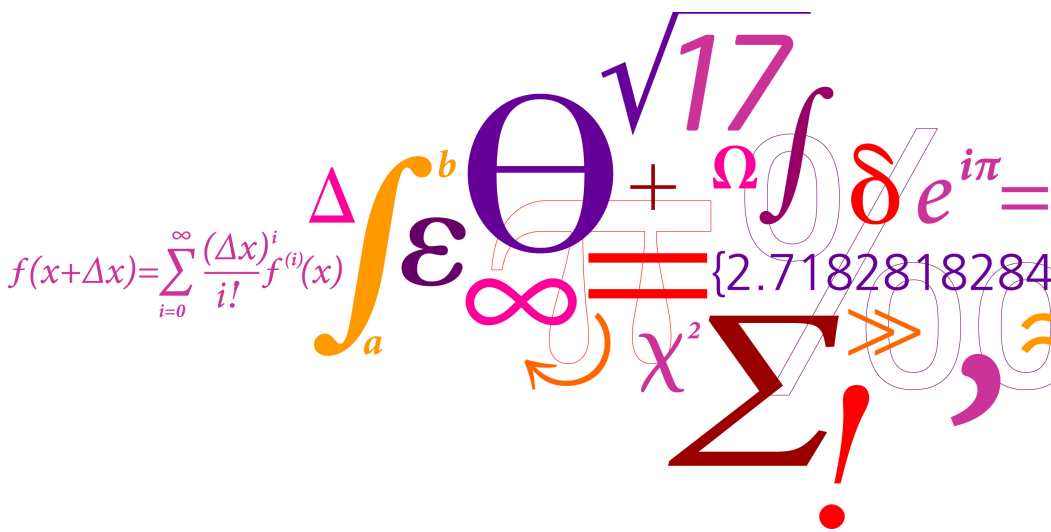
02224 Real-time systems

Timed Games, Controller Synthesis and UPPAAL-TIGA

Michael R. Hansen



DTU Compute
Department of Applied Mathematics and Computer Science

Model Checking vs Controllor Synthesis

Model Checking ($M \models \phi$)

- Given system model M and
- a property ϕ,

check (automatically) whether the behaviours of M satisfy ϕ.

Controller Synthesis

- Given environment model S,
- a model C describing the moves the controller *can* do,
- a property ϕ,

find a strategy S_c for the controller so that $S_c || S \models \phi$
or show there is not such strategy.

The right code is generated automatically

Model Checking vs Controllor Synthesis

Model Checking ($M \models \phi$)

- Given system model M and
- a property ϕ,

check (automatically) whether the behaviours of M satisfy ϕ.

Controller Synthesis

- Given environment model S,
- a model C describing the moves the controller *can* do,
- a property ϕ,

find a strategy S_c for the controller so that $S_c \| S \models \phi$
or show there is not such strategy.

The right code is generated automatically

Model Checking vs Controllor Synthesis

Model Checking ($M \models \phi$)

- Given system model M and
- a property ϕ,

check (automatically) whether the behaviours of M satisfy ϕ.

Controller Synthesis

- Given environment model S,
- a model C describing the moves the controller *can* do,
- a property ϕ,

find a strategy S_c for the controller so that $S_c \| S \models \phi$
or show there is not such strategy.

The right code is generated automatically

Timed Game Automata

- Introduced by Maler, Phueli and Sifakis in 1995.
- Controller continuously observes the system and can perform two actions
  - wait (that is, delay)
  - take a controllable move (thereby preventing delay)
- A 2-player game:
  Controller moves against uncontrollable environment moves
- Reachability games: $control : A <> Win$, where Win is a state formula describing the winning states.
- Safety games: $control : A[\,]\ not\ Lose$ where $Lose$ is a state formula describing the losing states of the controller.
- Memoryless strategy: $State \rightarrow Action$

Uppaal-TIGA was introduced in 2005 by
Cassez, David, Fleury, Larsen and Lime

Timed Game Automata

- Introduced by Maler, Phueli and Sifakis in 1995.
- Controller continuously observes the system and can perform two actions
  - wait (that is, delay)
  - take a controllable move (thereby preventing delay)
- A 2-player game:
  Controller moves against uncontrollable environment moves
- Reachability games: *control* : *A* <> *Win*, where *Win* is a state formula describing the winning states.
- Safety games: *control* : *A*[ ] *not Lose* where *Lose* is a state formula describing the losing states of the controller.
- Memoryless strategy: *State* $\rightarrow$ *Action*

Uppaal-TIGA was introduced in 2005 by
Cassez, David, Fleury, Larsen and Lime

Timed Game Automata

- Introduced by Maler, Phueli and Sifakis in 1995.
- Controller continuously observes the system and can perform two actions
  - wait (that is, delay)
  - take a controllable move (thereby preventing delay)
- A 2-player game:
  Controller moves against uncontrollable environment moves
- Reachability games: *control* : *A* <> *Win*, where *Win* is a state formula describing the winning states.
- Safety games: *control* : *A*[ ] *not Lose* where *Lose* is a state formula describing the losing states of the controller.
- Memoryless strategy: *State* $\rightarrow$ *Action*

