This article has been accepted for publication in a future issue of a journal/proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: Belli, F., Jack, O., Schmidt, A., Object-Oriented Extension of PROLOG - Considered Harmful for First-Order Predicate Logic, Tagungsband GWAI-Workshop ``Logik als Programmiersprache"/GWAI '90, pp. 70-72

Object-Oriented Extension of PROLOG Considered harmful for First-Order Predicate Logic?

F. Belli, O. Jack

University of Paderborn

1. Motivation for the logic and object-oriented progamming multi-paradigm

In recent years the logic and object-oriented programming have attracted growing interest. Both offer advantages over traditional programming methods.

Logic programming languages like PROLOG have a built-in mechanism for backtracking and pattern matching via unification. This provides a natural way for deductive retrieval and for presentation of rule-based knowledge. Because of its declarative style, PROLOG is not, however, well suited for tasks of explicitly procedural nature.

Object-oriented programming takes the advantages of traditional progamming into account, considering the modern software engineering, and provides a natural way to solve problems, in which data objects can be categorized hierarchically. Data encapsulation and inheritance encourage a structured implementation style and enhance the maintainability of software. The defiency of object-oriented programming is that it provides no generalized methods for deductive retrieval and pattern matching, which are basic methods of knowledge-based applications. For these reasons it is only natural to combine both, logic and object-oriented programming, to enable the use of all of their advantages.

2. Theses on object-orientation and first-order predicate logic

a) Object-orientation is not immediately compatible with first-order predicate logic.

The notions of class and instance seem not to be expressible in first-order predicate logic. Higher-order predicate logic may be needed for this purpose.

The notion of inheritance means that an object inherits methods and variables from its fatherclass. Moreover, inherited methods can be redefined by the object too. The consequence is that the functional behavior of the method is not unambiguously determined by the method name. The functional behavior of a method depends also on the object, for which the method is invoked. Even the number of arguments proceeding the method name depends on the objects, for which the method should be activated.

It is not very likely that this property can be described using "pure" first-order predicate logic.

b) Object-orientation based on logic progamming leads to a "contamination" of first-order predicate logic.

c) The "contamination" of first-order predicate logic, which is needed to achieve object-orientation, is contained in the "contamination" which already took place in PROLOG.

Prolog is not a pure first-order predicate logic:

(i) PROLOG is restricted to Horn-Clauses. So we can only express "positive facts", i.e. we can express what is true, but not, what is false. This is a restriction of full first-order predicate logic.

(ii) PROLOG includes some built-in "predicates" such as CUT which may cause that no SLD-Refutation to a goal will be found, although there exists one. PROLOG includes the "predicates" ASSERTA and ASSERTZ, which change the knowledge-base (i.e. the logic program) dynamically. These procedural aspects of PROLOG do not exist in first-order predicate logic.

We try to achieve object-orientation based on logic programming with a minimum of this "contamination". To determine the elements, which must be added to first-order predicate logic, we proceed as follows:

- precise definition of object-orientation.

  Nowadays the notion of object-orientation is often informal. Because of the informal notion there are various interpretations of object-orientation. This leads to problems in the "adoption" of object-orientation in other programming paradigms.

- consideration of this precise definition in first-order predicate logic and PROLOG.

- uniform representation of object-orientation in PROLOG (as "contaminated" first-order predicate logic).