The reachability analysis is useful in verifying the safety of real-time protocols and systems. Different with the FSM model we discussed in class, the reachability analysis for timed automaton (TA) is more difficult due to the infinite size. In this section, the author introduces the techniques used to perform reachability analysis in TA step by step.

A location $s$ in a timed automaton $A$ can correspond to more than one state in its corresponding transition system $S_A$ with different clock interpretations. To determine whether a location is reachable, it is sufficient to check if any of these states are reachable in the transition system.

Step 1. In order to truncate the transition system to a finite size, we create another transition system with time label hided, called time-abstract transition system $U_A$.

Step 2. Since this time-abstract transition system still has infinite number of states, we reduce $U_A$ to its stable quotients.

Step 3. Equivalence relation is defined between two states with the same location and region-equivalent clock interpretations. The quotient of a TA with respect to the region equivalence is called region automata.

The complexity of reachability is linear in the number of locations, exponential to the number of clocks and exponential in the encoding of the constant.

Step 4. The regions can be collapsed by considering convex unions of clock regions, namely the clock zones. The zone automaton is a transition system taking zones as states. We can use matrices to represent clock zones.

Step 5. Implementation. The solution for reachability of input locations is provided by searching the region automaton zone automaton. The actual search can be done by an on-the-fly enumerative engine or a BDD-based symbolic engine.

Some remarks are made at the end of this section. First, we can choose time domain to be the set of rational numbers instead of nonnegative real number. It would not affect the analysis result. However, if we choose the set of nonnegative integer, the system becomes discrete-time model and the result change quite a bit.

Doubts after 2nd level reading

With all these techniques for reachability analysis introduced, it is still not clear how each state is explored. Is there any graphical representation or solution to this problem?