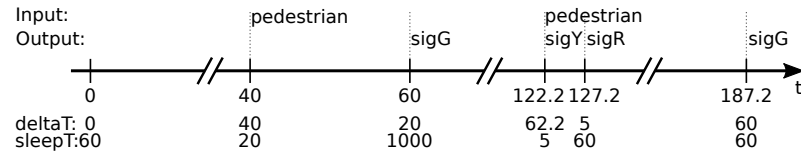


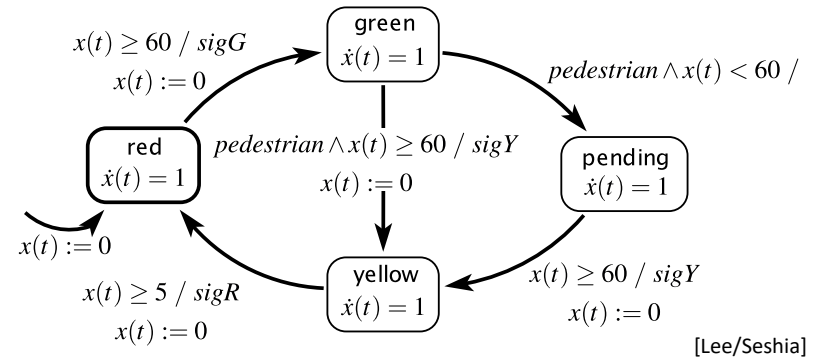
Traffic Light as Timed Automaton



Alexander Schulz Rosengarten, Reinhard von Hanxleden
Kiel University

Frédéric Mallet, Robert de Simone, Julien DeAntoni
INRIA Sophia Antipolis

continuous variable: $x(t) : \mathbb{R}$
inputs: pedestrian: pure
outputs: sigR, sigG, sigY: pure



[Lee/Seshia]

Alur, Dill, *A theory of timed automata*, Theoretical Computer Science, 1994

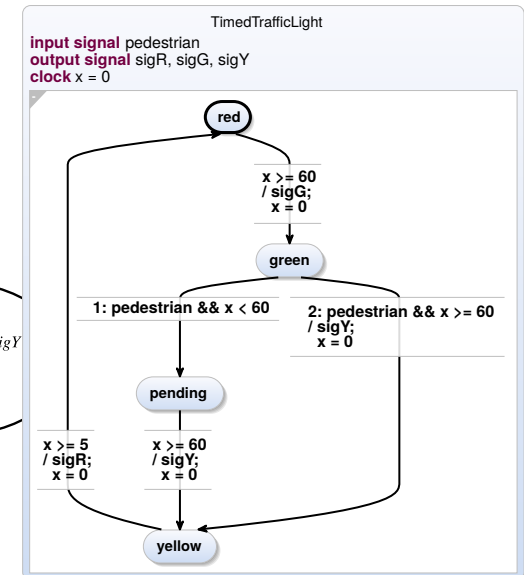
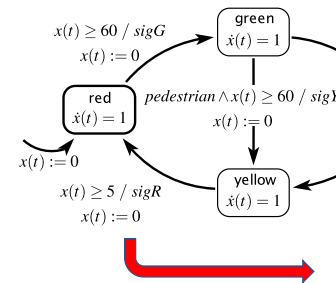
Lecture 17: Time in SCCharts

Alexander Schulz Rosengarten, Reinhard von Hanxleden
Kiel University

Frédéric Mallet, Robert de Simone, Julien DeAntoni
INRIA Sophia Antipolis

Traffic Light in SCCharts

continuous variable: $x(t) : \mathbb{R}$
inputs: pedestrian: pure
outputs: sigR, sigG, sigY: pure

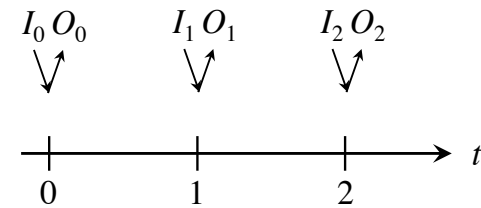


Roadmap

1. Traffic Light Example
2. Execution Models
3. Dynamic Ticks
4. Time in SCCharts: “clock”
5. Multiclocks in SCCharts: “period”
6. Demo

5

Discrete (Logical) Time in Synchronous Programming



- Synchrony Hypothesis:
Outputs are synchronous with inputs
- Computation "does not take time"
- Actual computation time does not influence result
- Sequence of outputs **determined** by inputs

