve *non-monotonicity* (which nowadays is a crucial concept involved in knowledge representation and reasoning).

The purpose of this paper is to extend the framework of [13] to allow arbitrary (even non-monotonic) intensional operators in the bodies of program clauses. To our knowledge, this is the first general semantic framework for non-monotonic intensional logic programming. Our construction is based on *game semantics for logic programming*, an approach which was initially proposed in [16] and has recently been revived and extended in [1,4,15]. In the game-theoretic approach, the semantics of a program is defined using the outcome of a two-person game. In contrast to the traditional semantics of logic programming, the game semantics has the important advantage that (once the basic notions involved are understood) it has a simplicity in the explanation and a quite intuitive flavor. In particular, as demonstrated in [4], the non-monotonic operator of *negation-as-failure* can be handled in a very natural way (see [4] for details). This simplicity in the treatment of negation motivated the present work, in which we consider a broader class of non-monotonic operators under the context of intensional logic programming. The main contributions of the paper are the following:

- We construct a simple two-person game for the class of intensional logic programs considered in [13]. We demonstrate that for any given program, we can use the outcome of the game in order to define a model of the program that is shown to coincide with the minimum model semantics obtained in [13]. In this way we provide an equivalent formulation to the approach of Orgun and Wadge for facilitating the further study of intensional logic programs.
- We extend the proposed game to handle intensional logic programs that even use non-monotonic operators in the bodies of clauses. In this way we obtain the first (to our knowledge) general semantic framework for non-monotonic intensional logic programming. It should be noted that intensional logic programming, due to its variety of operators, allows a much broader framework for non-monotonicity than the one we are familiar with from classical logic programming (where the main source of non-monotonicity is the operator of negation-as-failure).

The rest of the paper is organized as follows: Section 2 contains preliminary material regarding intensional logic programming and infinite games. Section 3 presents the proposed game for intensional logic programs that use monotonic operators. Section 4 extends the game for non-monotonic intensional logic programming.