Computing DBF/RBF for Timed/Task Automata

Nan Guan

The Hong Kong Polytechnic University

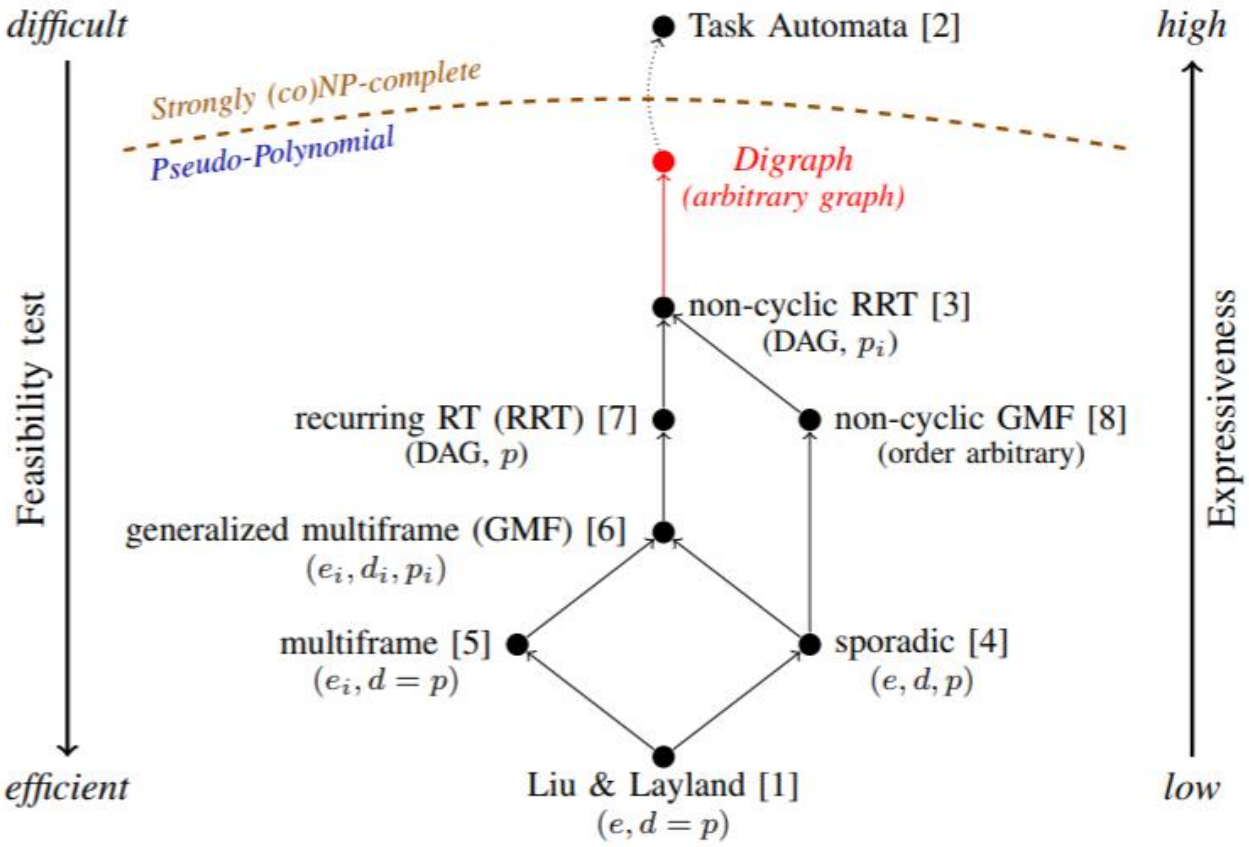
@RTSOPS, 2019 July, Paris

Outline

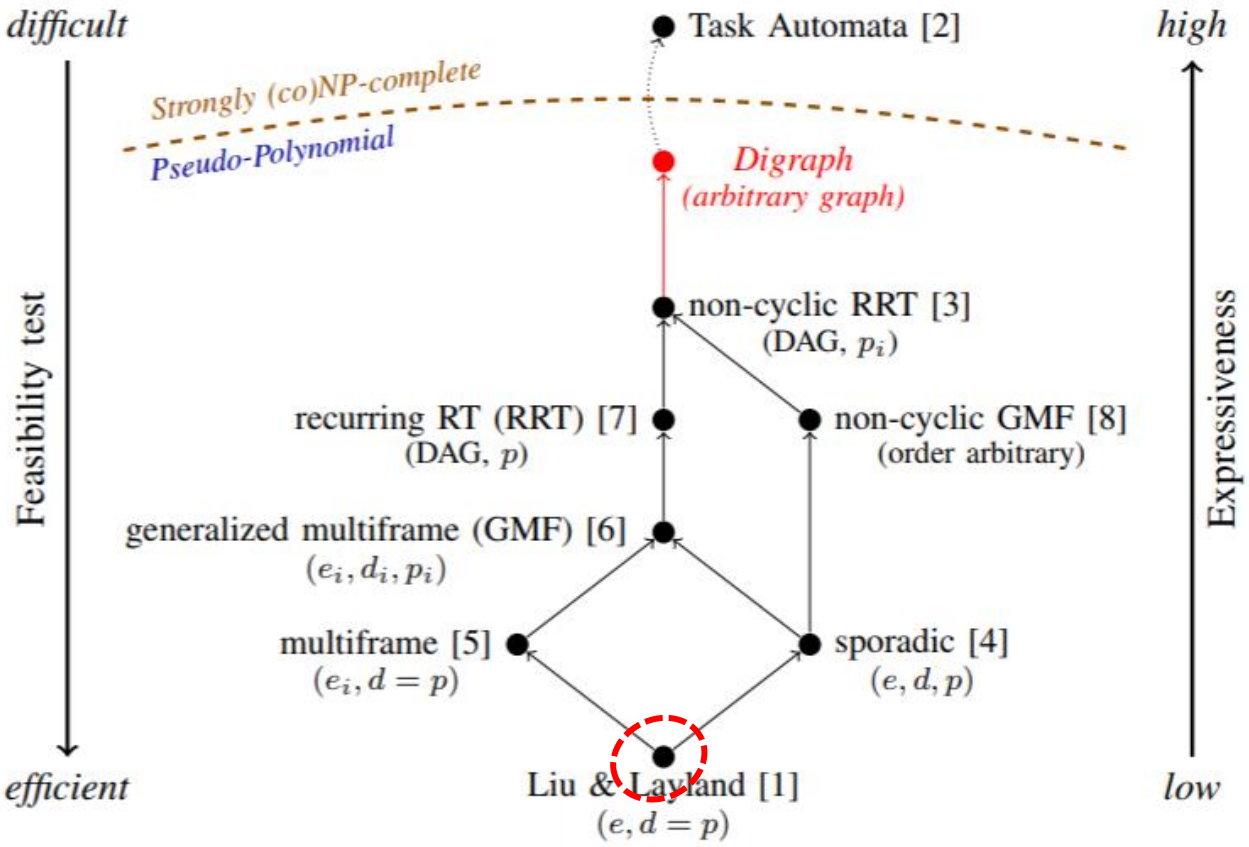
- Problem
  - What are Timed/Task Automata?
  - Why compute their DBF/RBF?