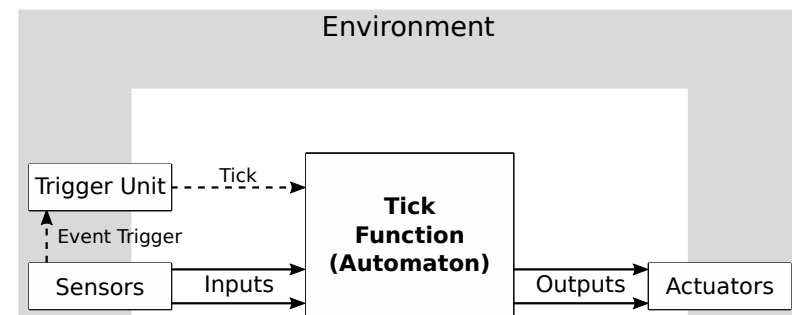
7

Roadmap

1. Traffic Light Example
2. Execution Models
3. Dynamic Ticks
4. Time in SCCharts: “clock”
5. Multiclocks in SCCharts: “period”
6. Demo

6

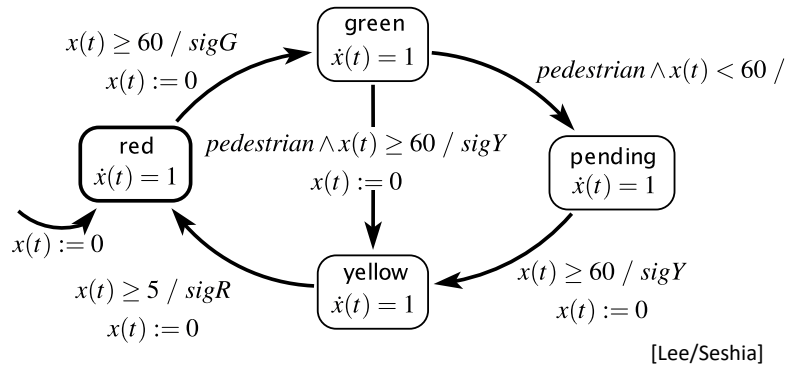
Event-Triggered Execution



8

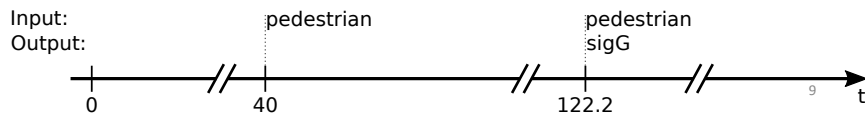
continuous variable: $x(t) : \mathbb{R}$
 inputs: pedestrian: pure
 outputs: sigR, sigG, sigY: pure

Assume pedestrian button pressed at $t = 40$ and $t = 122.2$

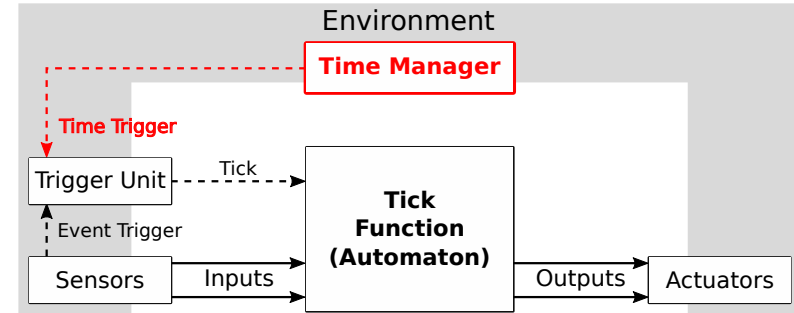


[Lee/Seshia]

Event-Triggered Execution, with initial tick at $t = 0$:



Time-Triggered Execution



11

Synchronous Execution

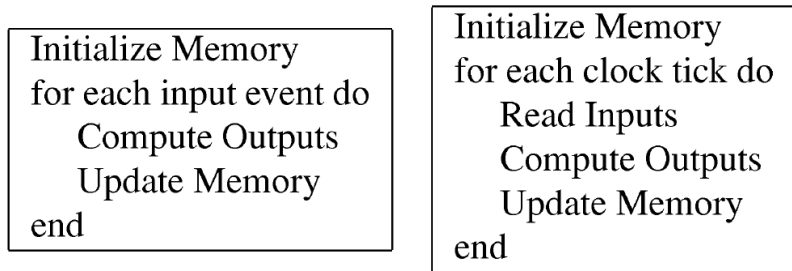


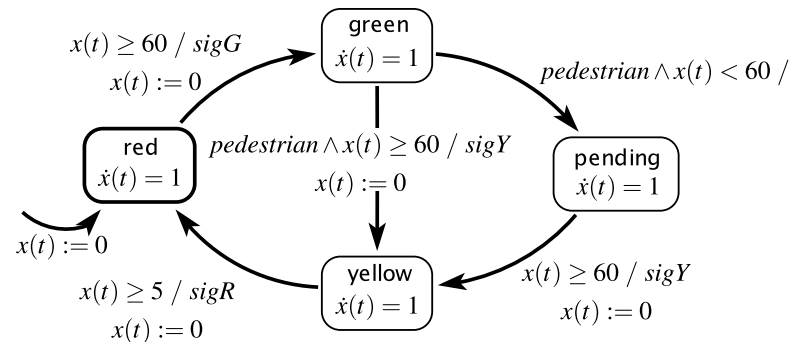
Fig. 1 Two common synchronous execution schemes: event driven (left) and sample driven (right).

