Journal of Automata, Languages and Combinatorics **17** (2012) 2–4, 185–204

COMPUTING WITH CAPSULES

Jean-Baptiste Jeannin
Department of Computer Science, Cornell University
Ithaca, New York 14853-7501, USA
e-mail: jeannin@cs.cornell.edu

and

Dexter Kozen
Department of Computer Science, Cornell University
Ithaca, New York 14853-7501, USA
e-mail: kozen@cs.cornell.edu

ABSTRACT

Capsules provide an algebraic representation of the state of a computation in higher-order functional and imperative languages. A capsule is essentially a finite coalgebraic representation of a regular closed λ-coterm. One can give an operational semantics based on capsules for a higher-order programming language with functional and imperative features, including mutable bindings. Static (lexical) scoping is captured purely algebraically without stacks, heaps, or closures. All operations of interest are typable with simple types, yet the language is Turing complete. Recursive functions are represented directly as capsules without the need for fixpoint combinators.

Keywords: capsules, semantics, functional programming, imperative programming

1. Introduction

Capsules provide an algebraic representation of the state of a computation in higher-order functional and imperative programming languages. They conservatively extend the classical λ-calculus with mutable variables and assignment, enabling the construction of certain regular coterms (infinite terms) representing recursive functions without the need for fixpoint combinators. They have a well-defined statically-scoped evaluation semantics, are typable with simple types, and are Turing complete.

Representations of state have been studied in the past by many authors. Approaches include syntactic theories of control and state [11, 12], the semantics of local storage [14], functional languages with effects [22, 23, 24], monads [28], closure structures [3, 4, 5] and denotational semantics [27, 35, 36]. Capsules provide a purely algebraic alternative in that no combinatorial structures are needed. Perhaps the most important aspect of capsules is that static scoping and local variables are captured without the need for closures. Cumbersome combinatorial machinery such as