Modeling and Analysis of Real-Time Systems with Mutex Components

APDCM'10

Guoqiang Li[1], Xiaojuan Cai[1],Shoji Yuen[2]

[1]BASICS, Shanghai Jiao Tong University

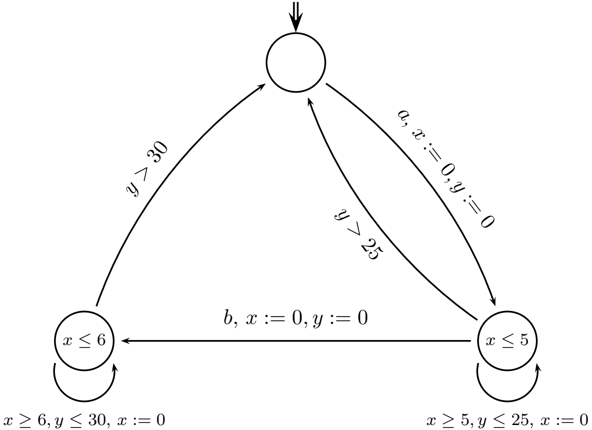
[2]Graduate School of Information Science, Nagoya University

19th, April. 2010

Backgrounds and Aims

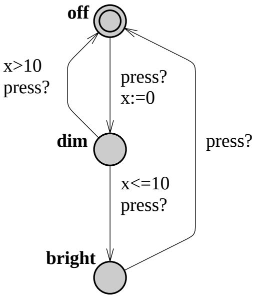
- Formal models for complex real-timed systems (e.g. timed automata).
- A real-time system consists of several functionally independent components that interact with each other, e.g. processors, controllers, various chips, etc.
  - Synchronization is modeled by parallel composition of timed automata [RTSS'95]
  - Mutex . . .
- In synthesis of a whole system, the "global" control of components is a key issue in design.
- Whether such a synthesis is decidable?

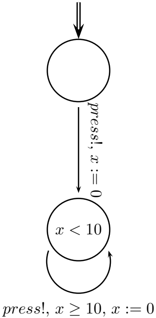
Timed Automata [Alur & Dill TCS 94]



Parallel Composition [Wang Yi et. al. RTSS'95]

- Actions are divided into two disjoint sets $\Sigma = E \cup H$, for external and internal actions respectively.
- External actions E are partitioned to two disjoint sets $E = E_o \cup E_i$, for triggering symbols, ranged over by $a!, b!, \ldots$, and triggered symbols, ranged over by $a?, b?, \ldots$.



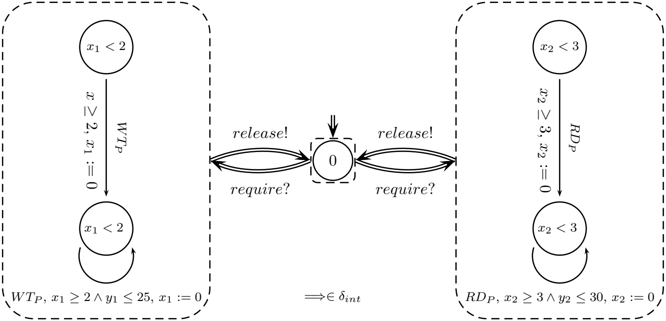


Why Need Controller Automata?

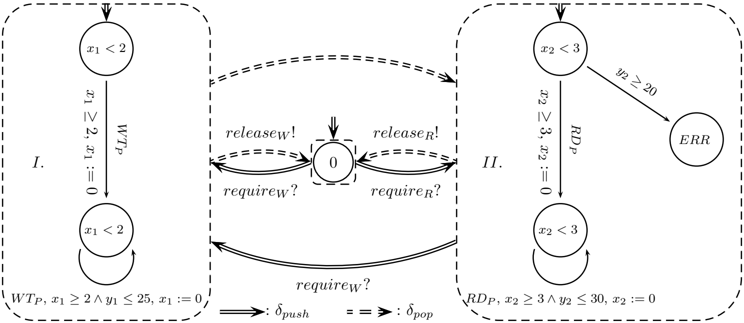
- Usually, mutex can be implemented by synchronization.
- However, in real-time system, time in an awaited component will elapse when it hangs up.
- There are three relations for two mutex components:
  - Competition e.g., Reading/Writing a shared buffer
  - Preemption and Resumption e.g., Interrupt
- Controller automata provide global controls among a group of timed automata.

Controller Automata

- Controller automata provide transitions for timed automata that represents different components.
- There are three kinds of transitions, push, pop and internal actions.

