



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

Dept. of Doctoral Studies
Faculty of Mathematics and Physics
Charles University
Ke Karlovu 3
121 16 Praha 2

Prague, April 6, 2018

Doctoral Thesis – Advisor’s reference

Michal Brabec: Procedural code integration in streaming environments

The main topic of the thesis is the design of a novel intermediate representation for parallel programs, intended to be compatible with run-time environments based on stream-like processing, also called data-flow graphs. Together with the design of this new representation called Hybrid Flow Graph (HFG), a corresponding compiler architecture is presented which converts C# programs (from their CIL representation) to HFG, performs several optimizing transformations on the HFG, and generates runnable code in two flavors, targeted at two parallel run-time platforms. The main theoretical contribution of this thesis is the introduction and formal definition of the Hybrid Flow Graph, a non-traditional form of intermediate code capable of mixing control-flow and data-flow structures in one layer. The operational semantics of the HFG is formally defined using graph-rewriting rules; similar graph-rewriting techniques are used in the formal description of several transformations required in the compilation chain.

The thesis deals with the translation of (a subset of) the C# language to HFG as an example of a procedural input language. Several steps in this translation are equivalent to well-known phases of traditional compilers – the difference is in the use of non-procedural intermediate representation. In this sense, the thesis is both a theoretical and an experimental evidence that this approach is viable.

The experiments described in the thesis show that the performance degradation caused by the necessary adherence to the protocols of the particular run-time system is relatively low. In many cases, the degradation is negligible compared to the performance gain obtained by the parallelism available in the run-time system. Thus, the demonstrated compiler/run-time chain forms an usable system for automatic parallelization of (simple) non-parallel C# programs.

As a non-trivial use-case, the thesis describes the application of the system on the Wagner–Fischer algorithm for edit distance implemented in non-parallel C# and compares the results with a parallel C++ implementation. Although there is still some performance gap, the C# version compiled via HFG produces a usable implementation at significantly lower development cost than the C++ version.

The thesis contains both a non-trivial theory, especially the definition of the HFG, and a large implementation part accompanied by an extensive set of regression tests as well as performance experiments. For practical usability, there is still a lot of work to be done, in particular the proposed mixing of HFG from different sources was not yet demonstrated and a number of optimization

steps is also missing. However, the remaining work in the project could probably constitute another thesis (such a thesis was already underway but the author abandoned his PhD. study).

The most valuable parts of the thesis are probably the elegant use of graph rewriting in the description of both semantics and transformation of the HFG. Furthermore, the thesis produced a significant, well organized, documented and tested code base which can be developed further.

The theoretical foundations as well as part of the experiments presented in the thesis have been published in an impacted journal and several proceedings of international peer-reviewed conferences. The author has also shown that he is a skilled scientific presenter (as demonstrated by a best-paper award).

I think that the thesis as well as the publications show that Michal Brabec is capable of performing high-quality and relevant research. Therefore, I recommend the thesis for defense as well as awarding the PhD degree to the author.

David Bednárek
advisor