Project 22703: SYRF

Deliverable 5.4.2

Verification of mixed A/S systems

1 Approach

In last year deliverable we motivate the study of mixed Asynchronous/Synchronous systems by our belief that a large number of real-life reactive systems actually consisted of local *synchronous islands*, where synchronous parallelism is used as a *programming* feature to build complex end-point controlers from basic components (possibly by synthesis), in the middle of a larger *asynchronous "ocean" network*, itself provided (as opposed to "programmed"), and which should simply be modeled with asynchronous specification primitives if verification and analysis are to be conducted on the whole global system.

We were concerned in this deliverable with efficient verification of such systems, and more specifically with the efficient dual partitioning of the transition relation with respect to its application in the computation of the (implicit) reachable state space using BDD techniques. This partitioning works along individual control registers for local synchronous parts, and according to individual events at the global asynchronous level.

2 Results achieved this year

The work this year was mostly spent in the programming of a prototype tool implementing the partitioning of transition for analysis of mixed A/S systems. This tool is "almost completed" but we figured that efforts had to be spent so that valued signals could be dealt with, if significant examples were to be tackled. So data representation inside model-checking (using discrete types) is the main unfinished part. This topics bears strong connections with deliverable 6.1.2, where translation to hardware description formalisms also requires dicrete type implementation of data.

Still, we refined our proposed algorithms and the detailed method was presented at the IWLS'98 workshop in Lake Tahoe this year [BdS98]. This paper is provided as an annex to the deliverable. As mentioned above it contains technical material on implementation of transition relation partititioned across two levels : an inner *synchronous* level where partitioning follows "classical" division according to boolean control registers; an outer level where partitioning is divided according to global actions (or *events*), more in the lines of process algebras. Of course here the behaviours of single components can be more complex than a simple synchronisation offer, which accounts for the main issue in combining local behaviours coorectly into global ones.

References

[BdS98] A. Bouali, R. de Simone, A Symbolic Representation of Asynchronous Networks of Synchronous Processes, *in: Proceedings of the Tenth International Workshop on Logic Synthesis IWLS'98*, Lake Tahoe, CA, USA, juin 1998.