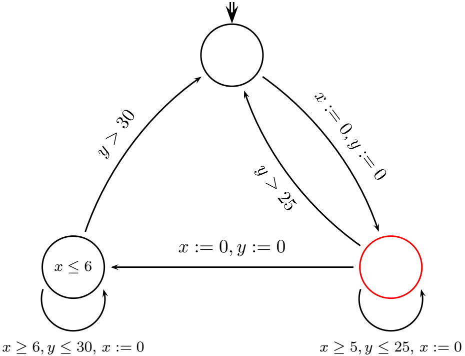


An Example: Reading/Writing with Priority



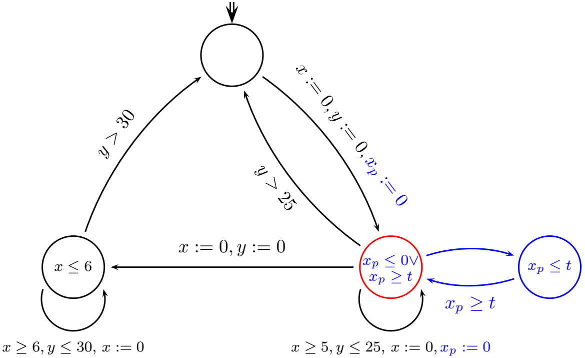
Time Lag in Timed Automata

- When a timed automaton is preempted by another one, the system will stop running current timed automaton, store the current status, and begin to run the latter timed automaton.
- A time lag adds a location and a fresh clock to wait a certain time when preempted by another timed automata.

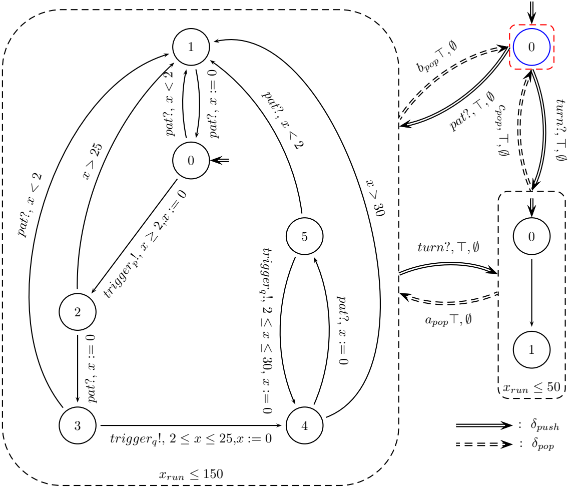


Time Lag in Timed Automata

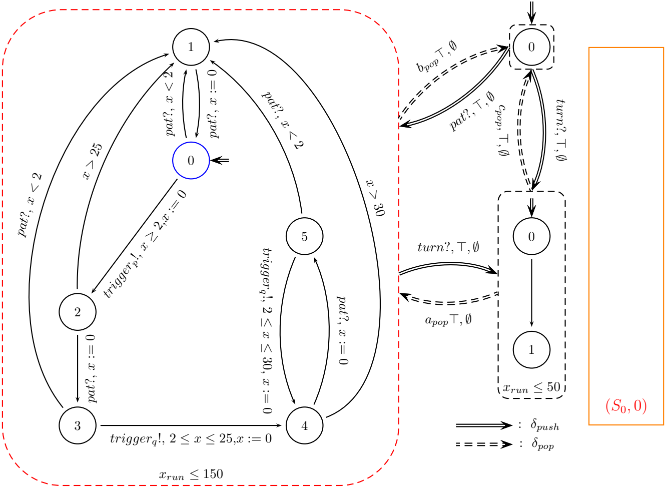
- When a timed automaton is preempted by another one, the system will stop running current timed automaton, store the current status, and begin to run the latter timed automaton.
- A time lag adds a location and a fresh clock to wait a certain time when preempted by another timed automata.



