WHY NOVICES WILL FIND LEARNING PROLOG HARD

Josie Taylor

Cognitive Science Research Paper

Serial No. CSRP. 043

The University of Sussex
Cognitive Studies Programme
School of Social Sciences
Falmer
Brighton BN1 9QN

WHY NOVICES WILL FIND LEARNING PROLOG HARD

ABSTRACT

This paper discusses the belief that logic programming languages, including PROLOG, are easy to learn because of the relationship purported to hold between predicate logic and natural language. By reference to the literature, it is pointed out that logically unsophisticated beginners are prone to confuse natural language meaning with logical form, which leads them into errors of judgement regarding truth or validity. Encouraging novice programmers to view logic programming in terms of its natural language equivalents is undesirable because it may mislead them into complacency, and an over-reliance on natural language interpretation which they do not have the sophistication to constrain.

Josie Taylor,
Cognitive Studies Programme,
Graduate Division,
Graduate School, Arts D,
University of Sussex,
Falmer, Brighton.

This paper examines the belief that logic programming languages are a priori easier formalisms to learn and use than other programming formalisms. This belief appears to stem from assumptions about the relationship between first-order predicate logic and natural language. It is not our intention to attack logic programming, nor its only currently available representative, PROLOG. Neither do we wish to imply that logic programming is not a useful formalism.

We are concerned instead with how logically-naive novice programmers might learn to program in PROLOG, what causes them problems, and why. As a preliminary step in this enterprise, a literature survey was conducted. The survey encompassed the fields of human factors in programming, the psychological studies of logical deductive reasoning, and the arguments in support of logic programming by its major proponents. A significant discrepancy was noted between hypotheses generated from the literature on deductive reasoning, and the apparent claims of logic programming devotees. This discrepancy is the subject of discussion here.

Logic programming languages - including PROLOG - are higher-level than other currently used programming languages. As Kowalski (1973) points out, predicate logic as a programming language has machine-independent syntax and semantics. These semantics are well understood in terms of classical logic. Liberation from machine constraints is partly what makes logic programming higher-level than other languages. Furthermore, the specification language is (at least in theory) the programming language. This streamlines the process of program development by eliminating successive translations from one representation to another, and allows for more effective deployment of programmer time and effort.

In very practical terms, we can see the advantages that logic programming languages have for the programmer. However, the line of argument with which we take issue is the one propounded by Kowalski, which broadly speaking says that: since predicate calculus is a formalisation of the logic of natural language and rational human thinking processes, logic programming is more 'human-oriented' than other programming formalisms, is thereby more 'natural' and, furthermore, can be understood in terms of its natural language equivalents (Kowalski, 1979).

This argument supports both a weak and a strong interpretation. Whether or not Kowalski himself would stand by both, it seems that both have been taken up by workers in the field of Artificial Intelligence, and this has led to the belief we wish to undermine. The weak interpretation takes the above argument in the context in which it was originally proposed - i.e. 'human-oriented' is the opposite of 'machine-oriented', and knowing how rigid some machine-oriented languages can be to work with, the epithet is acceptable. Again, familiarity with older, lower-level programming languages makes meaningful the remark about understanding programs in terms of their natural language equivalents.

However, the following quotations are to be found in a book published in 1979:

> 'The meaning of programs expressed in conventional languages is defined in terms of the behaviour they invoke within the computer. The meaning of programs expressed in logic...can be defined in machine-independent, human-oriented terms. As a consequence, logic programs are easier to construct, easier to understand, easier to improve, and easier to adapt to other purposes.' Kowalski (1979, Preface)

and further on:

> '...when programs are expressed in symbolic logic, they can be

> understood in terms of their human-oriented, natural language equivalents.' Kowalski (1979, p9)

This book assumes no previous knowledge of logic, problem-solving, or computer programming, and so we must assume that the remarks are aimed not at workers within the field, but at beginners. If such beginners have no appreciation of where Kowalski's argument has developed from, how are they to interpret these remarks? It is our contention that, decontextualised in this manner, the comments encourage a much stronger interpretation than that described above. The strong interpretation says that logic programming is more natural than any other programming formalism because it lies closer to natural human thought processes. This may well lead the naive beginner to assume that logic programming is easy, and that his 'natural' language skills will be of use to him in the enterprise. Such an interpretation is clearly misguided. But apart from the general undesirability of having one's argument misinterpreted, what is so problematic about beginners adopting this strong view?

We believe that accepting the strong interpretation leads the novice programmer into false expectations of what the process of programming is about, and what he can expect of his own performance. The reason this is so critical is that the very particular assumption he is encouraged to make - that logic can be understood in terms of its natural language equivalents - is noted in the psychological literature as the biggest obstacle to effective deductive reasoning.

The significant difference between the weak and the strong interpretation is that, in the strong interpretation, the separation between the formalism of classical logic as devised by logicians, and the hypothesised 'logical' thinking processes used by people in their everyday lives, is blurred. This is unfortunate, since psychologists

are uncertain about whether a 'natural' logic exists, let alone what kind of classical logic it might resemble (see Johnson-Laird, 1983).

Classical logic (of whatever type) is a formalism, and formalisms are designed to be unequivocal. In both the fields of human factors and the psychology of deductive reasoning, we are looking at people struggling to solve problems within the framework of a formalism which, in the interests of maintaining its formality, embodies certain necessary, but counter-intuitive, constraints. People find such tasks difficult. In trying to cope with counter-intuitive notions, the beginner is liable to look to inappropriate sources for help. One such source is natural language semantics. To actively encourage the beginner to refer to natural language as a means of understanding logic seems a dubious enterprise.