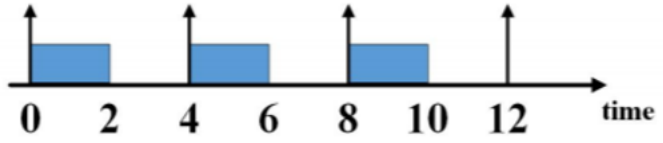
- A Starting Point

Real-Time Task Models

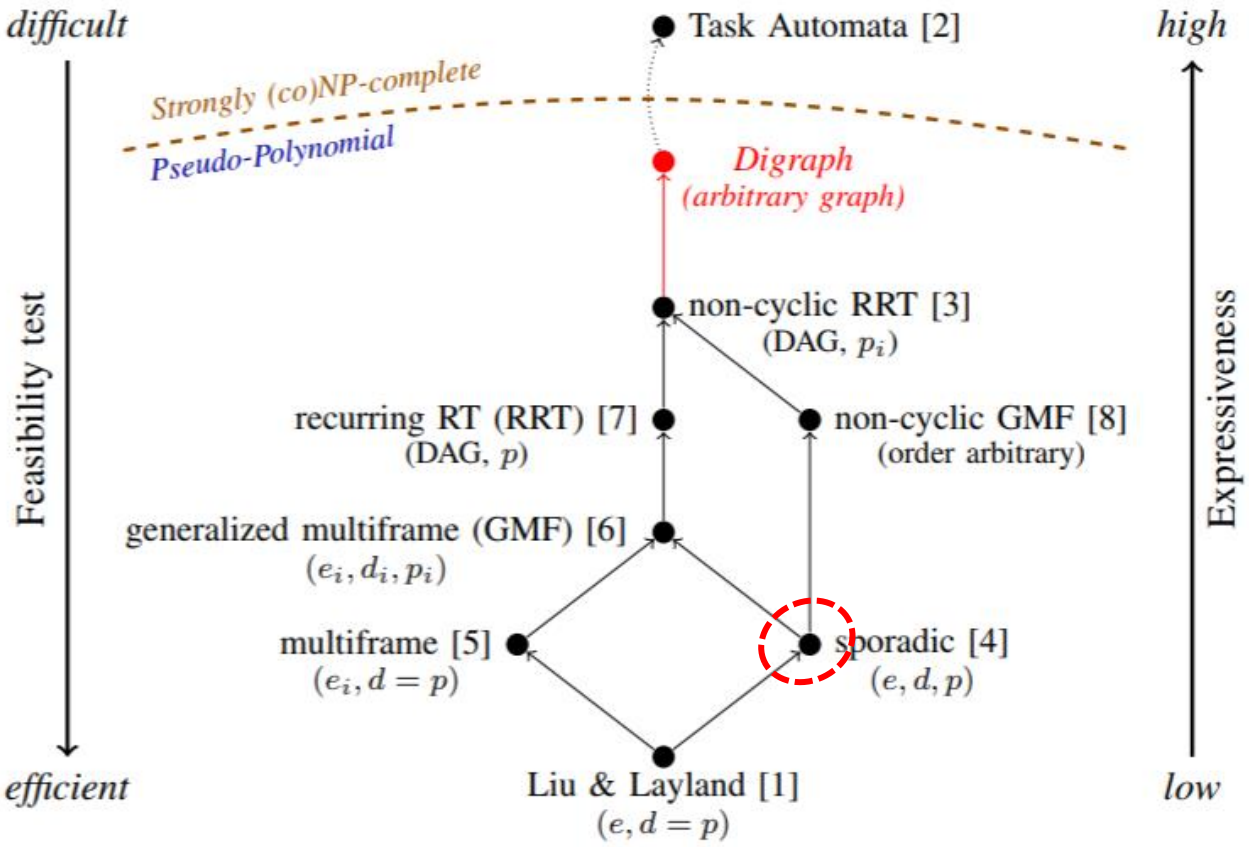


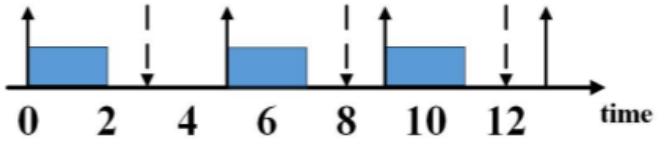