[Benveniste et al., *The Synchronous Languages Twelve Years Later*, Proc. IEEE, 2003]

10

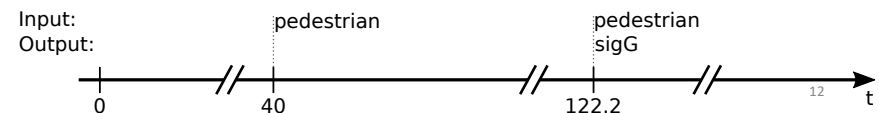
continuous variable: $x(t) : \mathbb{R}$
 inputs: pedestrian: pure
 outputs: sigR, sigG, sigY: pure

Assume pedestrian button pressed at $t = 40$ and $t = 122.2$



[Lee/Seshia]

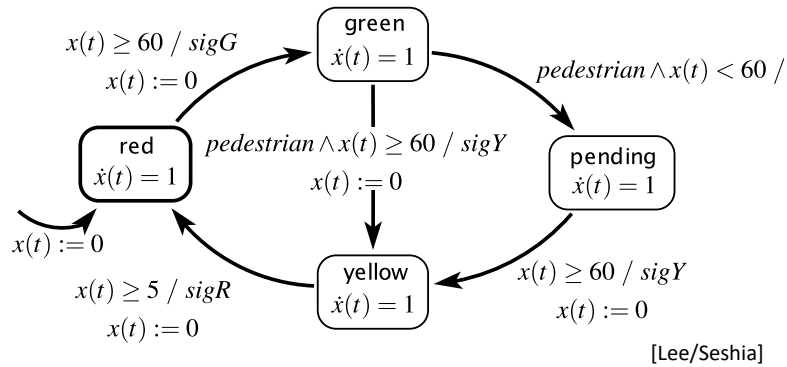
Recall: Event-Triggered Execution:



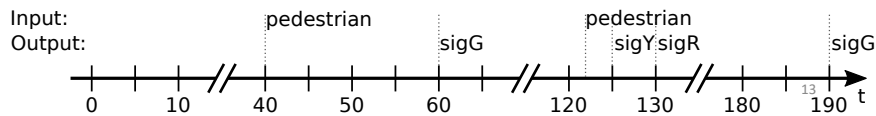
12

continuous variable: $x(t) : \mathbb{R}$
 inputs: pedestrian: pure
 outputs: sigR, sigG, sigY: pure

Assume pedestrian button pressed at $t = 40$ and $t = 122.2$



Time-Triggered Execution (every 5 sec):



Multiform Notion of Time

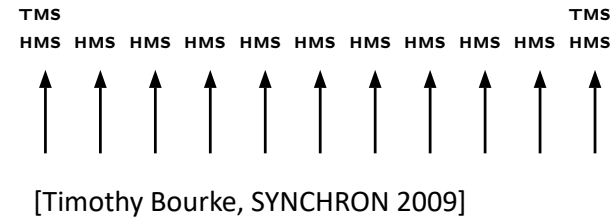
Only the simultaneity and precedence of events are considered.

This means that the physical time does not play any special role.

This is called multiform notion of time.

[<https://en.wikipedia.org/wiki/Esterel>]

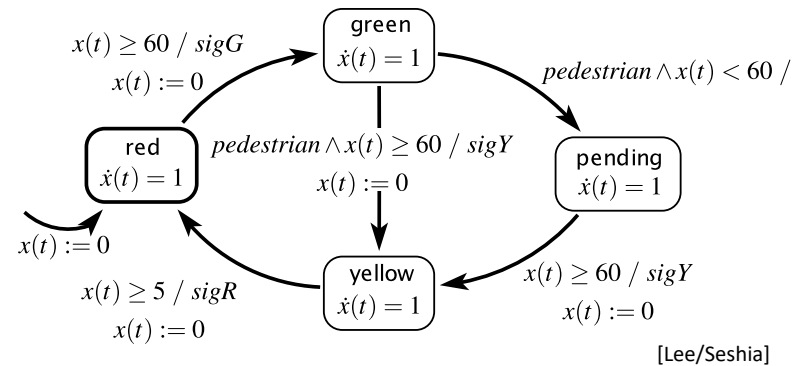
Packaging Physical Time as Events



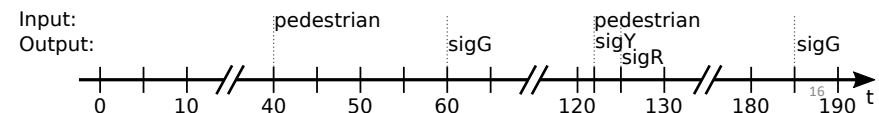
Event "HMS": 100 μsec have passed since last HMS
 Event "TMS": 1000 μsec have passed since last TMS

