IMPACT: Innovative Models for Prolog with Advanced Control and Tabling

Ricardo Rocha[1], Ricardo Lopes[1], Fernando Silva[1], and Vítor Santos Costa[2]*

[1] DCC-FC & LIACC
University of Porto, Portugal
{ricroc,rslopes,fds}@ncc.up.pt
[2] COPPE Systems & LIACC
Federal University of Rio de Janeiro, Brazil
vitor@cos.ufrj.br

Extended Abstract

During the past years, our research group has been working in the design and implementation of Logic Programming Systems. In previous work, we have produced systems to support sequential, parallel and distributed execution of Prolog; to support novel techniques and models, such as tabling, through the YapTab system [1], or the Extended Andorra Model (EAM), through the BEAM [2]; and to support the combination of the above, such as parallel tabling [3]. With the IMPACT project we want to combine the power of tabling with that of EAM in order to produce an execution model with advanced control strategies that guarantees termination, avoids looping, reduces the search space, and is less sensitive to goal ordering. Ultimately, we believe such a system will allow novel logic programming applications.

To the best of our knowledge, YapTab is the only proposal that compares favorably with current versions of XSB [4], the most successful and well-known tabling Prolog system. YapTab has been designed taking into account the major purpose of further integration with parallelism to achieve the first available parallel tabling computational system, the OPTYap [3].

The Andorra based execution models were designed to explore more advanced control strategies aiming at reducing the search space and maximizing available parallelism. To study whether the EAM with implicit control was practical, we have developed the BEAM, a novel system that refines Warren's original design. Performance results showed that our model is less sensitive to goal ordering and quite effective at reducing the search space.

We have noticed that both EAM and tabling have common characteristics that seem to fit naturally. First, tabling contributes to one of the main premises of the EAM, reducing search space through the reuse of goals. Second, both paradigms are less sensible to goal ordering. Third, tabling avoids looping thus

* This work has been partially supported by APRIL (POSI/SRI/40749/2001), Myddas (POSC/EIA/59154/2004) and by funds granted to LIACC through the Programa de Financiamento Plurianual, Fundação para a Ciência e Tecnologia and Programa POSC.

guaranteeing termination for programs with recursive computations. Is thus our belief that BEAM with tabling opens up a design space for novel mechanisms to improve both control and performance of existing applications and to achieve a more declarative formulation of search and database applications.

To address the integration between tabling and the EAM we have identified the following tasks for the IMPACT project. **(1)** Build an external module in C implementing tabling primitives that provide direct control over the search strategy. This module will allow tabling to be easily incorporated into other Prolog systems. **(2)** Use the external module with BEAM to study how tabling interacts with the EAM. This involves running and validating the system with a set of benchmark applications. **(3)** Design a new model to extend the BEAM system to support tabling running within the EAM environment. There are several major problems that need to be tackled. **(i)** Integrate the basic operations of tabling evaluation, such as checking for variants, checking for new answers, and performing completion, with the BEAM execution rules. This includes studying and defining how to start the execution of tabled subgoals, when new answers should be allowed to be exported, and how completion should be done. Note that whereas these operations are already complex for Prolog, they are even more intricate within the EAM. **(ii)** Study the implications of tabling evaluation on the BEAM rewrite rules. Some of the BEAM rewrite rules will need to be modified to support tabling. We will address the implications on the reduction, splitting, eager-splitting, promotion, propagation, and-compression, deterministic-reduce-and-promote, success propagation, and failure propagation rules. **(iii)** Study how pruning should be implemented on BEAM with tabling. BEAM allows both implicit and explicit pruning, but aggressive pruning may interfere with tabling and result in incomplete tables. **(4)** Implement the combined system. The proposed work will be implemented on top of the existing systems and will profit from the expertise of our research group in the area.

In this project we focus on combining tabling with the EAM. Our ultimate goal is to develop a system that supports simultaneously and efficiently parallelism, tabling, and the EAM model.

References

1. Rocha, R., Silva, F., Santos Costa, V.: YapTab: A Tabling Engine Designed to Support Parallelism. In: Conference on Tabulation in Parsing and Deduction. (2000) 77–87
2. R. Lopes, V.S., Silva, F.: A Novel Implementation of the Extended Andorra Model. In: International Symposium on Pratical Aspects of Declarative Languages. Number 1990 in LNCS, Springer-Verlag (2001) 199–213
3. Rocha, R., Silva, F., Santos Costa, V.: On applying or-parallelism and tabling to logic programs. Journal of Theory and Practice of Logic Programming **5** (2005) 161–205
4. Sagonas, K., Swift, T., Warren, D.S.: XSB as an Efficient Deductive Database Engine. In: ACM SIGMOD International Conference on the Management of Data, ACM Press (1994) 442–453