Temporal Concurrent Constraint Programming

Frank D. Valencia

BRICS⋆, University of Aarhus, Denmark
fvalenci@brics.dk

The temporal ccp model *tcc* [3] is aimed at specifying timed systems. Time is conceptually divided into discrete intervals. In a particular time interval, a ccp process receives a stimulus (i.e. a constraint) from the environment, it executes with this stimulus as the initial store, and when it reaches its resting point, it responds to the environment with the resulting store. Also the resting point determines a residual process, which is then executed in the next time interval. This temporal ccp model is inherently deterministic and synchronous.

The *ntcc calculus* [2] is a nondeterministic version of tcc which also allows asynchronous behavior. The motivation for this extension was partly the desire to be able to specify natural temporal behaviors like "the system must output c *within* the next t time intervals", which is not possible in tcc. Also, the extension is argued to be consistent with the declarative flavor of ccp, i.e. to free the programmer from over-specifying a deterministic solution, when a non-deterministic simple solution is more appropriate (following the arguments behind Dijkstra's language of guarded commands). Furthermore, it is argued that a very important benefit of allowing the specification of non-deterministic and asynchronous behavior arises when modeling the interaction among several components running in parallel. These systems often need non-determinism to be modeled faithfully.

In [2] a relative complete proof system for linear-time properties of ntcc processes is studied. In [1] various notions of behavior for the ntcc calculus are introduced: the input-output and the language equivalence and their congruences, all motivated operationally and/or logically. The notions are related, and proved to be decidable for a substantial fragment of the calculus. The expressive power of ntcc has been illustrated by modeling bounded response and invariance specifications, constructs such as cells, bounded broadcasting, some applications involving the programming of RCX$^{\text{TM}}$ controllers [2] and a version of a Predator/Prey (Pursuit) game [1].

References

1. M. Nielsen and F. Valencia. Temporal concurrent constraint programming: Applications and behavior. Technical report, BRICS, August 2001.
2. C. Palamidessi and F. Valencia. A temporal concurrent constraint programming calculus. In *Proc. of the Seventh International Conference on Principles and Practice of Constraint Programming*, 26 November 2001.
3. V. Saraswat, R. Jagadeesan, and V. Gupta. Foundations of timed concurrent constraint programming. In *Proc. of the Ninth Annual IEEE Symposium on Logic in Computer Science*, pages 71–80, 4–7 July 1994.

⋆ Basic Research in Computer Science, Centre of the Danish National Research Foundation.

T. Walsh (Ed.): CP 2001, LNCS 2239, p. 786, 2001.
© Springer-Verlag Berlin Heidelberg 2001