Real-Time Task Models



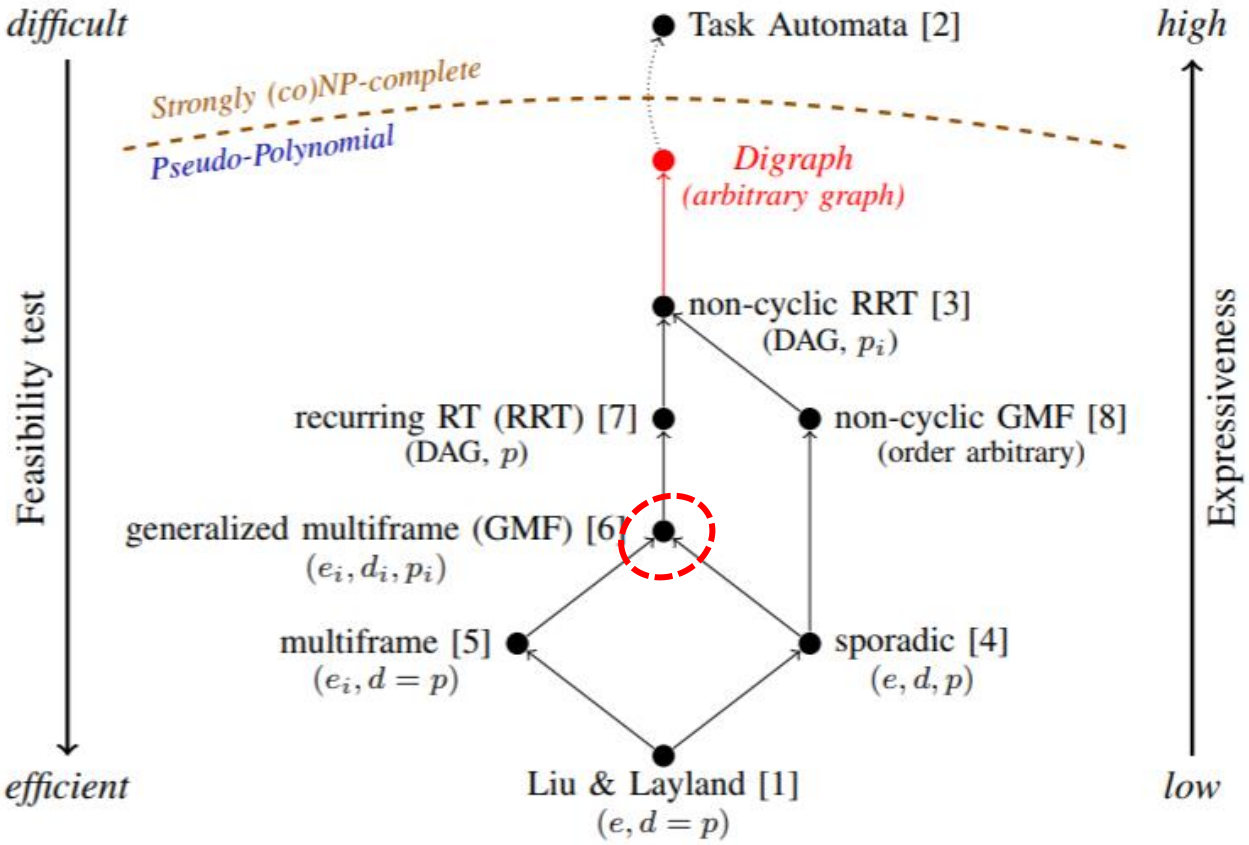


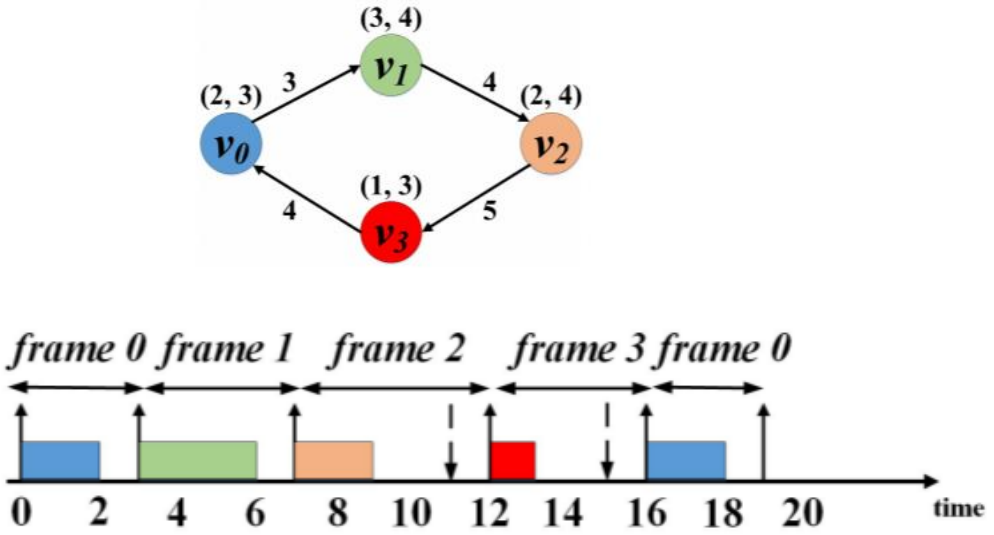
Real-Time Task Models



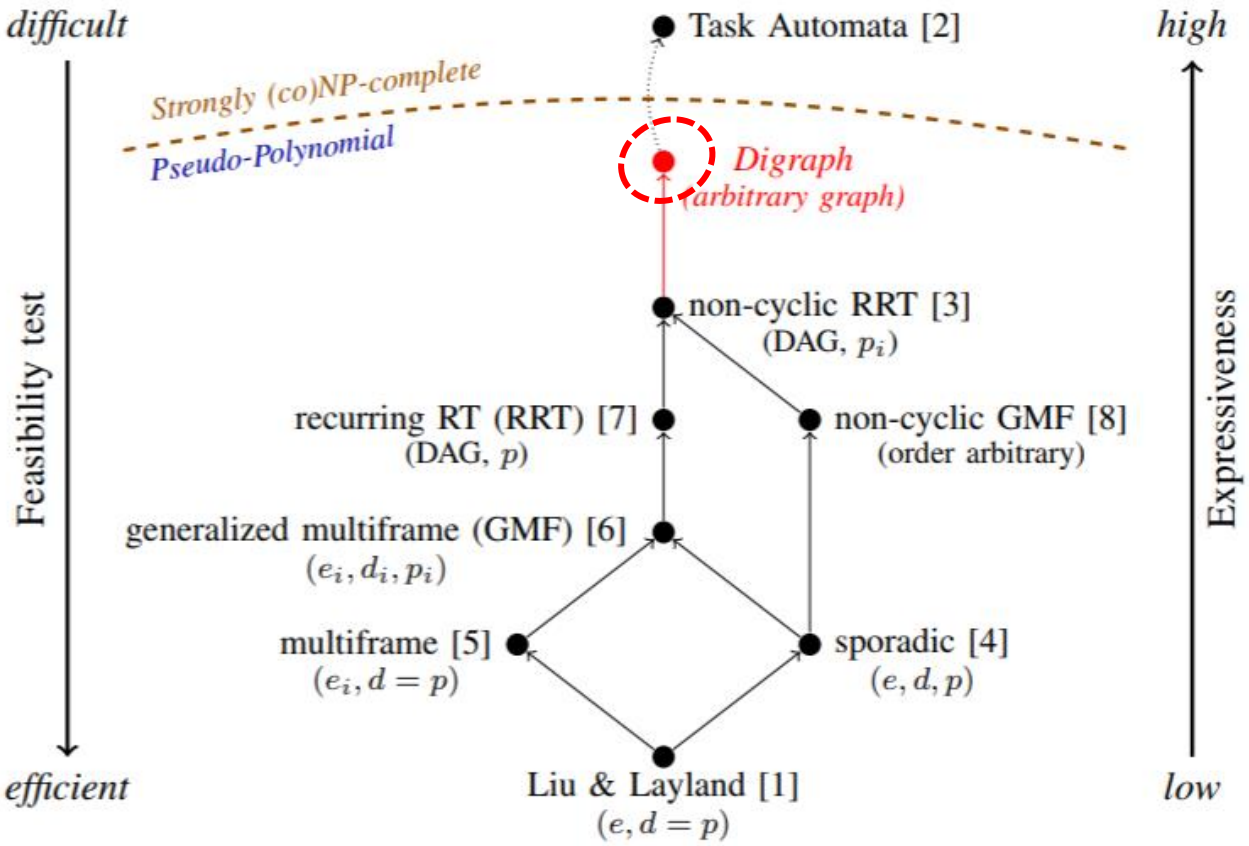