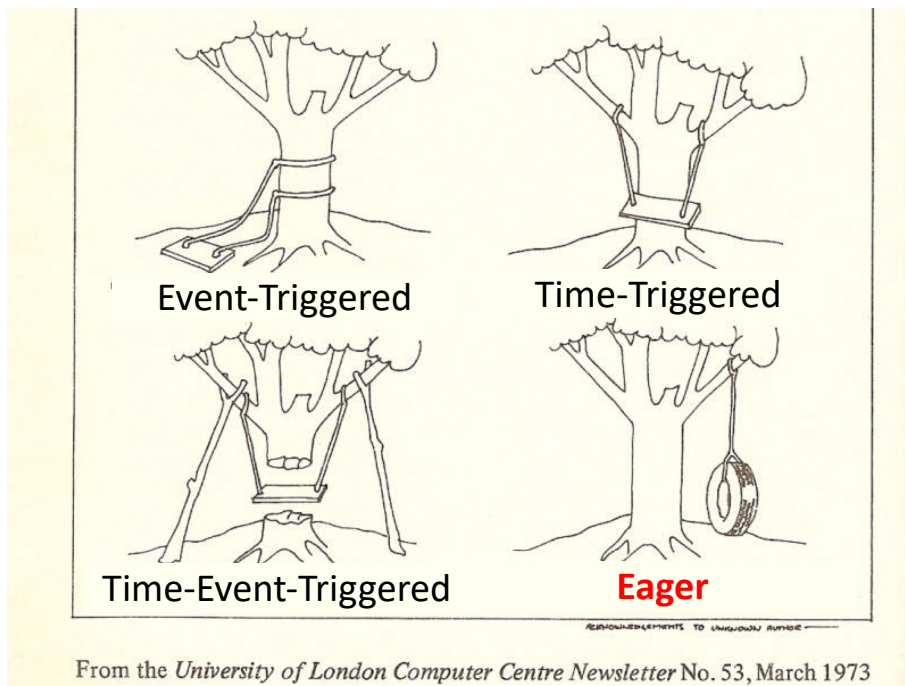
continuous variable: $x(t) : \mathbb{R}$
 inputs: pedestrian: pure
 outputs: sigR, sigG, sigY: pure

Assume pedestrian button pressed at $t = 40$ and $t = 122.2$



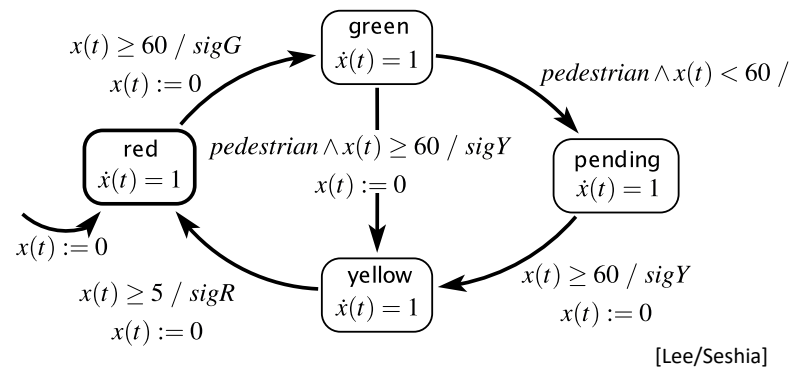
Time-Event-Triggered Execution, Multiform Time:



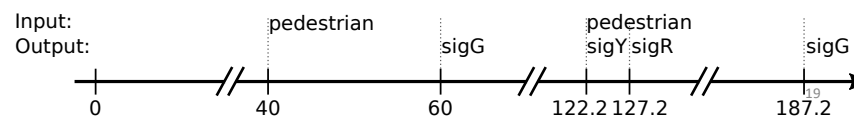


continuous variable: $x(t) : \mathbb{R}$
 inputs: pedestrian: pure
 outputs: sigR, sigG, sigY: pure

Assume pedestrian button pressed at $t = 40$ and $t = 122.2$



Eager Semantics:



What the User (Probably) Wanted

„We assume here that a transition is taken as soon as it is enabled. Other transition semantics are possible.“

[Lee/Seshia 2017]

We call this **eager** semantics.