Uppaal-TIGA was introduced in 2005 by
Cassez, David, Fleury, Larsen and Lime

Timed Game Automata

- Introduced by Maler, Phueli and Sifakis in 1995.
- Controller continuously observes the system and can perform two actions
  - wait (that is, delay)
  - take a controllable move (thereby preventing delay)
- A 2-player game:
  Controller moves against uncontrollable environment moves
- Reachability games: *control* : *A* <> *Win*, where *Win* is a state formula describing the winning states.
- Safety games: *control* : *A*[ ] *not Lose* where *Lose* is a state formula describing the losing states of the controller.
- Memoryless strategy: *State* $\rightarrow$ *Action*

Uppaal-TIGA was introduced in 2005 by
Cassez, David, Fleury, Larsen and Lime

Timed Game Automata

- Introduced by Maler, Phueli and Sifakis in 1995.
- Controller continuously observes the system and can perform two actions
  - wait (that is, delay)
  - take a controllable move (thereby preventing delay)
- A 2-player game:
  Controller moves against uncontrollable environment moves
- Reachability games: *control* : *A* <> *Win*, where *Win* is a state formula describing the winning states.
- Safety games: *control* : *A*[ ] *not Lose* where *Lose* is a state formula describing the losing states of the controller.
- Memoryless strategy: *State* $\rightarrow$ *Action*

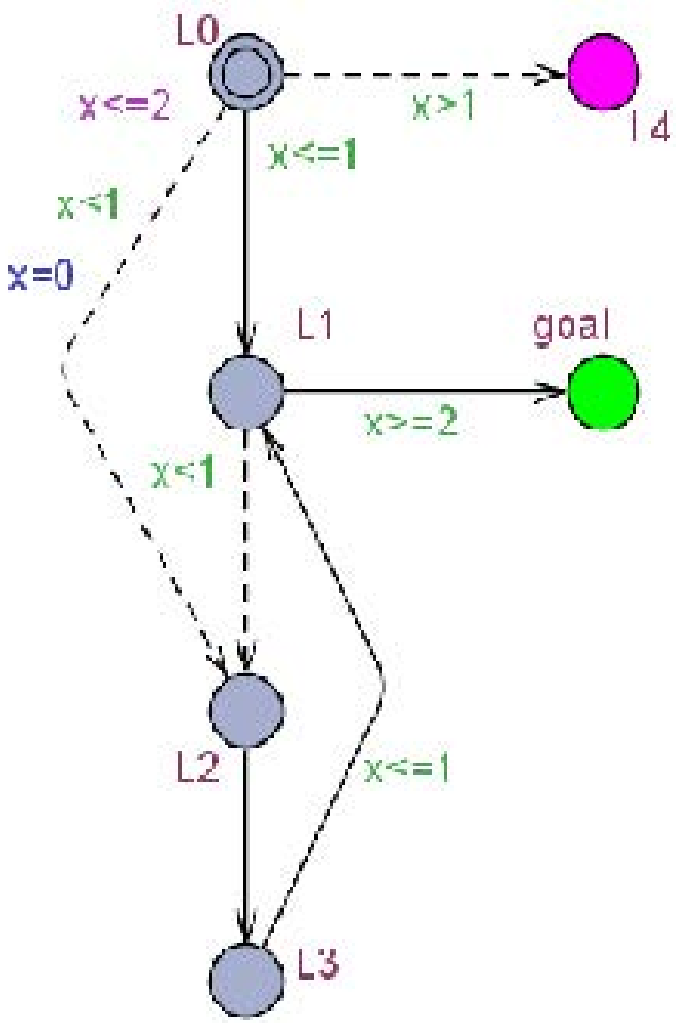
Uppaal-TIGA was introduced in 2005 by
Cassez, David, Fleury, Larsen and Lime

Timed Game Automata

- Introduced by Maler, Phueli and Sifakis in 1995.
- Controller continuously observes the system and can perform two actions
  - wait (that is, delay)
  - take a controllable move (thereby preventing delay)
- A 2-player game:
  Controller moves against uncontrollable environment moves
- Reachability games: *control* : *A* <> *Win*, where *Win* is a state formula describing the winning states.
- Safety games: *control* : *A*[ ] *not Lose* where *Lose* is a state formula describing the losing states of the controller.
- Memoryless strategy: *State* $\rightarrow$ *Action*