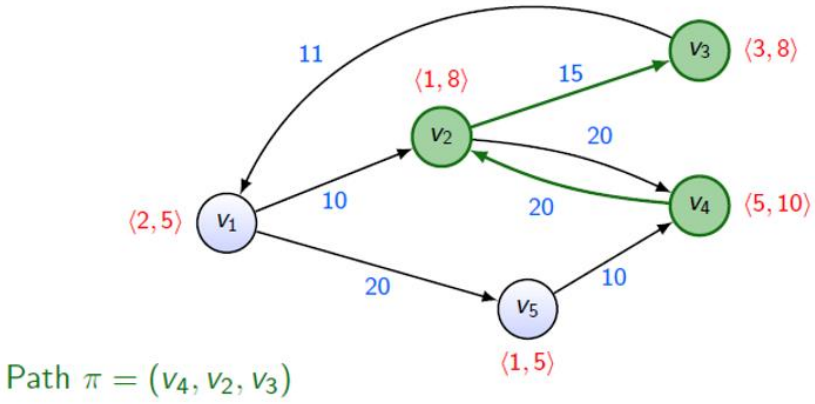
Real-Time Task Models

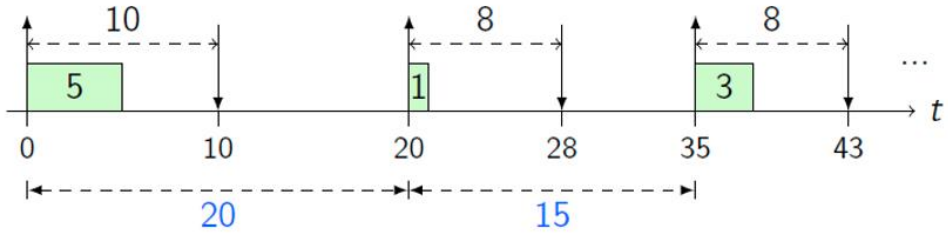




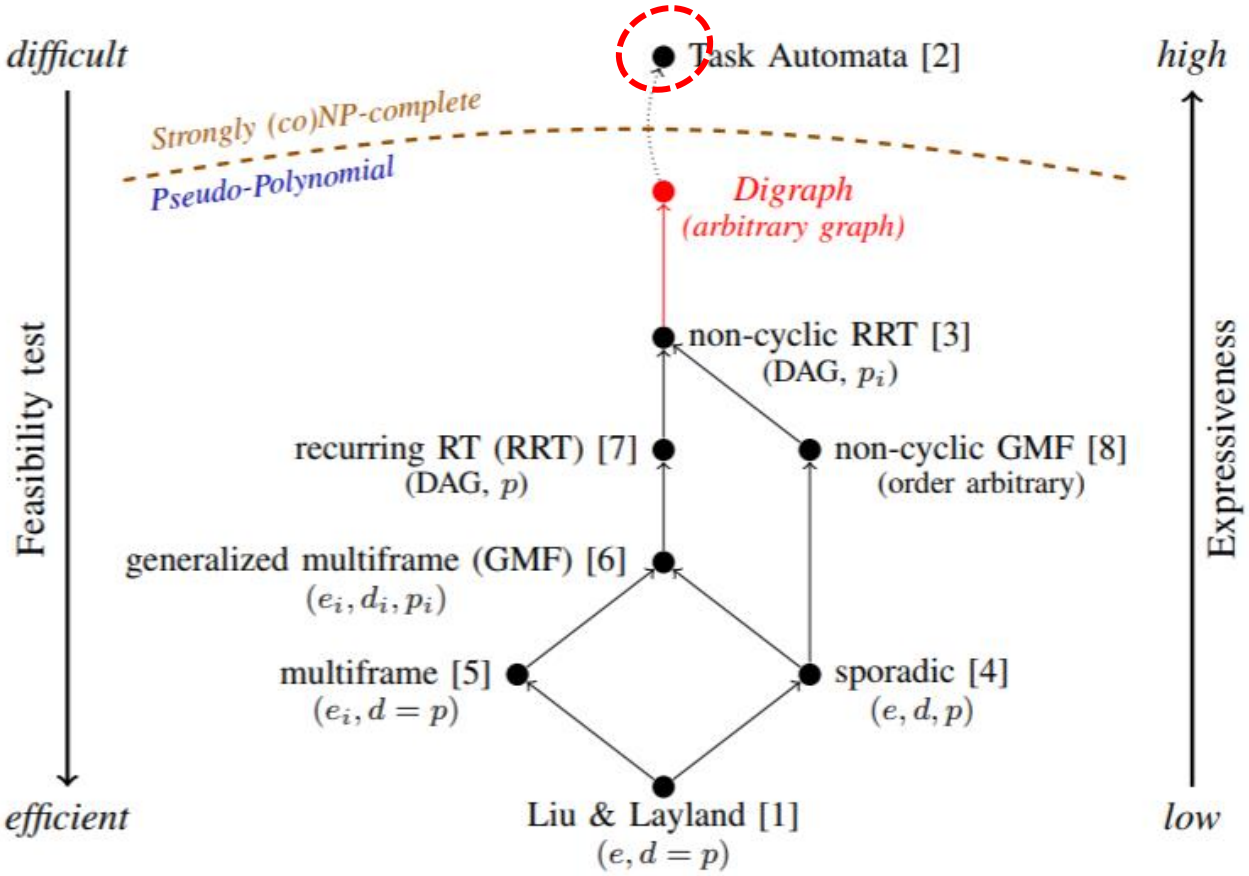