Time in SCCharts – Requirements

1. Seamless fit into synchronous paradigm
 - Still deterministic behavior – outputs fully determined by inputs
 - No changes to underlying SC (Sequentially Constructive) MoC
2. Approximate eager semantics
 - Modulo run-time variations and imperfections of physical timers
3. Scalability
 - E.g., allow arbitrary number of (concurrent) timers
4. Fine granularity
 - Gcd may be arbitrarily small, w/o performance penalty
 - E.g., may have timeouts of 1 sec and 3.1415926 msec in same model
5. Time composability
 - E.g., waiting 1 sec. twice should mean the same as waiting 2 sec's once

Time in SCCharts – Requirements

6. Preserve temporal order and simultaneity
 - E.g., timers started in same tick and running same duration should expire in same tick
7. Minimize impact of physical timer variations
 - E.g., avoid accumulations of timer imperfections
8. Give application access to physical time and tick computation time
 - Facilitates e.g. load-dependent execution modes
9. Lean, application-independent interface to environment
 - E.g., interface should not change if number of timers changes
10. Fit into Single Language-Driven Incremental Compilation (SLIC) concept
 - New timing constructs are just syntactic sugar on top of existing SCCharts
 - Transforming away timing constructs requires only local changes
 - No changes needed to compilation back-end

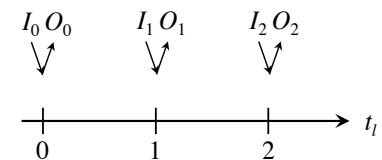


Roadmap

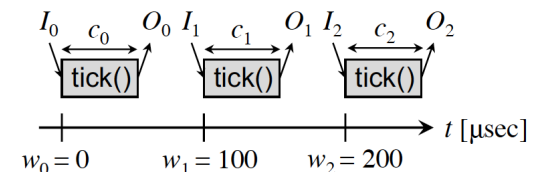
1. Traffic Light Example
2. Execution Models
3. Dynamic Ticks
4. Time in SCCharts: “clock”
5. Multiclocks in SCCharts: “period”
6. Demo

Dynamic Ticks

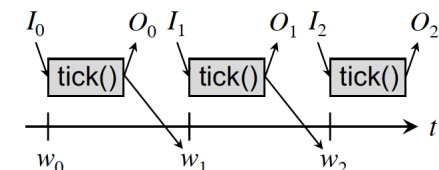
• Recall logical time:



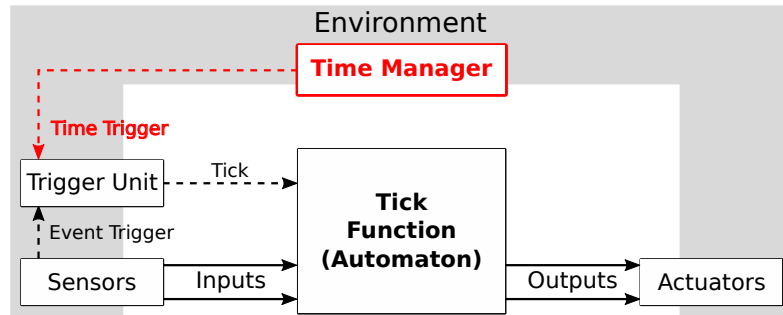
• Physical time, time-triggered:



• Physical time, dynamic ticks:

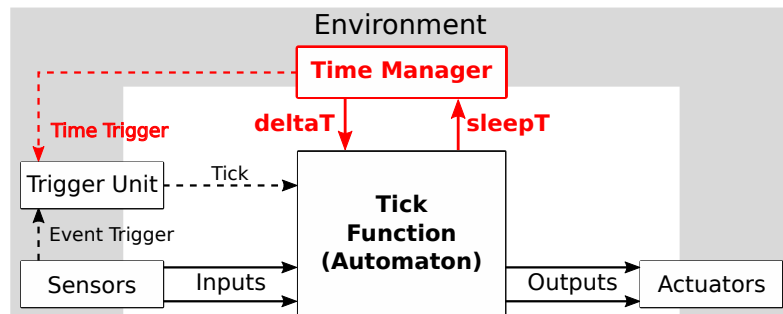


Recall: Time-Triggered Execution



25

Eager Execution with Dynamic Ticks

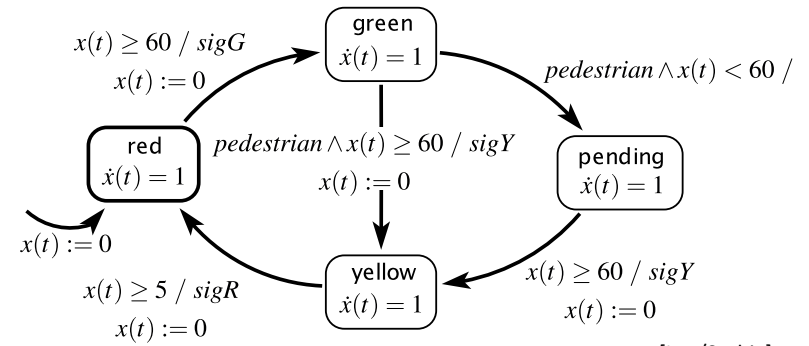


deltaT: Time since last tick
sleepT: Requested delay until next tick

26

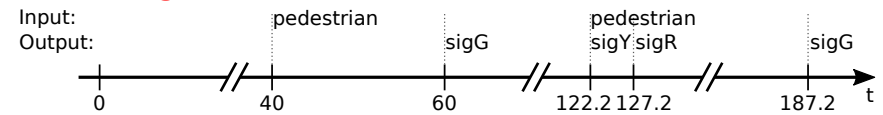
continuous variable: $x(t) : \mathbb{R}$
 inputs: pedestrian: pure
 outputs: sigR, sigG, sigY: pure

