Perspective Switching Using Theories and Interpretations *

Abstract

William M. Farmer

The MITRE Corporation

An important aspect of intelligence is the ability to switch perspectives. In the course of reasoning about a particular subject matter, it can be convenient to change from one perspective of the subject matter to another perspective, which is often at a different level of abstraction. For example, in some situations it may be advantageous to view a building as a collection of floors and in other situations as a collection of wings. If the arrangement of the rooms is not important for the task at hand, the building might be viewed as simply an unstructured collection of rooms. These are three perspectives on how a building can be decomposed. Utilizing multiple perspectives and freeing moving from one to another is a powerful method for both managing complexity and organizing knowledge. The key idea behind this method is to always employ the "right" perspective for each reasoning task.

This talk will address the following two questions:

(1) What logical machinery is needed to formalize perspective switching?

(2) How can knowledge associated with one perspective be retrieved for use with another perspective?

A perspective can be formalized quite naturally as an *axiomatic theory* consisting of a *formal language* and a set of *axioms*. The language provides a vocabulary for talking about certain objects and their properties, while the axioms are assumptions about the objects and their properties. A structure (i.e., a domain of objects and properties) is a *model* of the theory if it satisfies the axioms of the theory. Logical derivation from the axioms of the theory yields the *theorems* of the theory. Each theorem holds in every model of theory.

If different perspectives can be represented as different theories, switching from one perspective to another is then just a matter of changing from one theory to another. However, it is not enough to just be able to change perspective: there needs to be a way of relating one perspective to another so that results (i.e., theorems and definitions) can be shared between different perspectives.

The logical device for relating one theory to another is the notion of a theory interpretation. An *interpretation* Φ of a theory T_1 in a theory T_2 is a certain kind of homomorphism from the expressions of T_1 to the expressions of T_2 which preserves theoremhood. Interpretations serve as conduits through which theorems and definitions can be freely "transported" from one theory to another. Since Φ preserves theoremhood, if A is a theorem of T_1, then $\Phi(A)$ is a theorem of T_2. $\Phi(A)$ is the same statement as A but seen from a different perspective.

Theories and interpretations are the building blocks of the "little theories method" in which a complex system or body of knowledge is described by a network of theories linked by interpretations. The little theories method is an old idea that has been used in mathematics since the late 1800s and has been advocated in recent years by a number of computer scientists for various purposes (e.g., see [1, 2, 5, 6, 7, 8]). The IMPS theorem proving system [3] supports the little theories method and contains a large database of traditional mathematics organized as a network of theories.

Suppose that an intelligent agent is trying to solve a goal G in a some theory T within a theory network. How can an intelligent agent find results *in other theories* that may be useful to solving the goal? One mechanism that can be used to answer this question is *translation matching* [4]. It is works as follows. A theorem A in a theory T' is matched in a syntactic

*Supported by the MITRE-Sponsored Research program. Presented at the 1996 International Multidisciplinary Conference on Intelligent Systems. Published in: J. Albus, A. Meystel, and R. Quintero, eds., *Intelligent Systems: A Semiotic Perspective*, pp. 206–207, National Institute of Standards and Technology, Gaithersburg, MD, October 20-23, 1996.

way against G yielding a partial mapping from the expressions of T' to the expressions T. The interpretations in the theory network are searched to find an interpretation Φ which is an extension of the partial mapping. If no such interpretation exists, the partial mapping is checked to see if it is itself an interpretation Φ. In either case, $\Phi(A)$ is presented to the agent as a theorem that is potentially useful to solving G.

Using translation matching, IMPS will present the user a well-pruned list of the theorems that can be applied to a given subgoal. In situations where 500 theorems or so are available to the user, there are rarely more than 10 theorems presented to the user as options. This mechanism has proven in IMPS to be a highly effective way to automatically retrieve relevant theorems distributed across a large network of theories.

Equipped with a knowledge base organized as a network of theories and with retrieval mechanisms based on translation matching, an intelligent agent could exploit perspective switching in a variety of ways. A difficult problem could be attacked by transporting it to a more convenient setting. One piece of abstract knowledge could be reused over and over again in various concrete situations. A theory for a particular application could be constructed by instantiating previously developed abstract reasoning machinery residing in the theory network. Certain symmetries could be formalized as interpretations of a theory into itself. A theory could be "recast" with new vocabulary and assumptions without forfeiting the results obtained in the original theory. And, finally, parts of the theory network could be developed independently and in parallel but be integrated with each other using appropriate interpretations.

References

[1] R. Burstall and J. Goguen. Putting theories together to make specifications. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 1045–1058, 1977.

[2] W. M. Farmer, J. D. Guttman, and F. J. Thayer. Little theories. In D. Kapur, editor, *Automated Deduction—CADE-11*, volume 607 of *Lecture Notes in Computer Science*, pages 567–581. Springer-Verlag, 1992.

[3] W. M. Farmer, J. D. Guttman, and F. J. Thayer. IMPS: An Interactive Mathematical Proof System. *Journal of Automated Reasoning*, 11:213–248, 1993.

[4] W. M. Farmer, J. D. Guttman, and F. J. Thayer. Contexts in mathematical reasoning and computation. *Journal of Symbolic Computation*, 19:201–216, 1995.

[5] R. Nakajima and T. Yuasa, editors. *The IOTA Programming System*, volume 160 of *Lecture Notes in Computer Science*. Springer-Verlag, 1982.

[6] D. Sannella and A. Tarlecki. Towards formal development of programs from algebraic specifications: Implementations revisited. *Acta Informatica*, 25:233–281, 1988.

[7] D. R. Smith and M. R. Lowry. Algorithmic theories and design tactics. *Science of Computer Programming*, 14:305–321, 1990.

[8] W. M. Turski and T. S. E. Maibaum. *The Specification of Computer Programs*. Addison-Wesley, 1987.