LOGIC PROGRAMMING APPLICATIONS FOR VISUAL KNOWLEDGE REPRESENTATION AND INTERPRETATION

TERRY SCHMIDT PASCHICH

Computer Science Department

APPROVED:

Michael Gelfond, Chair

Chitta Baral

Yi-Chieh Chang

Dean of the Graduate School

LOGIC PROGRAMMING APPLICATIONS FOR VISUAL KNOWLEDGE REPRESENTATION AND INTERPRETATION

by

TERRY SCHMIDT PASCHICH, B.A.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Computer Science Department

THE UNIVERSITY OF TEXAS AT EL PASO

December, 1991

ACKNOWLEDGEMENTS

I want to express my deepest appreciation to the chairman of my graduate committee, Dr. Michael Gelfond, for his consistent advice and encouragement (and especially for his seemingly unending patience) during the course of this work. I also want to extend my thanks to Dr. Chitta Baral and Dr. Yi-Chieh Chang for serving in my committee.

ABSTRACT

The application of strongly introspective theories to formalization of reasoning about computational vision problems is investigated. Strongly introspective epistemic theories are used as the specification language for representing and interpreting visual knowledge. The construction of an epistemic specification is viewed as a stepwise expansion of an epistemic theory T_0 by disjunctive definitions $D_1, \ldots, D_n$ of new predicates. The interpretation of an image is defined as the world view of an epistemic theory representing the deductive knowledge base of the image domain knowledge, scene domain knowledge, and mapping relations between the two domains. An algorithm is presented that constructs the unique world view of epistemic theory $T_0 + D$, where T_0 is an epistemic specification with one world view and D is a disjunctive definition of new predicates w.r.t. theory T_0. Application of the algorithm is demonstrated for simple vision problems. This work is compared to that of Reiter, in which classical logic is used to specify the knowledge base for image depiction and interpretation.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

Chapter 1

INTRODUCTION

Knowledge representation is one of the most important fields of research in artificial intelligence. It is fundamental to any computational pursuit. How information is symbolically represented has profound effect on the mechanization of problem solving. In recent years, there has been considerable research into the study of the applicability of logic programming based formalisms to knowledge representation. (For a review, see [Kowalski 90], [Przymusinska and Przymusinski 90], [Przymusinski 89].) This thesis examines the use of logic programming for representing and interpreting visual knowledge. The primary emphasis of this research focuses on the use of logic programming as a specification language for computational vision problems.

[Reiter and Mackworth 89] examines the use of classical logic as a foundation for addressing the problem of depiction and image interpretation in a visual knowledge representation system. This research paper was the first to present a precise definition of the concept of an interpretation of an image. This was a very important development, since without such a definition, there is no way to determine whether a computational vision system is sound or complete. One of the major purposes of this thesis is to provide a definition of image interpretation in logic programming.

This thesis hypothesizes that logic programming offers significant advantages over classical logic as a specification language for computational vision problems. One advantage of using logic programming as a specification language is that, in an ideal logic programming system, new information is added to a knowledge representation system simply by adding new rules or new facts to the system. With

classical logic, existing statements often have to be deleted and then replaced. It is also easier to handle exceptional conditions with logic programming. This thesis investigates the premise that logic programming allows an easier and more natural approach to the mechanization of computational vision problems than is possible with classical logic.

1.1. Knowledge Representation and Logic Programming

There are many logic programming based formalisms for knowledge representation. The development of knowledge systems that support this thesis require a formalism which provides sufficient flexibility for depicting and reasoning about computational vision problems. It would be convenient to use Prolog as the specification language for visual knowledge systems, since it is also a good logic programming implementation language for a large class of problems. However, Prolog has neither sufficient power nor expressiveness for this task. In particular, negative and disjunctive statements are required. The representation of theories with classical negation or with rules containing disjunctive conclusions is beyond the capability of Prolog programs. A more expressive form of logic programming is needed.

The language of strongly introspective epistemic theories developed by Michael Gelfond [Gelfond 91] was selected for the specification of computational vision problems in this thesis. This language contains the basic features of Prolog, but also allows representation of both negative and disjunctive statements. It differs from the language of classical logic in that it contains a new set of logical connectives. Theories developed in this language, i.e. epistemic theories, can express both the notion of classical negation and negation as failure of Prolog. They also handle disjunctive information in a fairly natural manner. Knowledge represented in an

epistemic theory can be reasoned about by mechanically constructing sets of beliefs from the theory. Epistemic knowledge systems and belief sets can also be used for explaining observations.

The framework proposed in [Reiter and Mackworth 89] for specifying a knowledge base for depiction and image interpretation was constructed in classical first-order logic. It provided the foundation for the development of a methodology for interpreting simple scenes from hand drawn (sketch) maps of geographical regions. To illustrate the use of logic programming for visual knowledge representation, this research, like that of Reiter and Mackworth, focuses on the depiction and image interpretation of simple maps. However, the methodology of the two approaches differs considerably.

Reiter and Mackworth used classical logic to define the notion of an image interpretation. First, the knowledge domain of a simple map image was divided into two distinct areas:

1. An image domain, which defines and describes a hand-drawn sketch map.
2. A scene domain, which describes real-world geographic objects that are mapped to objects described in the image domain.

Using classical first-order logic, image, scene and mapping axioms were developed. These axioms were then used to express both the graphical knowledge from a sketch map and the underlying assumptions about maps in general. Using this methodology, an interpretation of a map is a model of image, scene and mapping axioms.