Assume pedestrian button pressed at $t = 40$ and $t = 122.2$



[Lee/Seshia]

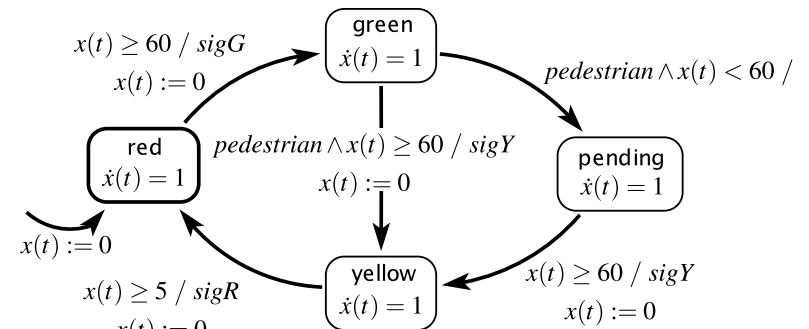
Recall: Eager Semantics



27

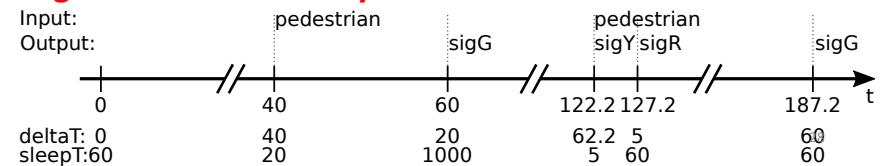
continuous variable: $x(t) : \mathbb{R}$
 inputs: pedestrian: pure
 outputs: sigR, sigG, sigY: pure

Assume pedestrian button pressed at $t = 40$ and $t = 122.2$



[Lee/Seshia]

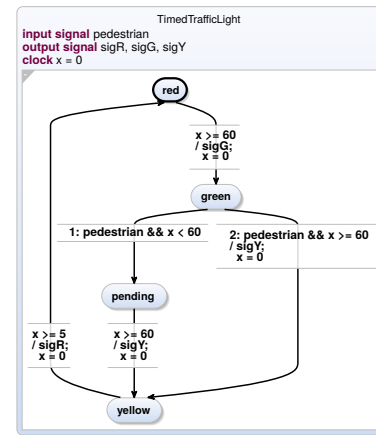
Eager Execution with Dynamic Ticks:



Multiform Notion of Time – Again!

- Semantically, treat clocks (time) as a unit-less number
- As in timed automata, clocks must satisfy *monotonicity* (modulo resets) and *progress*
- Current implementation maps time (clock variables) to an approximation of real numbers (float), interpreted as seconds
- However, could also map clocks to integers, interpreted as Euros spent, fathoms travelled, or beers consumed

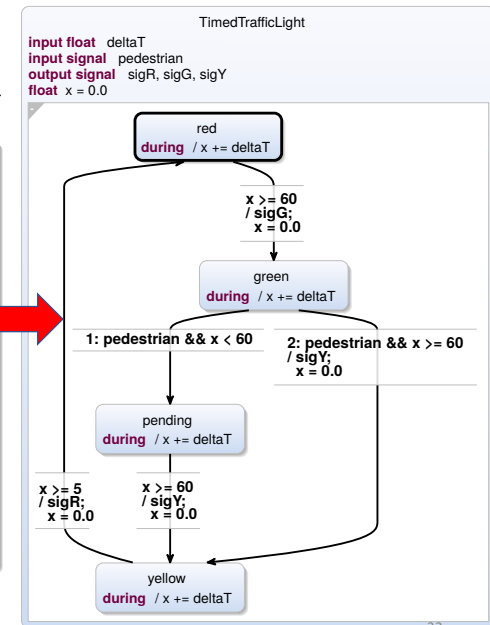
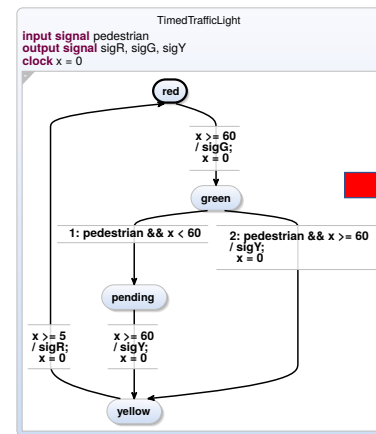
Recall: Traffic Light in SCCharts