Uppaal-TIGA was introduced in 2005 by
Cassez, David, Fleury, Larsen and Lime

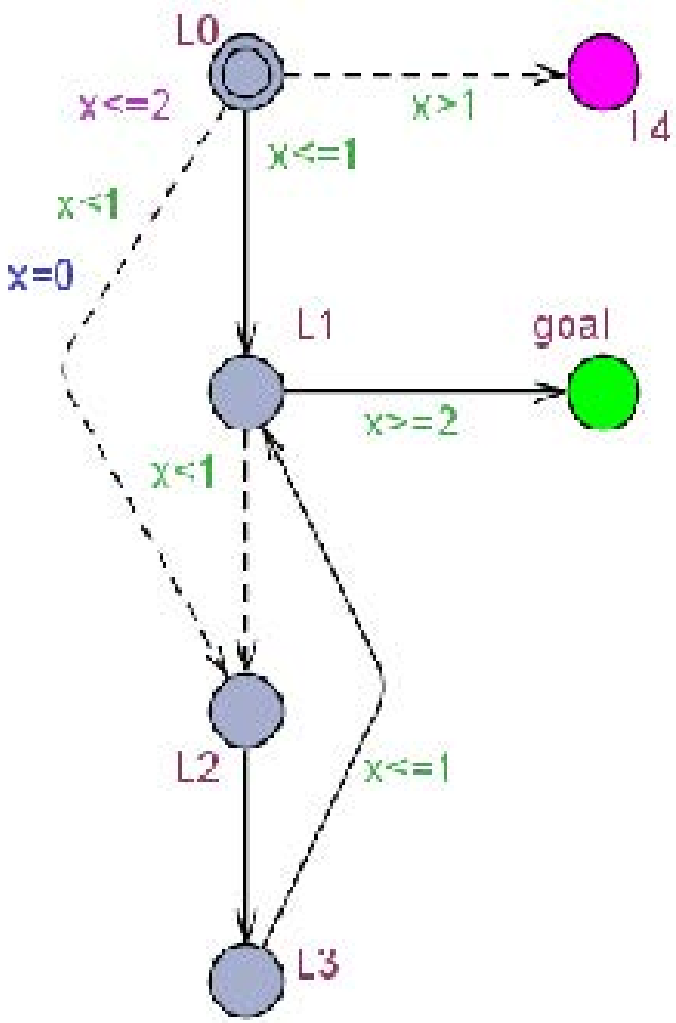
Timed Game Automata: An example



- Solid transitions are controlled
- dashed are uncontrolled (environment) transitions
- Priority to environment transitions
- Reachability objective: *control* : $A <> goal$
- Control can, in a state, chose to wait (delay) or chose to take a controllable transition
- Reachability and Safety Games are decidable. Memoryless and region-based strategies are sufficient. [AMS98]

Figure: From Uppaal-TIGA tutorial

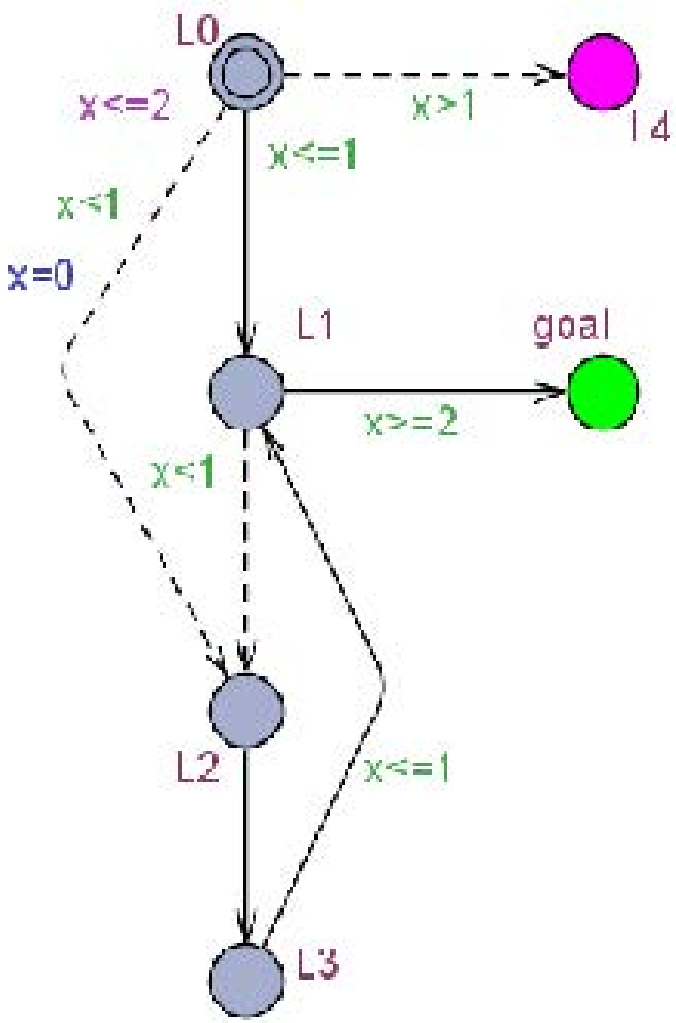
Timed Game Automata: An example



- Solid transitions are controlled
- dashed are uncontrolled (environment) transitions
- Priority to environment transitions
- Reachability objective: *control* : $A <> goal$
- Control can, in a state, chose to wait (delay) or chose to take a controllable transition
- Reachability and Safety Games are decidable. Memoryless and region-based strategies are sufficient. [AMS98]