Real-Time Task Models

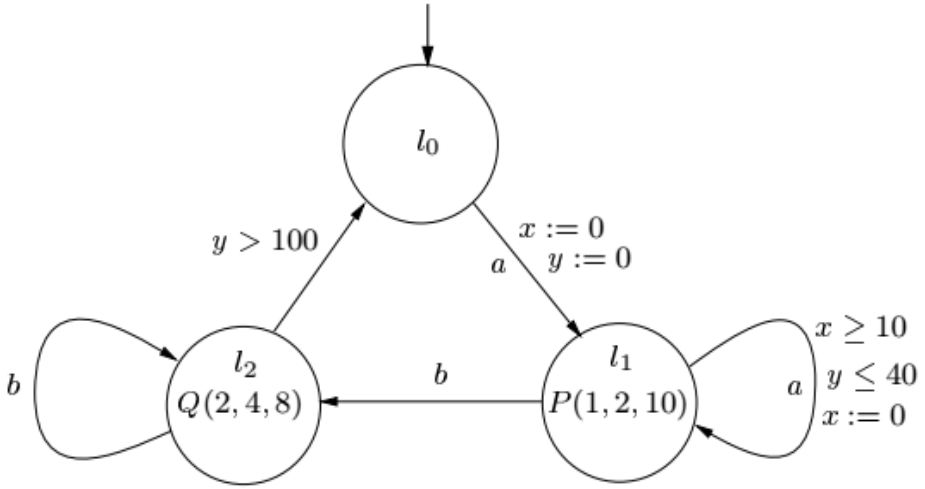






Real-Time Task Models

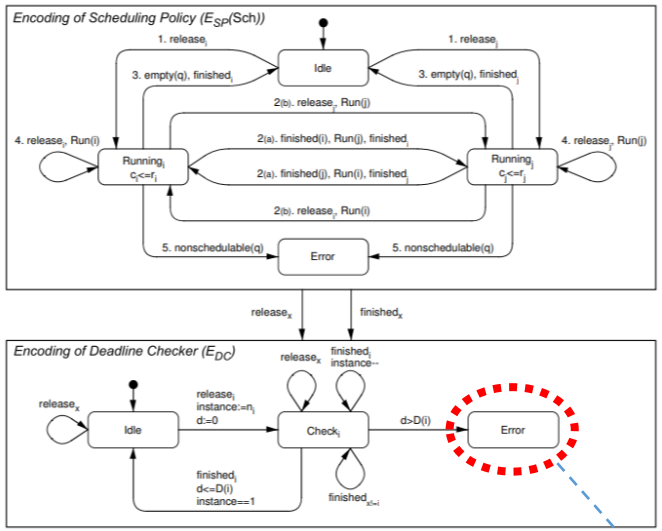




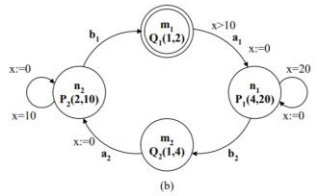
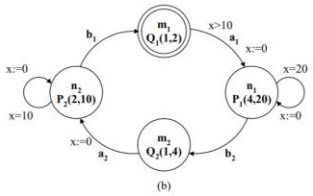
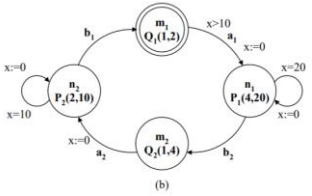
Timed Automata = FSM + clocks
Task Automata = Timed Automata + Task Release

