1 List of Deliverables

LOMAPS-CAMCL-4 P.N. Benton: *A Mixed Linear and Non-linear Logic: Proofs, Terms and Models* Intuitionistic linear logic regains the expressive power of intuitionistic logic through the ! ('of course') modality. Benton, Bierman, Hyland and de Paiva have given a term assignment system for ILL and an associated notion of categorical model in which the ! modality is modelled by a comonad satisfying certain extra conditions. Ordinary intuitionistic logic is then modelled in a cartesian closed category which arises as a full subcategory of the category of coalgebras for the comonad.

This paper attempts to explain the connection between ILL and IL more directly and symmetrically by giving a logic, term calculus and categorical model for a system in which the linear and non-linear worlds exist on an equal footing, with operations allowing one to pass in both directions. We start from the categorical model of ILL given by Benton, Bierman, Hyland and de Paiva and show that this is equivalent to having a symmetric monoidal adjunction between a symmetric monoidal closed category and a cartesian closed category. We then derive both a sequent calculus and a natural deduction presentation of the logic corresponding to the new notion of model.

LOMAPS-CAMCL-5 P.N. Benton, G.M. Bierman and V.C.V de Paiva: *Computational Types from a Logical Perspective I* Moggi's computational lambda calculus is a metalanguage for denotational semantics which arose from the observation that many different notions of computation have the categorical structure of a strong monad on a cartesian closed category. In this paper we show that the computational lambda calculus also arises naturally as the term calculus corresponding (by the Curry-Howard correspondence) to a novel intuitionistic modal propositional logic. We give natural deduction, sequent calculus and Hilbert-style presentations of this logic and prove a strong normalisation result.

LOMAPS-CAMCL-6 Alan Mycroft and Kirsten Lackner Solberg: *Uniform PERs and Comportment Analysis* Hunt showed that the notion of PER-based strictness properties subsumed the incomparable notions of ideal- and projection-based properties on non-lifted value spaces. We extend Hunt's idea so it can encompass other *comportment* properties (such as totality) by separating the information and fixpoint orderings. We then define a class of *uniform* PERs (in the sense that they treat non-bottom ground elements, e.g. integers, identically) and then show how these can be hereditarily defined on function spaces. These (or subsets thereof) can be taken as a space of abstract values for abstract interpretation in the style of Nielson.

LOMAPS-CAMCL-7 Alan Mycroft (ed.): *Second International Static Analysis Symposium (Glasgow, 1995)* Static Analysis is increasingly recognised as a fundamental tool for high performance implementations and verification systems of high-level programming languages. The last two decades have witnessed substantial developments in this area, ranging from theoretical frameworks to the design and implementation of analysers and their application in optimising compilers.

This volume contains the proceedings of the Second International Static Analysis Symposium (SAS'95), held in Glasgow (UK) 25–27 September 1995. The continuing aim of SAS'95 is to promote contacts and information exchange among scientists who share common interests in static analysis for different programming paradigms. Researchers from the fields of concurrent, constraint, functional, imperative, logic and object-oriented programming constitute the audience of SAS.

The proceedings contains 3 invited papers, 22 refereed full papers and 2 system descriptions.

LOMAPS-CAMCL-8 Alan Mycroft, Pierpaolo Degano and Corrado Priami: *Complexity as a basis for comparing semantic models of concurrency* We investigate some aspects of interexpressiveness of languages and their (denotational) semantic models by viewing semantic functions from a complexity-theory viewpoint. We classify semantic functions as *polynomial* or *finite* if a language term of size n produces a meta-language object respectively polynomially bounded in n or finite. Languages involving concurrency manifest most interest which we associate to the fact that their semantic models in general lack λ-style abstraction. The paper provides a quantifiable reason why labelled event structures form a more reasonable model for the choice and concurrency operators of CCS than do synchronisation trees. Similarly we show the representation of conflict by places within (at least occurrence forms of) Petri-nets is exponentially larger than the relational representation within corresponding event structures. An application is a criterion for selection of semantic models for real-world algorithmic purposes; for example, 'model checking' algorithms which use an exponentially larger semantic representation of programs are unlikely to be efficient.

