

ID-logic in perspective

Marc Denecker
Joost Vennekens

Report CW410, 20 April 2005



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

ID-logic in perspective

Marc Denecker
Joost Vennekens

Report CW 410, 20 April 2005

Department of Computer Science, K.U.Leuven

ID-logic is an extension of classical logic with inductive definitions. This report is a brief introduction to the origins, the foundations and the motivations for this logic, and it discusses its very special place in the spectre of mathematical, computational and knowledge representation logics by pointing to its strong relationships to various logics in these fields.

Abstract

Keywords : Inductive Definitions, Logic Programming, Mathematical Logic, Knowledge Representation Logic, Computational Logic.

Inductive definitions are common in mathematical practice. For instance, in logic, the basic concepts of formulas and the satisfaction relation \models are defined inductively. In set theory, many concepts are defined by (transfinite) induction on a segment of ordinals. Mathematical structures such as Borel sets or subgroups are inductively defined as the closure of some initial subset under some operations. In mathematical texts, inductive definitions are usually represented as collections of basic and inductive rules. Inductive definitions in mathematics may be *monotone* or *non-monotone*. A typical example of a non-monotone inductive definition is that of the satisfaction relation \models as can be found in most textbooks on first-order logic. It is non-monotone due to its non-monotone rule:

$$I \models \neg\psi \text{ if } I \not\models \psi$$

which adds a pair $(I, \neg\psi)$ in absence of the pair (I, ψ) . This prevalence of inductive definitions indicates that these offer a natural, objective and well-understood way of representing knowledge. At the same time, inductive definitions cannot easily be expressed in classical logic. For instance, the transitive closure of a graph is one of the simplest concepts typically defined by induction—the relation is defined by two inductive rules: (a) if (x, y) is an edge of the graph then (x, y) belongs to the transitive closure; if (x, y) and (y, z) belong to the transitive closure, then (x, z) belongs to the transitive closure — yet it is well-known that this concept cannot be defined in first-order logic. While this concept can be defined in second-order logic (expressing that the transitive closure is the least relation closed under the inductive rules), the use of second order logic is somewhat of a burden, certainly from a computational point of view, and, moreover, even this methodology breaks down when faced with non-monotone inductive definitions, such as that of the satisfaction relation (indeed, \models is not the least relation closed under the standard inductive rules).

It turns out, however, that certain knowledge representation logics do allow a natural and uniform formalisation of the most common forms of inductive definitions. Recently, the authors of [4, 6] pointed out that semantical studies in the area of logic programming might contribute to a better understanding of such generalised forms of induction. In particular, it was argued that the well-founded semantics of logic programming [19] extends monotone induction and formalises and generalises non-monotone forms of induction such as induction over well-founded sets and iterated induction [12, 2]. In [8, 7], the well-founded semantics was further generalised into a fixpoint theory of general non-monotone lattice operators. This theory, called *approximation theory*, generalises Tarski’s theory of fixpoints of monotone lattice operators and provides the algebraic foundation of the principle of iterated induction. In [5, 11], these results were used to extend classical logic with non-monotone inductive definitions. The language of *ID-logic* uses the well-founded semantics to extend classical logic with a new “inductive definition” primitive. In the resulting formalism, all kinds of definitions regularly found in mathematical practice—e.g., monotone inductive definitions, non-monotone inductive definitions over a well-ordered set, and iterated inductive definitions—can be represented in a uniform way. Moreover, this representation neatly corresponds to the form such definitions

would take in a mathematical text. For instance, in ID-logic the transitive closure of a graph can be defined as:

$$\left\{ \begin{array}{l} \forall x, y \text{ TransCl}(x, y) \leftarrow \text{Edge}(x, y) \\ \forall x, y \text{ TransCl}(x, z) \leftarrow \text{TransCl}(x, y) \wedge \text{TransCl}(y, z). \end{array} \right\}$$

However, ID-logic is able to handle more than only mathematical concepts. Indeed, inductive definitions are also useful in common-sense reasoning. For instance, in [10], it was shown that situation calculus can be given a natural representation as an iterated inductive definition in the well-ordered set of situations. The resulting theory is able to correctly handle tricky issues such as recursive ramifications, and is in fact, to the best of our knowledge, the most general representation of this calculus to date. In general, definitions are a distinctive and important form of human expert knowledge; ID-logic is an appropriate knowledge representation language for this type of knowledge.

The generalised principle of inductive definition underlying ID-logic and formalised in approximation theory turns out to be fundamental to the area of Non-Monotonic Reasoning in general. It turned out that the principle is fundamental in the area of non-monotonic reasoning¹. In particular, [8, 9] demonstrated that the semantics of two other major approaches to non-monotonic reasoning, default logic [18] and autoepistemic logic [16] are described by approximation theory. Although these logics are definitely not inductive definition logics, a default or autoepistemic theory characterises a *revision operator*, which happens to be an approximation operator in the lattice of possible world collections. All common semantics of both logics and some new ones can be derived as different types of fixpoints of this revision operator. Thus, the generalised principle of inductive definition also plays a fundamental role in the semantics of these knowledge representation formalisms.

As regards ID-logic, this language has strong links to several families of mathematical, computational and KR-logics. As an extension of classical logic with inductive definitions, it can be classified in the family of description logics, which it extends by allowing definitions for n-ary predicates and non-monotone inductive definitions. As an extension of classical logic with a fixpoint semantics for inductive definitions, it can be viewed as a new element in the family of fixpoint logics. Monotone fixpoint logics have their origin in the logical study of monotone inductive definitions [17, 1]. The contribution of ID-logic is that it formalises two non-monotone inductive principles (i.e., inductive definition over a well-founded order and iterated inductive definition) which differ from the non-monotone principle based on the inflationary fixpoint which is commonly studied in fixpoint logic (IFP)[14].