Figure: From Uppaal-TIGA tutorial

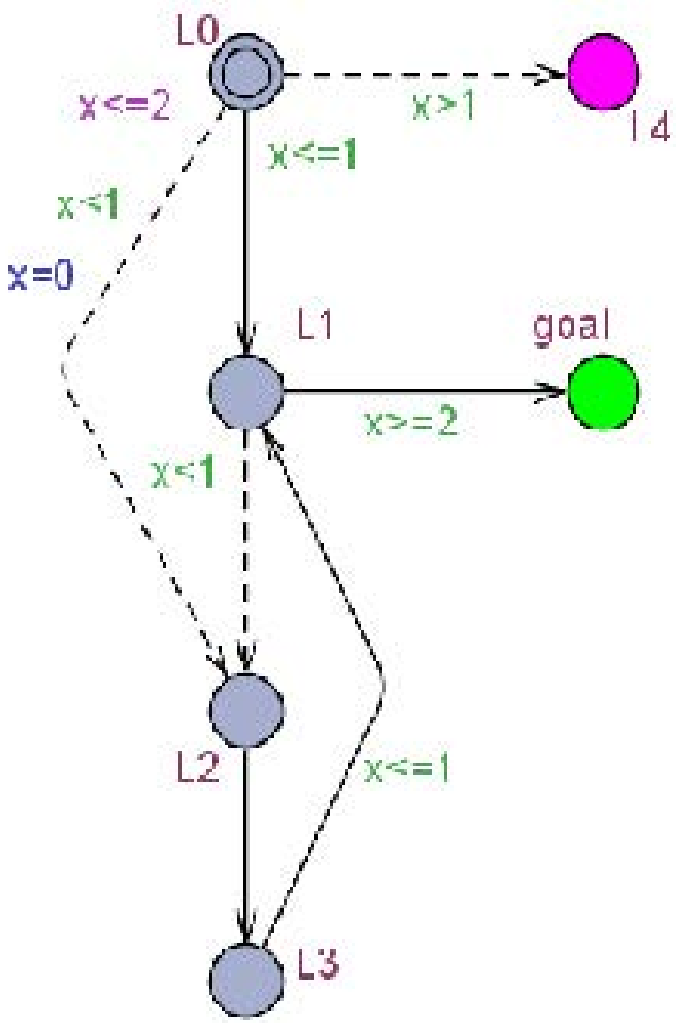
Timed Game Automata: An example



- Solid transitions are controlled
- dashed are uncontrolled (environment) transitions
- Priority to environment transitions
- Reachability objective: *control : A <> goal*
- Control can, in a state, chose to wait (delay) or chose to take a controllable transition
- Reachability and Safety Games are decidable. Memoryless and region-based strategies are sufficient. [AMS98]

Figure: From Uppaal-TIGA tutorial

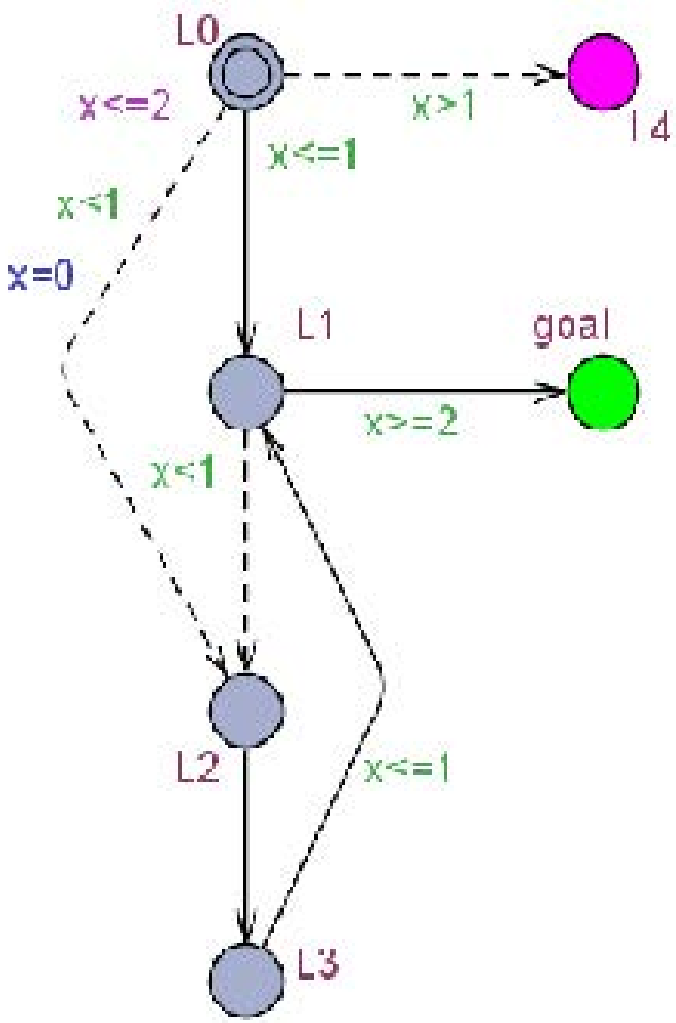
Timed Game Automata: An example



- Solid transitions are controlled
- dashed are uncontrolled (environment) transitions
- Priority to environment transitions
- Reachability objective: *control : A <> goal*
- Control can, in a state, chose to wait (delay) or chose to take a controllable transition
- Reachability and Safety Games are decidable. Memoryless and region-based strategies are sufficient. [AMS98]

