# Foundations Proposal

*Roger Bishop Jones*

ICL Defence Systems

*ABSTRACT*

A proposal for research work in logical foundations for Computer Science.

## 1. INTRODUCTION

Support is sought for research into new logical foundations for the application of formal methodologies in Computer Science.

## 2. OUTLINE OF PROPOSED WORK

Currently established logical foundations for mathematics are not wholly satisfactory for application in Computer Science. New ideas have been emerging over the last decade on how to construct foundations more appropriate to Computer Science. An approach has been developed at ICL which appears to promise significant improvements over existing foundations. Further essentially theoretical work is thought to be necessary to establish the logical foundations prior to undertaking the tool building programs necessary for the full exploitation of these ideas.

This consists of:

1. careful formal specification of the primitive formal systems and the establishment of important properties such as consistency.
2. the definition of a rich type theory using the primitive formal system.
3. design of the architectural aspects of formal methods environments to exploit the foundations.
4. assessment of the costs and timescales necessary to complete the development of environments.

## 3. SCALE AND SCOPE

To be undertaken by ICL as principle with (at least) 1 academic collaborator, either as consultant or subcontractor. Total effort to be expended: 2 man years over a period of 18 months.

## 4. CURRENT STATUS OF WORK

Two papers have been produced giving accounts of the basic ideas underlying the proposal. A significant amount of further detailed technical development is now required to establish more solidly the viability of the proposal, and to provide a basis for development of computer based

support environments for formal developments exploiting these ideas. For this purpose academic support in the areas of combinatory logic and recursive function theory is highly desirable.

## 5. MOTIVATION

This proposal is motivated by the need for higher standards of computer based support for the exploitation of formal methods in computer systems development. One factor inhibiting the development of satisfactory support environments is a perceived mismatch between the foundation systems which have been developed for mathematics, and the needs of computer science. The intention of this proposed work is therefore to provide logical foundations which fully reflect the needs of computer science. These new logical foundations will then be exploited in tool developments which provide support for the use of formal methods in computer system development, enabling formal proof of the properties of the systems under development where this required.

## 6. MOTIVATION

The detailed technical motivation for seeking new foundations comes from a wide variety of considerations. To fully explain these motivations is not possible in a brief proposal, but we mention here some of the factors involved:

*Abstraction*

The classical set theoretic heirarchy inhibits the full exploitation of abstraction by preventing abstraction over the entire universe. This significantly impairs the power and flexibility of specification languages with essentially set theoretic semantics. Abandoning classical mathematics in favour of intuitionist mathematics does not improve this position. Our proposed foundations are essentially unlike classical systems in permitting abstraction and quantification over the entire universe (the universe is not hierarchic).

*Polymorphism*

Many programs in practise are indifferent to certain aspects of the structure of the data they manipulate. Thus a sort operation cares only about the type of the parts of the data on which the sort is being made, and if parametrised by the ordering relation is indifferent to the type of the ordering field as well. While polymorphism in various degrees has been introduced into a number of typed programming languages and logics, none of these systems truly reflects the flexibility of type free programming languages.

*Persistence*

The use of strictly typed programming languages for implementing systems software is not currently feasible because of lack of flexibility in the type systems currently available. The use of typed programming languages normally takes place in an untyped environment through which type conversions not legal within the language may take place (notably data to program and vice-versa).

*Assurance*

The need for maximal confidence in the soundness of our logical foundations suggests that logical foundations should be kept simple, even though for practical reasons working languages need to be very rich and hence complex. It is therefore desirable to separate the logical foundations from the detailed syntax and semantics

of practical specification langauges in such a way that gives us productive languages based on firm logical foundations.

*Reflexiveness*

In practise complete computer systems are often *reflexive* or self modifying. To be able properly to deal with proofs about such systems foundations are desirable which admit self application of functions.

*Generality*