On verifying robustness of concurrent systems

Ahmed Bouajjani*[1]

[1]Université Paris Diderot, IRIF – - – France

Résumé

Concurrent systems are in general used by their clients under strong assumptions on their visible behaviors. This allows a modular design approach: at the level of the client, these assumptions allow to reason in an abstract way about the behaviors of the invoked systems. For instance, the users of a shared memory may assume that the implementation of the memory is sequentially consistent, which means that it behaves according to the standard interleaving model where write/read operations are considered to be atomic, and immediately visible to all parallel users.

In another context, the users of web services may consider that their requests are handled atomically in a serial way, etc. However, for performance reasons, the implementations of concurrent systems tend to parallelize operations, and to use optimizations (including reordering of actions) in order to increase the throughput of the system. This leads in general to relaxations in the semantics guaranteed by these implementations w.r.t. to strong consistency models. In this talk, we will address to issue of checking that a given program of the client is robust against this kind of relaxations, i.e., the observable behaviors of the client are the same under both the strong and relaxed consistency models. Robustness corresponds to a correctness criterion that ensures the preservation of all properties that can be proved assuming strong consistency. We show that this property can be checked efficiently in several cases by linear reductions to state reachability problems.

*Intervenant