

**Foster, Ian; Taylor, Stephen**

**Flat Parlog: A basis for comparison.** (English) Zbl 0639.68014

*Int. J. Parallel Program.* 16, No. 2, 87-125 (1987).

Three similar parallel logic programming languages have been proposed; Parlog, Flat Concurrent Prolog, and Guarded Horn Clauses. Quantitative comparison of the languages has not previously been possible since they employ different execution models and implementation techniques. In order to uncover the effects of semantic differences on efficiency, a common basis is required for experimentation. This paper presents a subset of the language Parlog called Flat Parlog which provides a basis for quantitative comparison. The language combines the directional semantics of Parlog with the simple execution model of Flat Concurrent Prolog.

A performance comparison between Flat Parlog and Flat Concurrent Prolog based on new implementations of both languages is presented. These new implementations are identical except for optimizations that are possible by virtue of semantic differences. Benchmark results indicate that Flat Parlog is more efficient; experiments have been able to quantify and explain this performance differential. A detailed description of the abstract machine for Flat Parlog is presented to illustrate the simplicity of the language.

**MSC:**

[68N01](#) General topics in the theory of software

[68N25](#) Theory of operating systems

[68T15](#) Theorem proving (deduction, resolution, etc.) (MSC2010)

Cited in 1 Document

**Keywords:**

[performance evaluation](#); [parallel logic programming](#); [Concurrent Prolog](#); [Flat Parlog](#)

**Software:**

[GHC](#) ; [PARLOG](#)

**Full Text:** [DOI](#)

**References:**

- [1] C. Mierowsky, S. Taylor, E. Shapiro, J. Levy, and M. Safra, The design and implementation of Flat Concurrent Prolog, Technical Report CS85-09, Weizmann Institute, Rehovot (1985).
- [2] E. Shapiro, Concurrent Prolog: A Progress Report, IEEE Computer (August 1986).
- [3] K. L. Clark and S. Gregory, PARLOG: parallel programming in logic, in *ACM Trans. on Programming Languages and Systems*, 8(1):1-49. · [Zbl 0592.68016](#)
- [4] K. Ueda, Guarded Horn Clauses, EngD thesis, University of Tokyo (1986). · [Zbl 0771.68037](#)
- [5] A. Takeuchi and K. Furukawa, Parallel logic programming languages, in *Proc. of the 3rd Intl Logic Programming Conf. (London, July)*, (ed.), E. Shapiro, New York, Springer-Verlag, pp. 242-254.
- [6] S. Taylor, An Abstract Machine for Implementing FCP on Parallel Architectures, Weizmann Institute of Science (February 1986).
- [7] S. Gregory, Parallel Logic Programming in PARLOG, Reading, Mass.: Addison-Wesley (1987). · [Zbl 0592.68016](#)
- [8] S. Taylor and E. Shapiro, Compiling Guarded Logic Programs into Decision Graphs, Weizmann Institute of Science (February 1986).
- [9] S. Safra, Partial Evaluation of Concurrent Prolog and its Implications, M.Sc. Thesis, CS86-24, Weizmann Institute of Science (July 1986).
- [10] A. Houry and E. Shapiro, A sequential abstract machine for Flat Concurrent Prolog, Weizmann Institute Technical Report CS86-19, Rehovot (1986).
- [11] S. Gregory, I. T. Foster, A. D. Burt, and G. A. Ringwood, An abstract machine for the implementation of PARLOG on uniprocessors. Submitted for publication. · [Zbl 0664.68029](#)
- [12] K. L. Clark and S. A. Tarnlund, A first order theory of data and programs, in *Information Processing 77; Proc. of the IFIP Congress 77* (ed.), B. Gilchrist, Amsterdam: Elsevier/Nother Holland, pp. 939-944.
- [13] S. Taylor, S. Safra, and E. Shapiro, Parallel execution of Flat Concurrent Prolog, in *Intl. Journal of Parallel Programming*, 15(3):245-275. · [Zbl 0614.68007](#)

- [14] D. H. D. Warren, An abstract Prolog instruction set. Technical note 309, SRI International, Menlo Park, California.
- [15] M. Codish and E. Shapiro, Compiling Or-parallelism to And-parallelism, inProc. of the 3rd Intl. Logic Programming Conf. (London, July), (ed.), E. Shapiro, New York, Springer-Verlag, pp. 283-297. · [Zbl 0595.68026](#)
- [16] I. T. Foster, Logic operating systems: design issues, inProc. 4th Intl. Conf. on Logic Programming, Melbourne (May 1987).

This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.