

Abstract

We call language L_1 *intensionally more expressive* than L_2 if there are functions which can be computed faster in L_1 than in L_2 . We study the intensional expressiveness of several languages: the Berry-Curien programming language of sequential algorithms, CDS0, a deterministic parallel extension to it, named CDSP, and various parallel extensions to the functional programming language PCF. The paper consists of two parts.

In the first part, we show that CDS0 can compute the minimum of two numbers n and p in unary representation in time $O(\min(n, p))$. However, it cannot compute a “natural” version of this function. CDSP allows us to compute this function, as well as functions like parallel-or. This work can be seen as an extension of the work of Colson [7, 8] with primitive recursive algorithms to the setting of sequential algorithms.

In the second part, we show that deterministic parallelism adds intensional expressiveness, settling a “folk” conjecture from the literature in the negative. We show that CDSP is more expressive intensionally than CDS0. We also study three parallel extensions to PCF: parallel-or (por) and parallel conditionals on booleans (pif_o) and integers (pif_i). The situation is more complicated there: pif_i is more expressive than both pif_o and por . However, pif_i still is not as expressive as the deterministic query construct of CDSP. Thus, we identify a hierarchy of intensional expressiveness for deterministic parallelism.

*This research was sponsored by the Office of Naval Research under Grant No. N00014-93-1-0750.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

To appear in the Proceedings of the 22nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, January 1995