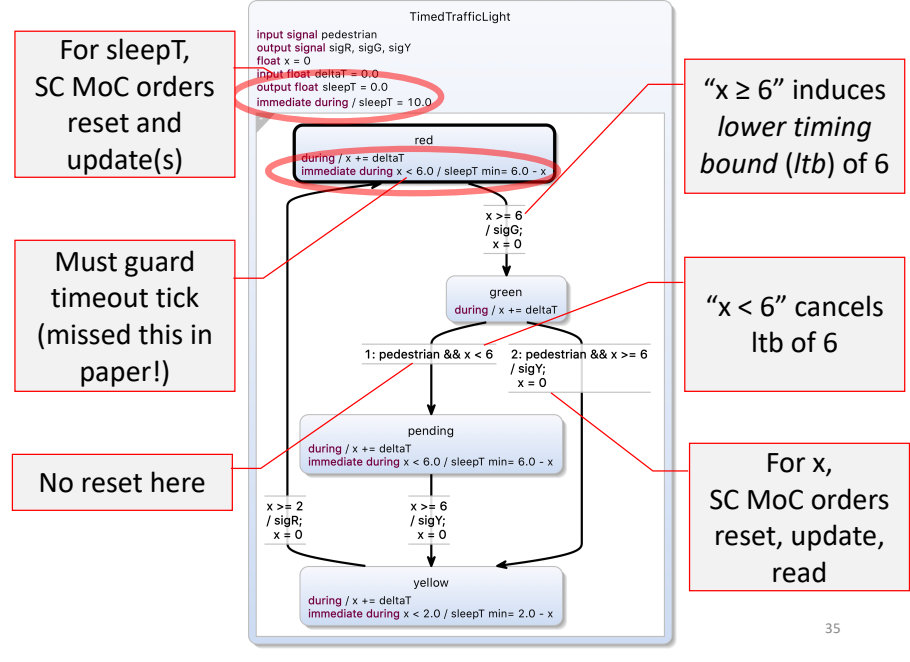
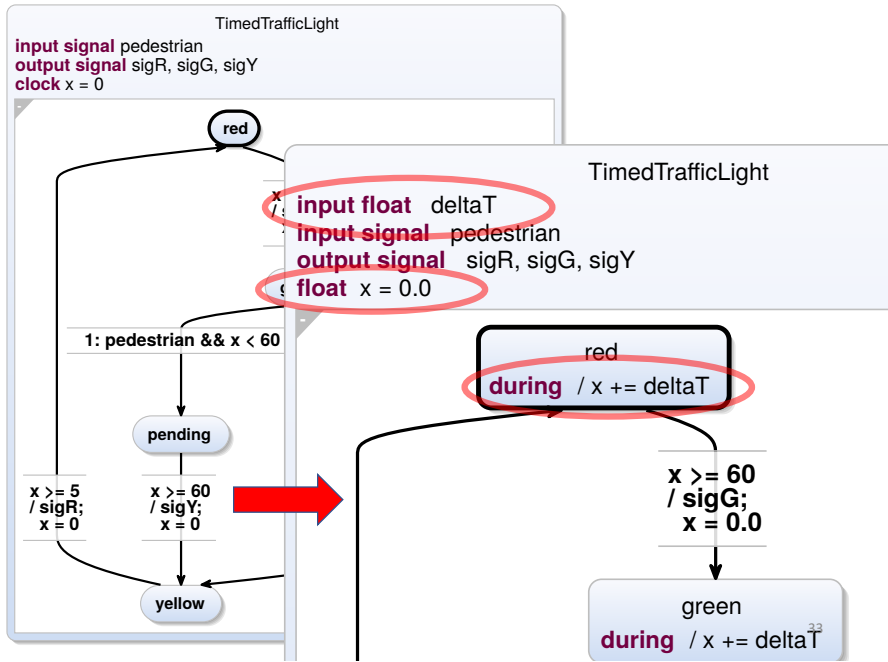