Figure: From Uppaal-TIGA tutorial

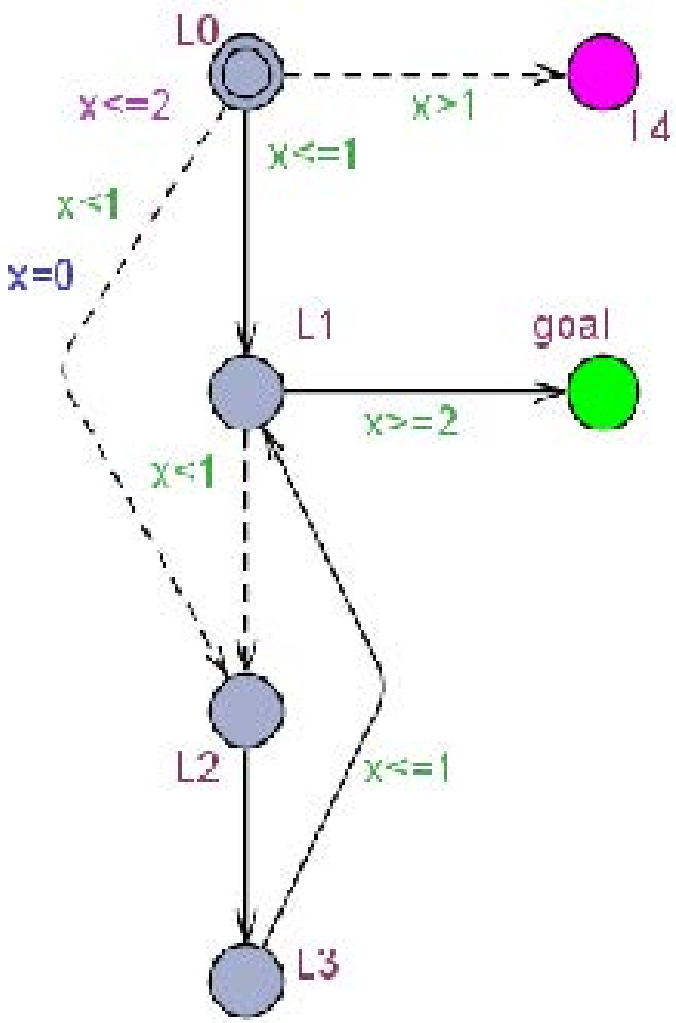
Timed Game Automata: An example



- Solid transitions are controlled
- dashed are uncontrolled (environment) transitions
- Priority to environment transitions
- Reachability objective: *control : A <> goal*
- Control can, in a state, chose to wait (delay) or chose to take a controllable transition
- Reachability and Safety Games are decidable. Memoryless and region-based strategies are sufficient. [AMS98]

Figure: From Uppaal-TIGA tutorial

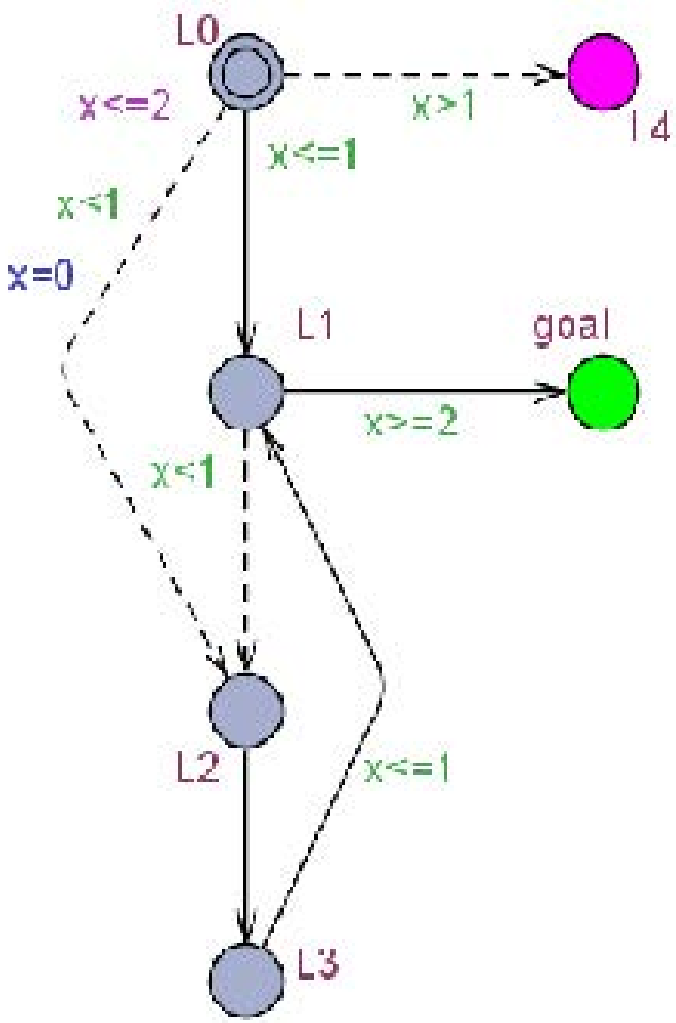
Timed Game Automata: An example



- Solid transitions are controlled
- dashed are uncontrolled (environment) transitions
- Priority to environment transitions
- Reachability objective: *control : A <> goal*
- Control can, in a state, chose to wait (delay) or chose to take a controllable transition
- Reachability and Safety Games are decidable. Memoryless and region-based strategies are sufficient. [AMS98]

