Modularity and Temporal Reasoning: a Logic Programming Approach

Vitor Nogueira and Salvador Abreu
Universidade de Évora and CENTRIA
Portugal
{vbn,spa}@di.uevora.pt

Abstract

Albeit temporal reasoning and modularity are very prolific fields of research in Logic Programming (LP), we find few examples of their integration. In this paper we propose the addition of temporal annotations to a modular extension of LP. Moreover, we also provide a sketch for a compiler, allowing this way for the development of applications based on such language.

1 Introduction

Over the past decades the volume of temporal data has grown enormously, making modularity a requisite for any language suitable for developing applications that deal with such information. We propose to integrate temporal reasoning with a modular logic language such that the relation between the *time of the module* and the *time of environment* will determine whether the module is eligible or not to include for solving a given goal.

2 Modular Language

The modular language is Contextual Logic Programming – CxLP, a simple and powerful extension of logic programming with mechanisms for modularity and object orientation which is partly described in [1]. CXLP has properties that makes it very suitable for integrating with temporal reasoning, for instance its quite straightforward to add the notion of *time of the context*.

3 Temporal Language

The temporal paradigm chosen was Temporal Annotated Constraint Logic Programming (TACLP) [2] since this a logical language that supports qualitative and quantitative (metric) temporal reasoning involving both time points and time periods (time intervals) and their duration. Moreover, it allows one to represent definite, indefinite and periodical temporal information.

4 Integrating Modularity and Temporal Reasoning

In CxLP with overriding semantics, to solve a goal `G` in a context `C`, a search is performed until the topmost unit of `C` that contains clauses for the predicate of `G` is found. This is the basic mechanism of CxLP (called *context search*) and it is this process that we are going to modify for temporal reasoning. In order to accomplish this, we add temporal annotations to contexts and to units and it will be the relation between those two annotations that will help to decide if a given unit is eligible to match a goal during a context search.

A compiler for the proposed language can be obtained by combining a program transformation with the compiler for TACLP, producing a CxLP with finite domain constraints program as output. Finally, since GNU Prolog/CX besides the CxLP primitives also has a constraint solver for FD, the implementation of this language is direct on such system.

We already applied our prototype implementation, an early version of which is described in [3], to the legal reasoning problem domain and are currently extending to other areas such as medicine and natural language processing. Finally, it is our goal to build on our previous work and experiment with this framework as the basis for constructing and maintaining temporal information systems.

References

[1] S. Abreu and D. Diaz. Objective: In minimum context. In C. Palamidessi, editor, *ICLP*, volume 2916 of *Lecture Notes in Computer Science*, pages 128–147. Springer, 2003.

[2] T. W. Frühwirth. Temporal annotated constraint logic programming. *J. Symb. Comput.*, 22(5/6):555–583, 1996.

[3] V. Nogueira and S. Abreu. Temporal contextual logic programming. In F. J. L. Fraguas, editor, *Proceedings of WFLP'06*, Madrid, Spain, November 2006. ENTCS.