Roadmap

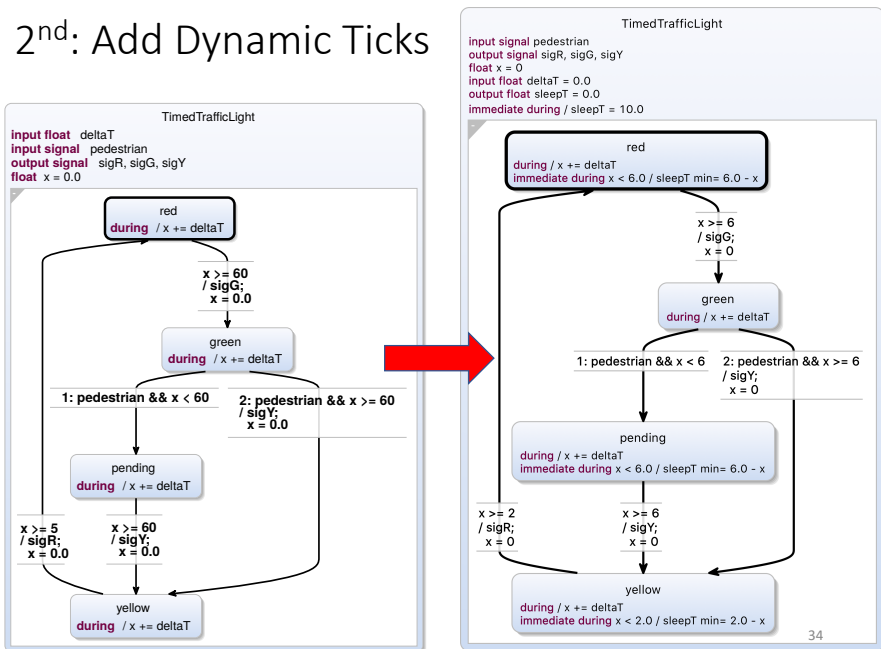
1. Traffic Light Example
2. Execution Models
3. Dynamic Ticks
4. Time in SCCharts: “clock”
5. Multiclocks in SCCharts: “period”
6. Demo

1st: Expand Clock





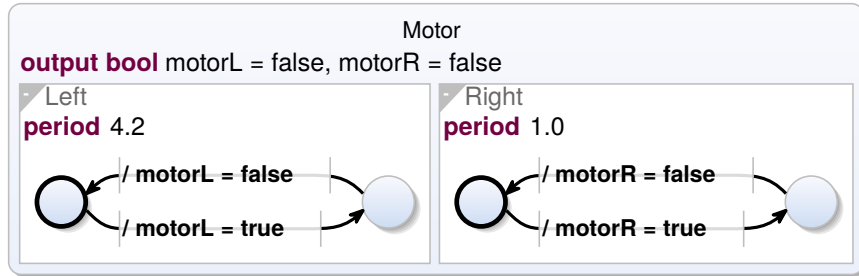
2nd: Add Dynamic Ticks



Roadmap

1. Traffic Light Example
2. Execution Models
3. Dynamic Ticks
4. Time in SCCharts: "clock"
5. Multiclocks in SCCharts: "period"
6. Demo

Multiclocks



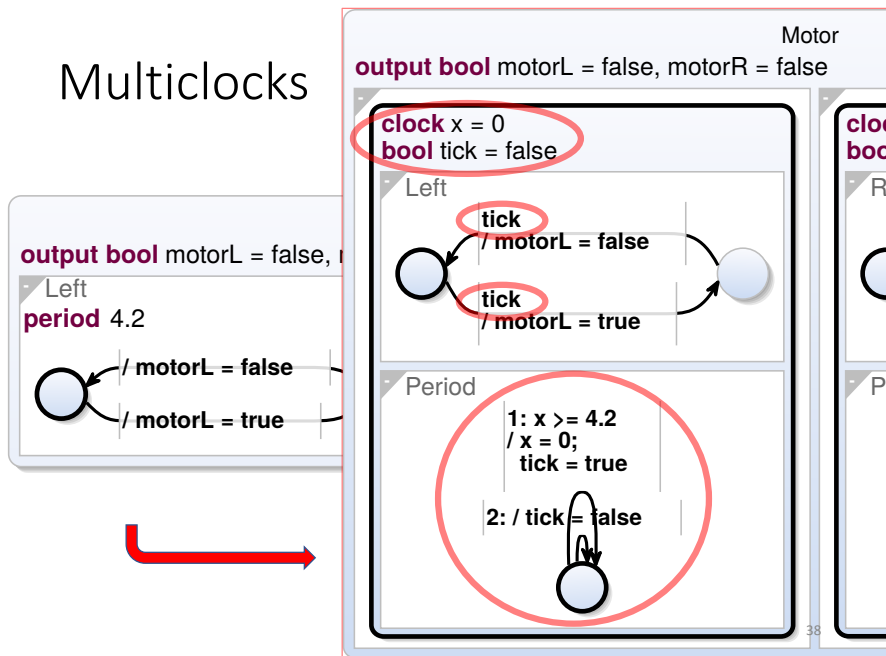
37

Roadmap

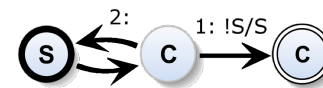
1. Traffic Light Example
2. Execution Models
3. Dynamic Ticks
4. Time in SCCharts: "clock"
5. Multiclocks in SCCharts: "period"
6. Demo

39

Multiclocks



38



SCCharts

<http://www.sccharts.com/>



KIELER

The Key to Efficient Modeling

<http://www.rtsys.informatik.uni-kiel.de/en/research/kieler>



Eclipse Layout Kernel

<https://www.eclipse.org/elk/>

All available as open source under EPL

40

