

Towards a Parallel Search for Solutions of Non-deterministic Computations

Fabian Reck and Sebastian Fischer

Department of Computer Science
Christian-Albrechts-University of Kiel
D-24118 Kiel
{fre,sebf}@informatik.uni-kiel.de

Non-determinism is one of the distinctive built-in features of logic and functional logic programming languages such as Prolog and Curry. In other programming languages non-determinism can be modeled using appropriate abstractions. In Haskell this is usually done by means of non-determinism monads. Since the different results of a non-deterministic computation do not depend on each other it should be possible to compute them in parallel. In this paper we explore different possibilities to execute non-deterministic Haskell programs in parallel. Together with our latest attempts to compile Curry programs to monadic Haskell we already achieve preliminary but encouraging results.

We present three different implementations of a parallel execution of non-deterministic monadic Haskell programs:

1. dividing the computation into a fixed number of sub-computations and assign each of them to a different thread,
2. using a Bag-of-Tasks approach to distribute small tasks to a fixed number of threads, and
3. using the `par`-combinator that is implemented in the GHC to give *hints* to the Haskell runtime system about which parts of the computation can be done in parallel.

A detailed description of these approaches together with a discussion of their properties and selected benchmarks is published in the proceedings of the 39th annual conference of German Gesellschaft für Informatik [1].

References

1. Reck, F., Fischer, S.: Towards a Parallel Search for Solutions of Non-deterministic Computations. In: INFORMATIK 2009, Im Focus das Leben, 39. Jahrestagung der Gesellschaft für Informatik. LNI, GI (2009)