Peter O'Hearn, John Power, Robert Tennent and Makoto Takeyama
Syntactic control of interference revisited
`http://www.elsevier.nl/locate/entcs/volume1/tennent`

In "Syntactic Control of Interference" (POPL, 1978), J. C. Reynolds proposes three design principles intended to constrain the scope of imperative state effects in Algol-like languages. The resulting linguistic framework seems to be a very satisfactory way of combining functional and imperative concepts, having the desirable attributes of *both* purely functional languages (such as PCF) *and* simple imperative languages (such as the language of **while** programs).

However, Reynolds points out that an "obvious" syntax for interference control has the unfortunate property that β-reductions do not always preserve typings. Reynolds has subsequently presented a solution to this problem (ICALP, 1989), but it is fairly complicated and requires intersection types in the type system. Here, we present a much simpler solution which does not require intersection types.

We first describe a new type system inspired in part by linear logic and verify that reductions preserve typings. We then define a class of "bireflective" models, which are shown to provide a sound interpretation of the type system; a companion paper, "Bireflectivity," in this volume provides a categorical analysis of these models. Finally, we describe a concrete model for an illustrative programming language based on the new type system; this improves on earlier such efforts in that states are not assumed to be structured using locations.

Peter O'Hearn and Uday Reddy
Objects, interference and the Yoneda embedding
`http://www.elsevier.nl/locate/entcs/volume1/ohearn`

We present a new semantics for Algol-like languages that combines methods from two prior lines of development:

- the object-based approach of Reddy (1994) and (1995), where the meaning of an imperative program is described in terms of sequences of observable actions, and
- the functor-category approach initiated by Reynolds (1981), where the varying nature of the run-time stack is explained using functors from a category of store shapes to a category of cpos.

The semantics gives an account of both the phenomena of local state and irreversibility of state change. As an indication of the accuracy obtained, we present a full abstraction result for closed terms of second-order type in a language containing active expressions, i.e. value-returning commands.

Frank Pfenning and Hao-Chi Wong
On a modal lambda calculus for S4
`http://www.elsevier.nl/locate/entcs/volume1/pfenning`

We present $\lambda^{\rightarrow\Box}$, a concise formulation of a proof term calculus for the intuitionistic modal logic S4 that is well-suited for practical applications. We show that, with respect to provability, it is equivalent to other formulations in the literature, sketch a simple type checking algorithm, and prove subject reduction and the existence of canonical forms for well-typed terms. Applications include a new formulation of natural deduction for intuitionistic linear logic, modal logical frameworks, and a logical analysis of staged computation and binding-time analysis for functional languages.

Michel Schellekens
The Smyth completion: a common foundation for denotational semantics and complexity analysis
`http://www.elsevier.nl/locate/entcs/volume1/schellekens`

The Smyth completion introduced by M.B. Smyth provides a topological foundation for Denotational Semantics. We show that this theory simultaneously provides a topological foundation for the complexity analysis of programs via the new theory of "complexity (distance) spaces". The complexity spaces are shown