Algebraic and Logical Study of Constructive Processes in Knowledge Representation *

Joost Vennekens
joost.vennekens@cs.kuleuven.be
Dept. Computerscience, KU. Leuven
Celestijnenlaan 200A, B-3001 Heverlee, Belgium

Constructive processes (i.e., derivations which gradually build up a model of the world) play an important role in the area of knowledge representation. We study such processes in an algebraic way, which proves to be a convenient way of deriving general theorems about different knowledge representation languages. Moreove, we also investigate the relation between constructive processes and causality. We observe that causal statements are essentially dynamic statements, which makes it natural to formalize their meaning in terms of constructive processes. Based on this observation, we develop the formal language of CP-logic as a new way of representing causal probabilistic knowledge.

Keywords: Knowledge representation, uncertainty in AI, logic programming

1. Introduction

The focus of this dissertation is on constructive processes in knowledge representation. In the area of non-monotonic reasoning, we find many logics whose semantics can be defined in terms of fixpoint constructs that simulate, e.g., human thought processes or actual physical processes.

2. Algebraic study of constructive processes

Many salient properties of constructive processes can be elegantly described in the framework of *approximation theory* [1]. This is an algebraic fixpoint theory, which naturally captures certain families of semantics for knowledge representation languages such as logic programming, default logic, and autoepistemic logic. The algebraic nature of this theory allows general results to be proven in a clean, compact, and elegant manner, without getting bogged down in syntactical details. Moreover, by proving a single theorem in approximation theory, one immediately obtains results for a number of different logics and, typically, also for an entire family of semantics for these logics. As such, the tedious task of reproving essentially the same property in slightly different circumstances can be avoided. In the first part of this dissertation, we study two important knowledge representation properties in approximation theory.

The first is that of *modularity* [3], i.e., we address the question of how to split a theory in such a way that the set of its components is, in some sense, equivalent to the theory as a whole. Such results are important for a variety of theoretical and practical purposes. For instance, they can be used to speed up computational tasks, to reduce the problem of analyzing a theory to the easier problem of analyzing its parts, or to show that updates of an existing knowledge base do not affect some already established property. We study this topic at the abstract, algebraic level of lattice operators in approximation theory and examine conditions under which it is possible to split such an operator into a number of smaller operators, while preserving certain of its fixpoints. We apply this result to logic programming, default logic, and autoepistemic logic, showing that, in each of these cases, our results extend existing theorems.

The second topic is that of *predicate introduction* [4], i.e., the common transformation of introducing a new symbol to abbreviate some complex formula. One might do this, for instance, to make a theory more readable and compact by singling out a subformula that appears in a number of different locations, or to transform it into some useful normal form. Once again, we study this topic at the level of approximation theory, by examining conditions under which it is possible to extend a

*Joost Vennekens is a postdoctoral researcher of FWO.

AI Communications
ISSN 0921-7126, IOS Press. All rights reserved

lattice operator to a larger lattice, in such a way that certain of its fixpoints are preserved. We then show how this result applies to autoepistemic logic and to logic programming, where it significantly generalizes existing results, by also allowing the new symbols to be defined recursively.

In both of these cases, approximation theory makes it possible to define natural algebraic counterparts to the relevant logical concepts, which then allows concrete results for different languages and different semantics to be derived in a uniform way. The emerging picture is one of approximation theory as a general, abstract framework for the study of knowledge representation properties of logics with a fixpoint semantics.

3. A causal probabilistic logic

The second part of this dissertation concerns the role of causality in probabilistic modelling. Here, too, constructive processes are important. Let us consider, for instance, the statement "smoking causes cancer." Intuitively, this means that the act of having a cigarette will initiate some sequence of events within the human body and that this process might eventually lead to the development of cancer. One of our central observations is that whenever we make such a causal statement, we are, at least implicitly, doing so in the context of the dynamic evolution of a domain. This motivates us to view such statements as describing the causes and effects of an *event*, of something that *happens*. We also consider events that are non-deterministic, which allows us to represent knowledge such as: "Smoking *might* cause cancer and the probability of this is 0.2." We then develop the knowledge representation language of *CP-logic*, in which a domain can be modeled by a set of such statements.

Our analysis of the intuitive meaning of these statements leads in a natural way to a formal semantics that is based on probabilistic processes. One of our key technical results is that, even though a single theory may generate any one of a number of different possible probabilistic processes, all of these will induce precisely the same probability distribution over their final states. Our study of causality in this explicitly dynamic context offers more insight into Pearl's work [2]. We identify the CP-event as a more basic unit of causal information than the parents-child relation of causal Bayesian networks. This yields a more flexible and fine-grained representation of causal events, which allows more straightforward and elaboration tolerant models of causal knowledge.

We also compare these causal probabilistic processes to processes used to define various fixpoint semantics for logic programs. To this end, we define a probabilistic extension of logic programs, called *logic programs with annotated disjunctions*, and prove that it is essentially equivalent to CP-logic. This result makes explicit the connection between causal reasoning and logic programming that has long been implicitly present in this field and shows that, under certain semantics, a rule of a normal or disjunctive logic program can be interpreted as a description of a causal event. Moreover, it also clarifies the position of our logic among related work on probabilistic logic programming.

4. Conclusion

In summary, the main contributions of this dissertation are as follows. First, we have shown that the methodology of studying constructive processes in the algebraic framework of approximation theory offers a viable way of deriving interesting properties of different logics in a general and uniform way. Second, we have also shown that constructive processes are important for a correct understanding of a certain kind of causal statements, because they offer a natural way of formalizing their inherent dynamic aspect. This dissertation is available at the following URL:

`http://hdl.handle.net/1979/849`

References

[1] Denecker, M., V. Marek, and M. Truszczyński. Approximating operators, stable operators, well-founded fixpoints and applications in non-monotonic reasoning. In *Logic-based Artificial Intelligence*, The Kluwer International Series in Engineering and Computer Science, pp. 127–144, 2000.

[2] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[3] Vennekens, J., D. Gilis, and M. Denecker. Splitting an operator: Algebraic modularity results for logics with fixpoints semantics. *ACM Transactions on Computational Logic* *7*(4), 765–797, 2006.

[4] Vennekens, J., M. Mariën, J. Wittocx, and M. Denecker. Predicate introduction for logics with a fixpoint semantics. *Fundamenta Informaticae* *79*(1-2), 187–227, 2007.