Real-Time Task Models

Scheduability as Reachability of TA -- Model Checking



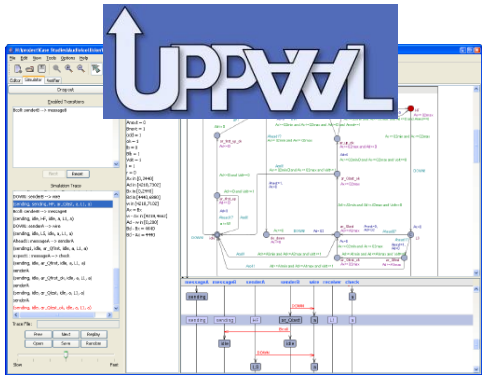
TA for scheduler and deadline checking

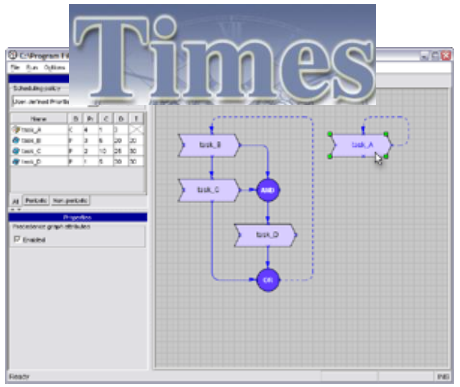
X  X  X … … X 

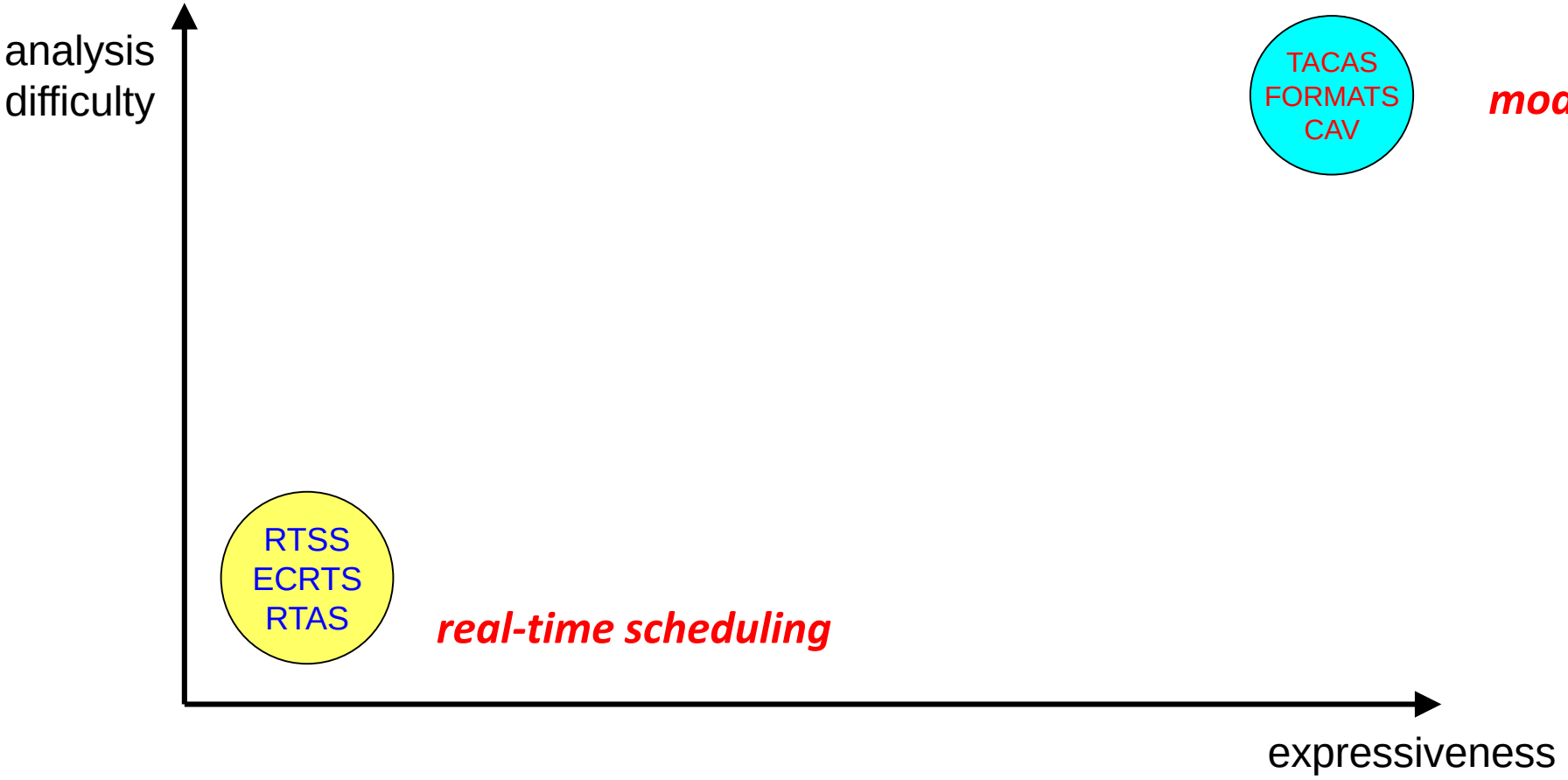
TA for task 1 TA for task 2 TA for task 2

