

Performance Verification Using Partial Evaluation and Interval Analysis

Jeffrey Walrath, Ranga Vemuri, and William Bradley

University of Cincinnati, ECECS Department, Cincinnati, Ohio 45221-0030

Email: ranga.vemuri@uc.edu

A performance model for a typical design represented in a high-level description language can be generated by augmenting the design components with *attributes* and *evaluation rules* [1]. An attribute represents some performance aspect of a design that can be either assigned a base initial value or calculated using an evaluation rule. Heat dissipation, dynamic power consumption, and maximum throughput rate are just a few examples of various performance aspects that can be represented with attributes.

Evaluation rules contained in the performance model can be classified as either *equational* or *procedural*. An equational performance model is a model containing only evaluation rules that are composed of mathematical operations such as addition, subtraction, and so forth. Likewise, a procedural performance model may contain equational rules, but it also has rules composed of complex programming constructs such as an assignment statement, if-then-else, case, and while control constructs and procedure calls.

Our method for performance verification involves placing relational constraints on attributes in the performance model and determining whether all constraints can be satisfied simultaneously [2]. Interval mathematics provide a convenient technique to represent relational constraints as intervals. Each attribute has an initial interval from negative infinity to positive infinity. Further constraints are specified by the user, the interval analysis technique is applied, and a verification result is produced. This result is in the form of a statement that the constraints can be met (“yes”), or they cannot be met (“no”).

During interval analysis, forward and backward evaluations of the evaluation rules continue with squeezing of all intervals until either no interval changes value or an interval becomes empty. In order to use interval mathematics for verification of performance models, it is required that every evaluation rule in the model be defined using invertible operations. Programming constructs such as if-then-else, case, and while control statements as well as assignments and procedure calls are not invertible. Thus,

our verification approach is suitable only for performance models that are equational.

With partial evaluation techniques [3], it is possible to reduce a procedural performance model to an equational model. By specifying some, but not all, of the input information, the procedural constructs contained within the model can be symbolically evaluated. If a sufficient number of inputs are specified, a residual model can be rendered which is entirely equational. Thus after partial evaluation, the residual model can be subjected to verification using interval analysis.

Further investigation is planned for more closely integrating both partial evaluation and interval propagation along with partially evaluating and verifying models containing dynamic attributes. A partial evaluator and interval-analysis based performance verification tool are implemented for the high-level description language PDL. PDL is a modeling environment which produces performance models suitable for input to both the partial evaluator and verification tool. The complete PDL System containing the evaluator and verifier can be obtained from the PDL homepage. In addition, a technical report detailing all the information presented here can be found at <http://www.ececs.uc.edu/~ddee1/pdl.html>.

References

- [1] Ranga Vemuri, Ram Mandayam, Vijay Meduri. “Performance Modeling Using PDL”. *IEEE Computer*, pages 44–53, April 1996.
- [2] William Bradley and Ranga Vemuri. “Performance Verification Using PDL and Constraint Satisfaction”. In *Proceedings of ASP-DAC’95, CHDL’95, VLSI’95*, pages 531–538. CHDL95, 1995.
- [3] Carsten K. Gomard Neil D. Jones and Peter Sestoft. *Partial Evaluation and Automatic Program Generation*. Prentice Hall, Englewood Cliffs, N.J., 1993.