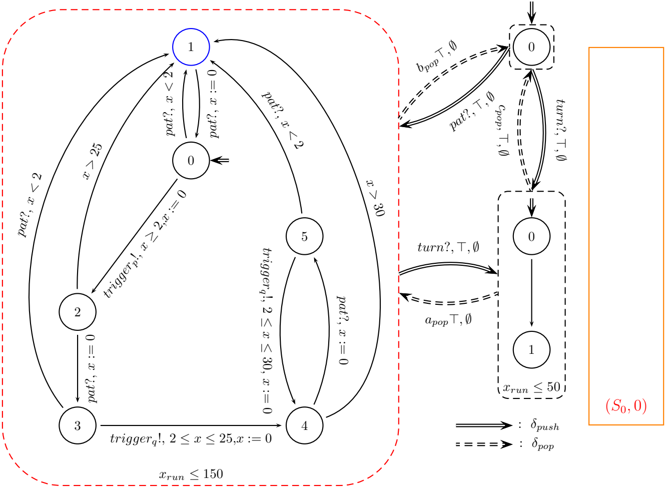
Running Controller Automata



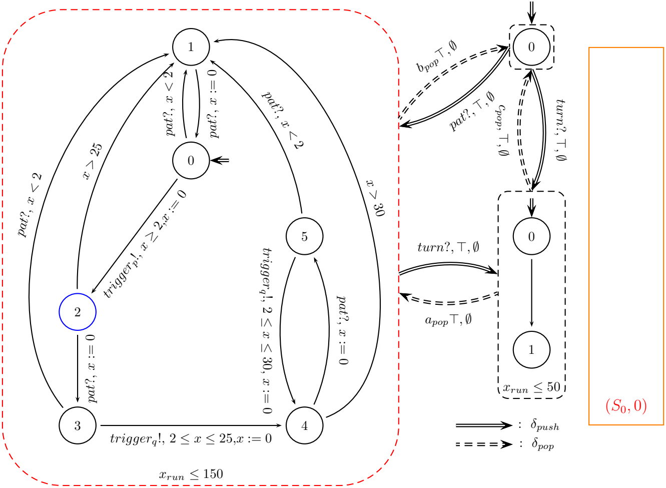
Running Controller Automata



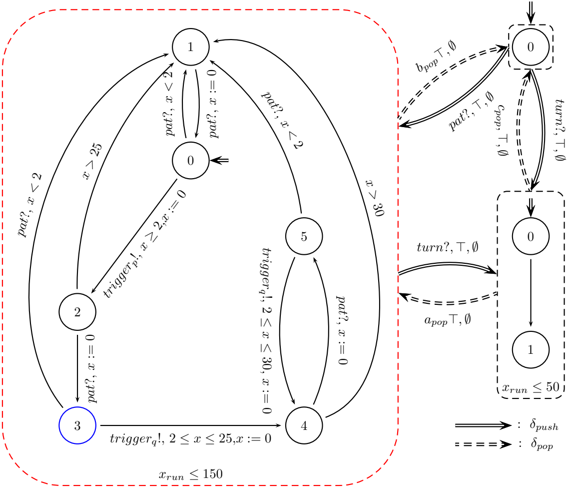
Running Controller Automata



Running Controller Automata

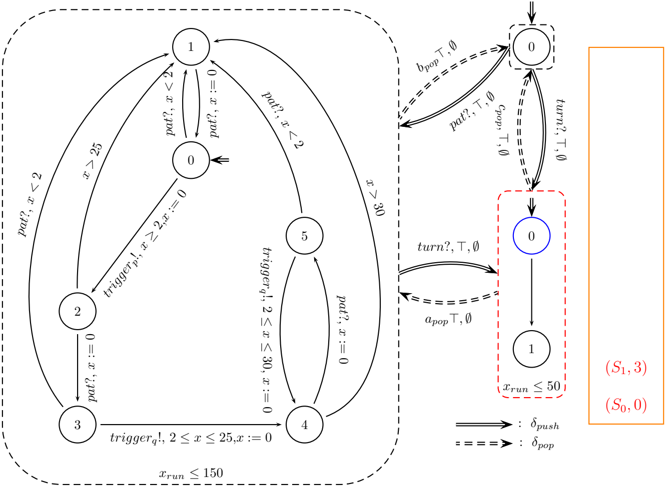


Running Controller Automata

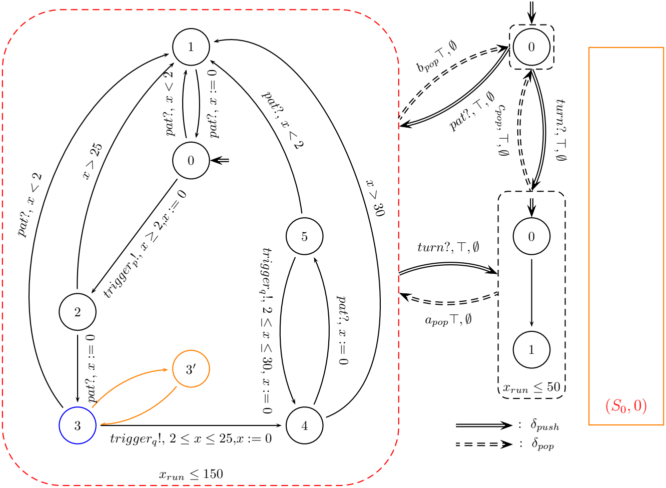


$(S_0, 0)$

Running Controller Automata



Running Controller Automata



Decidability Problems of Controller Automata

- Some comments...
  - controller automata are not beyond timed (pushdown) automata...
  - controller automata are stopwatch pushdown automata...
- Controller automata are less expressive than stopwatch automata
  - Fact. the frozen clocks are kept zero in CA.
- The decidability problems (e.g. reachability problem) of controller automata are in general undecidable.
  - Infinite insertion of fresh clocks and control locations.
- With a strict partial order on the state, an ordered controller automaton can be translated to a timed automaton.

Conclusion

- Controller automata are introduced, to perform global control on complex real-time systems.
- Analysis techniques (e.g. reachability) of controller automata are investigated.
- Future work:
  - Theoretical approaches: to investigate the languages category recognized by controller automata.
  - Practical approaches: to verify properties for complex real-time systems, e.g. liveness
  - Implementation work: translate an OCA to a timed automaton recognized by UPPAAL.

Thank You!

li.g@sjtu.edu.cn