Implementing Randomised Algorithms in Constraint Logic Programming

N. Angelopoulos, A. Di Pierro and H. Wiklicky
Department of Computer Science
City University, London EC1V OHB
{nicos,adp,herbert}@cs.city.ac.uk

Synopsis

Algorithms where the flow of information is determined by an element of random choice (a "coin flipping" device), aka randomised algorithms, have known in the last decade a tremendous growth of interest, especially in the field of complexity theory.

Up to now such algorithms have been implemented mostly by means of probabilistic programs written in imperative languages. These programs have been analysed largely only by ad-hoc methods. However, real problems, and especially problems related to probabilistic concepts, are usually quite complicated and intuition is very likely to fail. Therefore, a formal deductive system would be much more handy.

In this paper, we propose a declarative-based implementation of randomised algorithms, which exploits the Concurrent Constraint Programming (CCP) paradigm for the formalisation of such algorithms. The main advantage in doing so is that we can rely on a sound and mathematically rigorous semantical framework, instead of on intuition, for tasks related to reasoning about probabilistic programs (analysis, synthesis, transformation and verification).

The foundations of such a semantical framework have been devised in previous papers [3, 2], where we introduce a probabilistic language based on CCP, namely Probabilistic CCP (PCCP), and provide it with an operational and a denotational semantics. In PCCP randomness is expressed by means of a construct for probabilistic choice, which replaces (or refines) nondeterminism, and allows the program to make stochastic moves during its execution.

In this work we concentrate on the implementation aspects of PCCP by designing a meta-interpreter for it. PCCP also includes synchronisation and concurrency aspects. However, for the time being, we put aside these aspects, by treating the parallel construct as a purely sequential one. As a consequence, the guarded choice, yet present in our language as a *global* choice (guards are not necessarily to be *true*), does not allow any synchronisation. From this point of view we effectively use here only the (probabilistic) Constraint Logic Programming (CLP) fragment of PCCP.

For the constraint solving, the meta-interpreter exploits existing constraint handling facilities (more concretely the SICStus 3.#6 system). This is possible because the design of PCCP does not require any additional structure on the underlying constraint system (e.g. fuzzy or belief systems). The current prototype consists basically of two phases: first PCCP agents and declarations are handled by a preprocessor that transforms them into an internal representation; this is then "executed" in a second phase. For more details we refer to [1]. In that paper, the capabilities of our system in implementing randomised algorithms are also demonstrated, by presenting an example of a generic randomised algorithm commonly used for combinatorial optimisation, namely Simulated Annealing, along with its application for solving specific NP-complete problems, like the travelling salesman problem.

These results show that the PCCP language is not only of theoretical interest, but can also be used for "real" applications. We do not claim that the examples we present are efficient implementations from the complexity-theoretic point of view – this is not our objective. However, our system offers all the advantages stemming from declarativity, like readability, flexibility, clarity of semantics, etc.

References

[1] N. Angelopoulos, A. Di Pierro, and H. Wiklicky. Randomised Algorithms and CLP. Technical Report 1998/02, Department of Computer Science, City University, 1998.

[2] A. Di Pierro and H. Wiklicky. A Banach Space Based Semantics for Probabilistic Concurrent Constraint Programming. In Xuemin Lin, editor, *Proc. 4th Australasian Theory Symposium, CATS'98*, volume 20 – 3 of *Australian Computer Science Communications*, pages 245–259, Singapore, 1998. Springer-Verlag.

[3] A. Di Pierro and H. Wiklicky. An Operational Semantics for Probabilistic Concurrent Constraint Programming. In Y. Choo P. Iyer and D. Schmidt, editors, *Proc. ICCL'98 – International Conference on Computer Languages*, IEEE Computer Society and ACM SIGPLAN, Chicago, 1998. IEEE Computer Society Press.