LOMAPS-CAMCL-9 Patrick Cousot, Radhia Cousot and Alan Mycroft: Report on Dagstuhl-seminar on 'Abstract Interpretation' (eds.) Abstract interpretation is a mathematical framework for specifying program analyzers which conservatively approximate program behaviours by abstraction of the language semantics. As can be seen from the abstracts of the talks, the seminar succeeded in bringing together researchers from different areas of application of abstract interpretation to focus discussion on various:

- program analysis methods;
- programming paradigms;
- program runtime properties;
- composable abstract domain designs;
- fixpoint computation methods leading to efficient implementations;
- designs of generic program analyzers and compilers using powerful analysis methods, four of them being demonstrated; and
- applications.

LOMAPS-CAMCL-10 N. Benton and P. Wadler. *Linear logic, monads and the lambda calculus* A category theorist models Girard's linear logic by comonads (with some additional structure), and Moggi's computational metalanguage by monads (with different additional structure). Both of these systems are are of interest, in part because of the existence of various translations of the simply-typed lambda calculus (or, equivalently, propositional intuitionistic logic) into each of them. Are the similarities between the two systems a mirage, or the reflection of a deep connection?

Recently, Benton described an adjoint model that subsumes both models of intuitionistic linear logic and a restricted class of models of the computational metalanguage. This shows that Girard's and Moggi's systems are indeed connected.

Here we strengthen that link by showing that the various translations of the lambda calculus proposed by Girard and Moggi are intimately related. In particular, the well-known 'Girard translation' corresponds to the trivial embedding of the lambda calculus into Moggi's metalanguage; an alternative translation due to Girard corresponds to Moggi's call-by-value translation; and Moggi's call-by-name translation corresponds to a new variant of the standard Girard translation. We also consider extending these results to include recursion.

Further, there are translations between the lambda calculi corresponding to these models. Corresponding to the adjoint model is a mixed linear and non-linear lambda calculus devised by Benton; corresponding to Moggi's model is his computational metalanguage; corresponding to Girard's work are a number of linear lambda calculi. The monadic and linear calculi map into the adjoint calculus, and this map preserves meaning.

LOMAPS-CAMCL-11 A.J. Kennedy *Programming Languages and Dimensions* Scientists and engineers must ensure that the equations and formulae which they use are dimensionally consistent, but existing programming languages treat all numeric values as dimensionless. This thesis investigates the extension of programming languages to support the notion of physical dimension.

A type system is presented similar to that of the programming language ML but extended with polymorphic dimension types. An algorithm which infers most general dimension types automatically is then described and proved correct.

The semantics of the language is given by a translation into an explicitly-typed language in which dimensions are passed as arguments to functions. The operational semantics of this language is specified in the usual way by an evaluation relation defined by a set of rules. This is used to show that if a program is well-typed then no dimension errors can occur during its evaluation.

More abstract properties of the language are investigated using a denotational semantics: these include a notion of invariance under changes in the units of measure used, analogous to parametricity in the polymorphic lambda calculus. Finally the dissertation is summarised and many possible directions for future research in dimension types and related type systems are described.

LOMAPS-CAMCL-12 P.N.Benton *A Unified Approach to Strictness Analysis and Optimising Transformations* Although strictness analysis has been studied extensively for more than a decade, there has been comparatively little serious work on just how the information gained by strictness analysis can be used as the basis for optimising transformations and how these transformations can be proved correct.

Here we propose a variant of Moggi's computational lambda calculus as am intermediate language in which to express strictness-based transformations. We present an optimising translation of a PCF-like language into this intermediate language which combines a simple higher-order strictness analysis with its associated optimisations into a single provably correct system. The system can express not only the transformation of call-by-name into call-by-value, but also the complementary optimisations which are based on knowing that certain expressions will already have been evaluated to weak head normal form.