The research in this thesis also begins with image and scene domains and mapping relationships between the two. But rather than using classical logic as the specification language, logic programming with strongly introspective epistemic

specifications is used. Epistemic theories are built which represent the deductive knowledge base described by simple maps. An interpretation of a map described by an epistemic theory is a belief set of the theory. It will be shown that there is a natural correspondence between this notion of interpretation and that described by Reiter and Mackworth. Examples given will illustrate that both formalisms produce the same interpretations.

Other researchers are also investigating the field of depiction and image interpretation originally studied by Reiter and Mackworth. [Poole 90] discusses the use of a system that uses a simple form of hypothetical reasoning for solving computational vision problems. Although also based on classical logic, Poole's methodology differs from both Reiter and Mackworth's and that described in this thesis. His methodology requires background knowledge about vision problems to be provided by statements of facts, defaults, and conjectures. Specific knowledge about a particular image is added as observations. The system then abduces explanations of the observations. Poole used this approach to provide explanations of the sketch map problem of [Reiter and Mackworth 89]. He considered Reiter and Mackworth's image-scene domain mapping axioms and general map knowledge axioms to be facts. Conjectures were open formulas for objects of the scene domain. Observations described the image domain of the specific sketch map. Poole demonstrated that the hypothetical reasoning system generated explanations that corresponded exactly to the interpretations of Reiter and Mackworth. But he left open the question of which of the two formalizms is better able to be expanded into more complicated and realistic domains.

1.2. Organization of Thesis

Chapter 2 reviews the basic definitions of epistemic logic as developed in [Gelfond 91]. It also presents an algorithm for mechanically constructing world views of epistemic specifications.

Chapter 3 defines the notion of an interpretation of a map described by an epistemic theory and provides an illustrative example of the construction of the epistemic specification and image interpretation of a simple sketch map.

Chapter 4 describes examples of constructing epistemic specifications for images depicting incompletely described scenes. This chapter shows how epistemic logic can model concepts (such as ambiguity) that cannot be easily handled by classical logic.

Chapter 5 provides recommendations and suggestions for further research.

Chapter 6 concludes with a discussion of some of the advantages and disadvantages of using strongly introspective epistemic specifications for knowledge representation.

Chapter 2

EPISTEMIC SPECIFICATIONS

The language of epistemic theories is used in this thesis as the formalism for visual knowledge representation. One of its major advantages over other logic programming based formalisms for knowledge representation is that epistemic specifications are designed to allow reasoning about worlds with incomplete information. They allow expression of the notion of uncertainty, as well as the existence of default knowledge and the concepts of having knowledge that may be believed and knowledge that is known to be true.

The notion of epistemic specification was introduced in [Gelfond 91]. The remainder of this paper uses the extended language $\mathcal{L}$ introduced there. Section 2.1 starts by reviewing basic definitions from this paper.

2.1 Definitions

Consider a language $\mathcal{L}_0$ consisting of predicate symbols $p, q, \ldots$, object variables, function symbols, connectives &, $\neg$, $\exists$, and the modal operator K. Terms and formulas of $\mathcal{L}_0$ are defined in the usual way. Formulas of the form $p(t)$ where t is a term are called atoms. Literals are atoms and their strong negations (e.g. $p(t)$, $\neg p(t)$). Literals not containing variables will be called ground. The set of all ground literals is denoted by *Lit*.

Consider a collection $A = \{A_i\}$ of sets of ground objective literals and a set W of such literals. (A can be thought of as a collection of possible belief sets of a reasoner while W represents his current (working) set of beliefs.) The notion of truth ($\models$) and falsity ($=\!|$) of formulas of $\mathcal{L}_0$ w.r.t. a pair $M =< A, W >$ is defined

inductively:

Definition 1

$M \models p(a)$ iff $p(a) \in W$.

$M \models \mathrm{K}F$ iff $< A, A_k > \models F$ for every A_k from A

$M \models F\&G$ iff $M \models F$ and $M \models G$

$M \models \exists x F$ iff there is a ground term t such that $M \models F(t)$

$M \models \neg F$ iff $M =\!\mid F$

$M =\!\mid p(a)$ iff $\neg p(a) \in W$

$M =\!\mid \mathrm{K}F$ iff $< A, A_k > =\!\mid F$ for some A_k from A

$M =\!\mid F\&G$ iff $M =\!\mid F$ or $M =\!\mid G$

$M =\!\mid \exists x F$ iff for every ground term t, $M =\!\mid F(t)$

$M =\!\mid \neg F$ iff $M \models F$

Let language $\mathcal{L}_0$ be expanded by the connectives *or* and $\forall$ and a modal operator M (where MF is read as "F may be believed") defined as follows:

MF iff $\neg\mathrm{K}\neg F$

F *or* G iff $\neg(\neg F \& \neg G)$

$\forall x F$ iff $\neg\exists x \neg F$

Formulas of the expanded language $\mathcal{L}$ not containing modal operators are called objective formulas while those starting with K or M are called subjective.

It is easy to see that according to the definition above, the truth of subjective sentences does not depend on W while the truth of objective ones does not depend on A, i.e., there is a notion of objective formula being true (false) in W and subjective formula being true (false) in A. The former is denoted by $W \models F$ ($W =\!\mid F$)