Figure: From Uppaal-TIGA tutorial

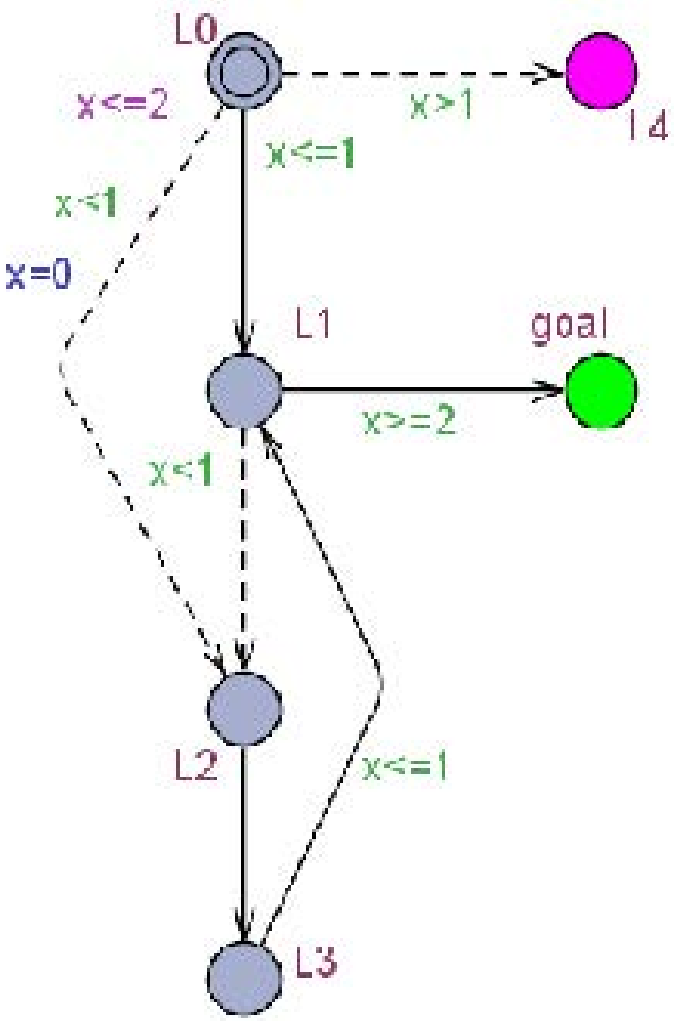
Timed Game Automata: An example



- Solid transitions are controlled
- dashed are uncontrolled (environment) transitions
- Priority to environment transitions
- Reachability objective: *control : A <> goal*
- Control can, in a state, chose to wait (delay) or chose to take a controllable transition
- Reachability and Safety Games are decidable. Memoryless and region-based strategies are sufficient. [AMS98]