ID-logic contributes to our understanding of Logic Programming and its variants such as Abductive Logic Programming and Datalog. Formally, ID-logic

¹The term “non-monotone” has a different meaning in the context of inductive definitions than in the context of non-monotone reasoning. A logic is non-monotone when adding formulas to a theory may not preserve inferred formulas. A monotone definition is one inducing a monotone operator. In fact, the fragment of monotone inductive definitions in ID-logic is a non-monotone logic.

extends Logic Programming under WFS; a normal logic program (with negation by failure, without disjunction and “strong negation”) can be viewed as a definition in ID-logic. In this view, Horn logic programs are viewed as monotone inductive definitions; programs with recursion over negation are viewed as iterated inductive definitions. As such, ID-logic induces an alternative epistemology for logic programming, solidly based on the mathematical principle of inductive definitions. This view is a refinement of the original view proposed by Clark [3] and induces a different informal semantics on logic programming than the currently more standard view endorsed in Answer Set Programming [13], in which program rules are viewed as *defaults*. In the latter view, negation as failure *not P* is seen as a form of negation by default *it can be assumed that $\neg P$* while the rule operator is (or is close to) material implication. A distinctive feature of the definitional view is that the rule operator reflects the conditional in an informal rule in a mathematical inductive definition, as in

$$I \models \neg F \text{ if } I \not\models F$$

which is not truth functional and definitely does not correspond to *material implication* while *negation as failure* is standard classical negation, as can be seen in the implicit use of negation in the condition of the above rule. In many applications of (Abductive) Logic Programming and Datalog, the view of a logic program as an inductive definition seems natural and appropriate in the sense that it correctly expresses the knowledge of the human expert. Even in many applications of Answer Set Programming, programs frequently encode inductive definitions together with classical logic axioms [15].

It appears from this that ID-logic and the principle of inductive definition formalised in it occupies a novel place in the spectrum of mathematical, computational and KR-logics. It offers a new epistemological perspective on a class of logics and points to close relationships between some of them. As such, the study of semantical and computational aspects of ID-logic can lead to synergy and integration of these different areas.

References

- [1] P. Aczel. An Introduction to Inductive Definitions. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 739–782. North-Holland Publishing Company, 1977.
- [2] W. Buchholz, S. Feferman, and W. Pohlers W. Sieg. *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies*, volume 897 of *Lecture Notes in Mathematics*. Springer-Verlag, 1981.
- [3] K.L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Databases*, pages 293–322. Plenum Press, 1978.
- [4] Marc Denecker. The well-founded semantics is the principle of inductive definition. In J. Dix, L. Fariñas del Cerro, and U. Furbach, editors, *Logics*

- in *Artificial Intelligence*, volume 1489 of *Lecture Notes in Artificial Intelligence*, pages 1–16, Schloss Dagstuhl, October 12-15 1998. Springer-Verlag.
- [5] Marc Denecker. Extending classical logic with inductive definitions. In J. Lloyd et al., editor, *First International Conference on Computational Logic (CL2000)*, volume 1861 of *Lecture Notes in Artificial Intelligence*, pages 703–717, London, July 2000. Springer.
- [6] Marc Denecker, Maurice Bruynooghe, and Victor Marek. Logic programming revisited: logic programs as inductive definitions. *ACM Transactions on Computational Logic*, 2(4):623–654, October 2001.
- [7] Marc Denecker, Victor Marek, and Mirek Truszczyński. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Information and Computation*, 192(1):84–121, 2004. URL = http://www.cs.kuleuven.ac.be/cgi-bin-dtai/publ_info.pl?id=41124.
- [8] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Approximating operators, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning. In J. Minker, editor, *Logic-based Artificial Intelligence*, chapter 6, pages 127–144. Kluwer Academic Publishers, Boston, 2000.
- [9] Marc Denecker, Victor Marek, and Mirosław Truszczyński. Uniform semantic treatment of default and autoepistemic logics. *Artificial Intelligence*, 143(1):79–122, 2003.
- [10] Marc Denecker and Eugenia Ternovska. Inductive Situation Calculus. In *Proceedings of Ninth International Conference on Principles of Knowledge Representation and Reasoning, Delta Whistler Resort, Canada*, pages 545–553, 2004. URL = http://www.cs.kuleuven.ac.be/cgi-bin-dtai/publ_info.pl?id=41085.
- [11] Marc Denecker and Eugenia Ternovska. A logic of non-monotone inductive definitions and its modularity properties. In Vladimir Lifschitz and Ilkka Niemelä, editors, *7th International Conference on Logic Programming and Nonmonotonic Reasoning*, 2004.
- [12] S. Feferman. Formal theories for transfinite iterations of generalised inductive definitions and some subsystems of analysis. In A. Kino, J. Myhill, and R.E. Vesley, editors, *Intuitionism and Proof theory*, pages 303–326. North Holland, 1970.
- [13] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–387, 1991.
- [14] Y. Gurevich and S. Shelah. Fixed-point Extensions of First-Order Logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.

- [15] Maarten Mariën, David Gilis, and Marc Denecker. On the relation between id-logic and answer set programming. In José Júlio Alferes and João Alexandre Leite, editors, *JELIA*, volume 3229 of *Lecture Notes in Computer Science*, pages 108–120. Springer, 2004.
- [16] R.C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, January 1985.
- [17] Y. N. Moschovakis. *Elementary Induction on Abstract Structures*. North-Holland Publishing Company, Amsterdam- New York, 1974.
- [18] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.
- [19] A. Van Gelder, K. A. Ross, and J.S. Schlipf. The Well-Founded Semantics for General Logic Programs. *Journal of the ACM*, 38(3):620–650, 1991.