Motivation

• The era of multicore architectures in computer hardware brings an unprecedented need for parallel programming. We are interested in multithreaded programs with shared data, where lock primitives are used to control access to the shared data.

• Mutual exclusion locks (mutexes) are used to protect shared data from inconsistent concurrent updates. Improper use of mutexes can lead to the “deadly embrace” problem and Circular-Mutex-Wait (CMW) deadlocks.

• In our on-going project, called Gadara, the objective is to control the execution of multithreaded programs in order to avoid deadlocks by using techniques from discrete control theory (see Fig. 1).

Main Contributions

• We propose a systematic program control architecture (see Fig. 1) based on Discrete Control Theory, which combines the strengths of offline analysis and control synthesis with online knowledge and control to dynamically avoid CMW deadlocks in concurrent programs.

• We define a special class of Petri nets, called Gadara nets, to formally model lock allocations and releases in multithreaded programs (see Fig. 2).

• We establish necessary and sufficient conditions for liveness and reversibility of Gadara nets, which connect these behavioral properties of the dynamic system to a certain structure in the net, called siphon.

• We develop an efficient control synthesis strategy for Gadara nets, based on structural analysis. The strategy is both correct and maximally permissive with respect to the goal of liveness enforcement for Gadara nets.

Acknowledgment

• The Gadara project is supported by the National Science Foundation and by HP Labs.