Figure: From Uppaal-TIGA tutorial

Backward computation of Winning States: An example



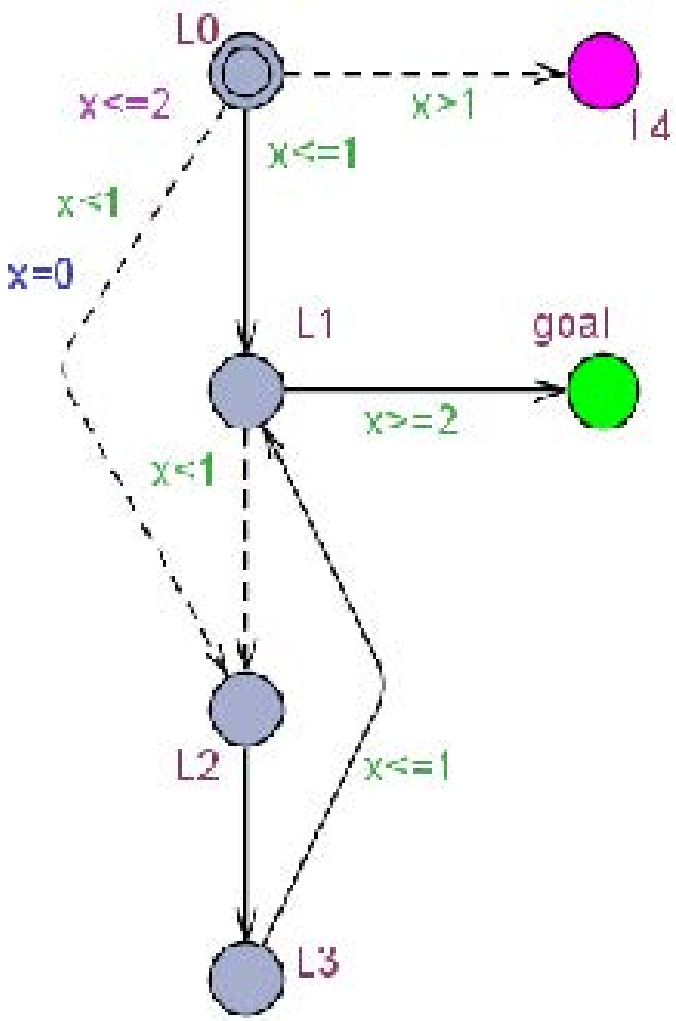
L0 : $\{\ \}$

L1 : $\{x \mid 1 \leq x\}$

L2 : $\{\}$

L3 : $\{\ \}$

Backward computation of Winning States: An example



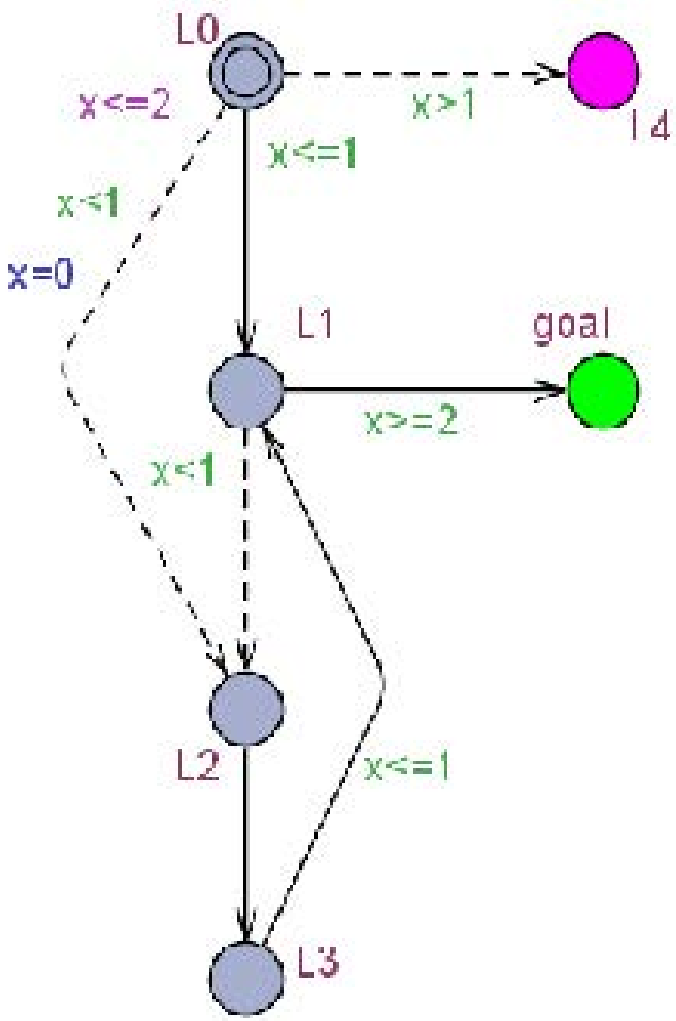
L0 : $\{\ \}$

L1 : $\{x \mid 1 \leq x\}$

L2 : $\{\}$

L3 : $\{x \mid 0 \leq x \leq 1\}$

Backward computation of Winning States: An example



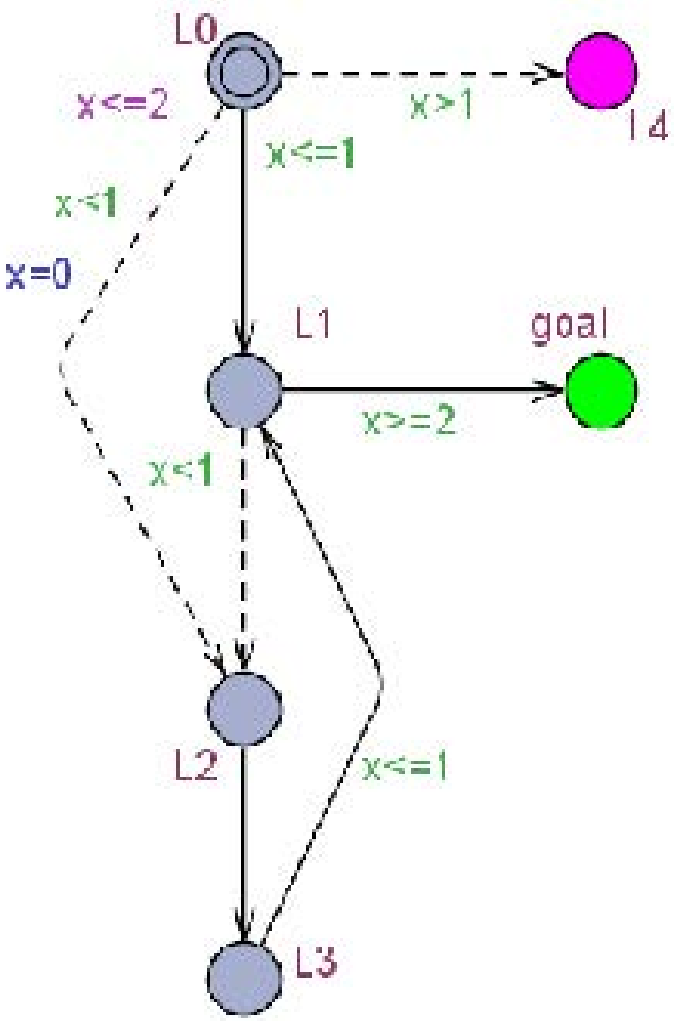
L0 : $\{\ \}$

L1 : $\{x \mid 1 \leq x\}$

L2 : $\{x \mid 0 \leq x \leq 1\}$

L3 : $\{x \mid 0 \leq x \leq 1\}$

Backward computation of Winning States: An example



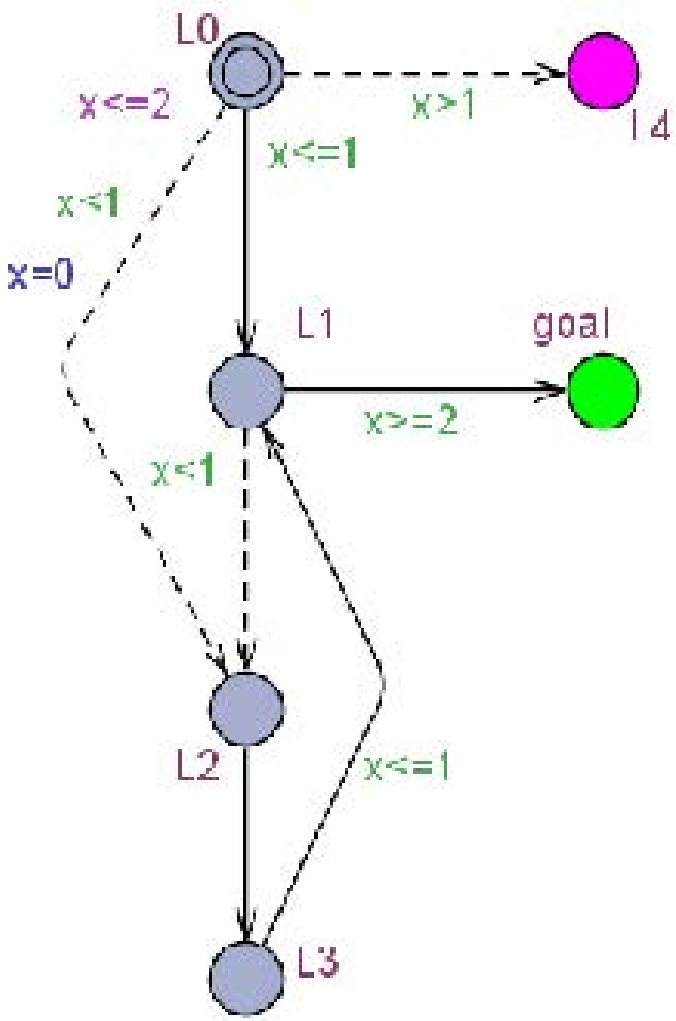
L0 : $\{\ \}$

L1 : $\{x \mid 0 \leq x < 1\} \cup \{x \mid 1 \leq x\}$

L2 : $\{x \mid 0 \leq x \leq 1\}$

L3 : $\{x \mid 0 \leq x \leq 1\}$

Backward computation of Winning States: An example