Kowalski (1979) makes the caveat that natural language will only provide an 'informal guide' to understanding logic. This is all very well for someone who understands either logic, or programming, or both. He or she will have a fundamental grasp of the limitations of the 'informal guide' as an aid to understanding. But the warning is not adequate for the optimistic novice who has taken the strong interpretation.

The deductive reasoning literature points to the prediliction people have to misinterpret logical expressions ad libertum.

This misinterpretation may arise from the nature of the material to be analysed. The dilemma is this: if the material to be reasoned about is abstract, then the task is often difficult; if concrete material is substituted for abstract (e.g. natural language expressions are substituted for symbolic expressions) then task difficulty diminishes greatly, but the reasoning process becomes error-prone. The errors stem

from a failure to appreciate the scope and meaning of natural language in the logical domain, contributing to the unwarranted interpolation of real-world knowledge in the formal domain.

The beginner often finds it difficult to disengage his beliefs and views from assertions in natural language. The most extreme example of this is when the material is contentious. Subjects have a tendency to assert that what they consider to be desirable is true, and what they consider to be true is valid (Lefford, 1946). Alternatively, they may be led astray by the 'atmosphere effect' (Begg and Denny, 1969). Assertions couched in natural language create an atmosphere which, in the absence of rigorous logical analysis, may seduce people into deriving conclusions which favour that atmosphere (for example, an inclination towards a positive conclusion rather than a negative). These examples argue against people referring to some kind of innate logic.

Chapman and Chapman (1959) postulate that subjects do reason logically, but that they may undertake illicit conversion of premises, so appearing to reason incorrectly. Henle (1962) agrees, pointing out that subjects may refuse to accept, or may misinterpret, premises.

We are all aware of the elasticity of natural language, and the skill with which we can make it mean what we want it to, according to our individual needs and wants. The conversions described above are examples of this. Compounding this tendency are the disparities between logic and natural language. For example, logical propositions have one of two truth values - they are either true or false - which is not the case in natural language. Connectives are defined solely as functions of the truth values of the propositions they interrelate; logic does not deal with temporal or causal events. (See Johnson-Laird and Wason, 1977).

Furthermore, there are valid inferences in predicate logic for which the English equivalent is not valid.

For example:

```
            (P -> q) -> r
            ~P
            =============
  Therefore r
```

is a valid argument. However:

```
       If I have eternal life if I believe in God
          then God exists.
       It is not the case that I believe in God.
    So God exists.
```

is evidently not valid (Staines, 1984).

The errors resulting from over-reliance on natural language interpretation in programming will be particularly apparent in database manipulatjon tasks. Ross (1982) points out that beginners are usually presented the declarative viewpoint of PROLOG angled towards database creation and search. His opinion is that this approach promotes complacency. The point is that PROLOG is almost too easy at this level. Beginners can fool themselves that they are learning to program, when in fact they are reading far too much significance into the information they are dealing with - far more than the machine is. The disparity between what the student thinks her program is doing (described at a linguistic level), and what it is actually doing (at the logical level), makes debugging difficult. For example, we have noted that, when asked to describe what their PROLOG programs were supposed to do, some students at Sussex were able to give quite competent English descriptions. But the code they had written did not work in the desired way because the logical structure of the problem solution had not been extracted from the English. The solution, as expressed in these

programs, lay in understanding the meaning of the English words, which of course PROLOG cannot do.

CONCLUSION

Kowalski's arguments in support of logic programming are open to misconstrual by novices because the distinction between formal logic and the 'natural' logic which may underly human deductive reasoning is blurred. The beginner is not in a position to understand the context of remarks about the 'human-oriented' aspect of logic programming language, and may be encouraged to rely too heavily on natural language semantics to understand logic.

The tendency to confuse meaning with form in logic has been identified in the psychological literature, and has been shown to have detrimental effects on people's ability to judge validity or truth. The implication that logic is more 'understandable' because of its relation to natural language is misleading.

In the initial stages of learning, we think that novices should be discouraged from viewing logic programming in terms of natural language at least until they have a sufficient grasp of the nature of logic to understand the limitations of its natural language equivalents.

REFERENCES

BEGG, I, & DENNY, J.P. (1969) 'Empirical reconsideration of atmosphere and conversion interpretations of syllogistic reasoning errors', Journal of Exp. Psychology, 81, 351-354.

CHAPMAN, L.J. & CHAPMAN, J.P., (1959) 'Atmosphere effect re-examined', Journal of Exp. Psychology, 58, 220-226.

HENLE, M., (1962) 'On the relation between logic and thinking', Psychological Review, 69, 366-378.

JOHNSON-LAIRD, P.N., (1983) Mental Models Cambridge University Press, Cambridge, U.K.

JOHNSON-LAIRD, P.N. & WASON, P.C. (eds.) (1977) Thinking - Readings in Cognitive Science Cambridge University Press.

KOWALSKI, R. (1973) 'Predicate logic as programming language', Memo No. 70, Department of Computational Logic, School of AI, University of Edinburgh.

KOWALSKI, R. (1979) Logic for Problem Solving North Holland Inc.

LEFFORD, A., (1946) 'The influence of emotional subject matter on logical reasoning', Journal of General Psychology, 34, pp. 127-51.

ROSS, P. (1982) 'Teaching Prolog to undergraduates', in AISBQ Autumn Issue 45

STAINES, P. (1984) 'Logical links and reliable knowledge representation' CSRP No. 41, Sussex University, Brighton, U.K.