L0 : $\{x \mid 0 \leq x \leq 1\}$

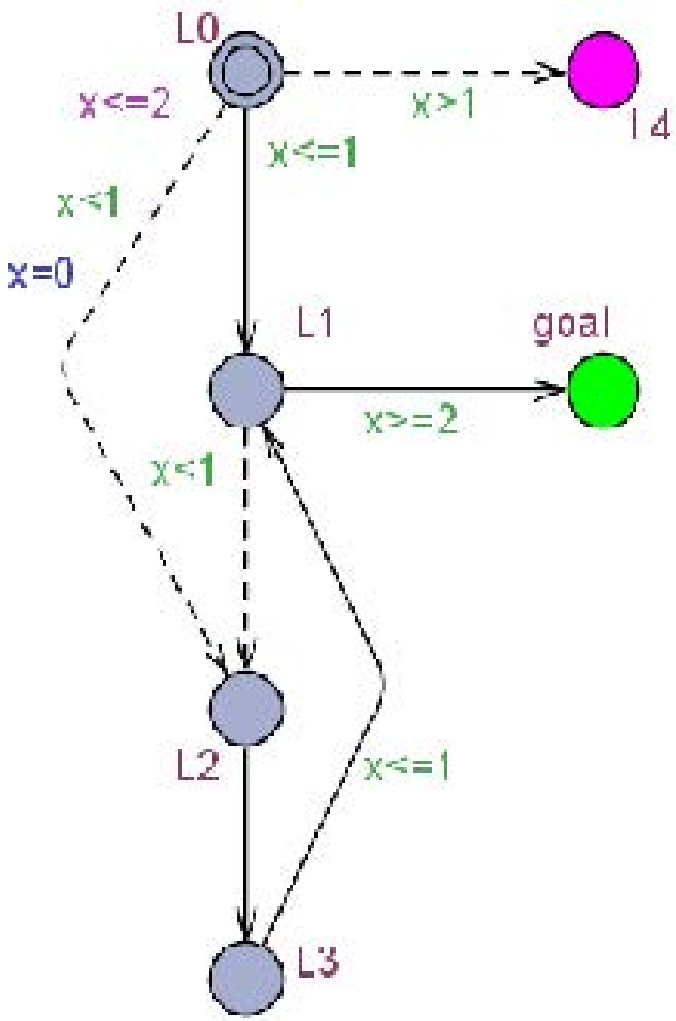
L1 : $\{x \mid 0 \leq x < 1\} \cup \{x \mid 1 \leq x\}$

L2 : $\{x \mid 0 \leq x \leq 1\}$

L3 : $\{x \mid 0 \leq x \leq 1\}$

Computation of Winning Strategies: An example

Reachability Games `control A <> goal`: Take actions that lead to winning states. (Partition states to guarantee progress)



L0, $x < 1$: wait
L0, $x = 1$: goto L1

L1, $x < 2$: wait
L1, $x \geq 2$: goto goal

L2, $x \leq 1$: goto L3

L3, $x < 1$: Wait
L3, $x = 1$: goto L1

UPPAAL TIGA's Query Language

Reachability properties:

- `control: A[p U q]`
- `control: A<> p`

Safety properties:

- `control: A[p W q]`
- `control: A[] p`

UPPAAL TIGA's Query Language

Reachability properties:

- `control: A[p U q]`
- `control: A<> p`

Safety properties:

- `control: A[p W q]`
- `control: A[] p`