Search the composed state transition graph, to check whether the state representing deadline miss is reachable.

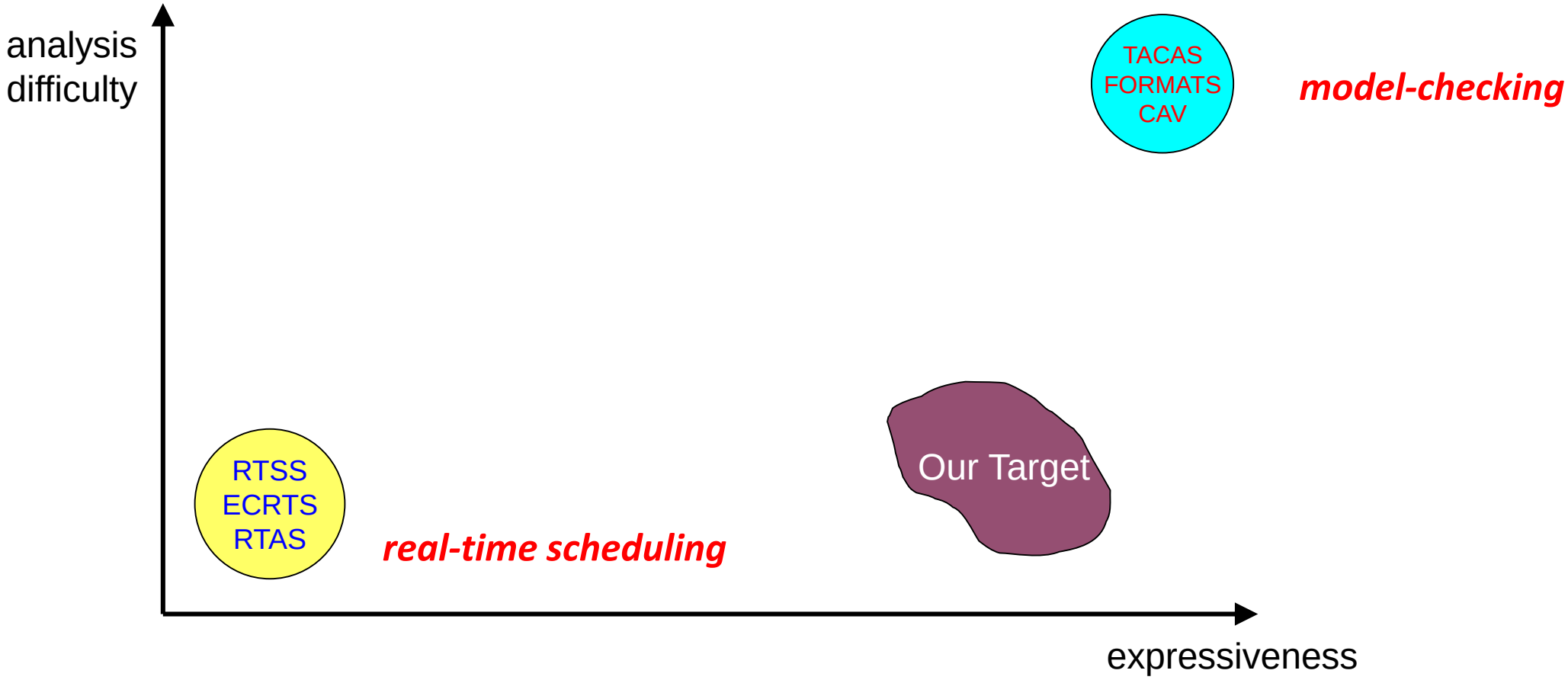
Real-Time Task Models



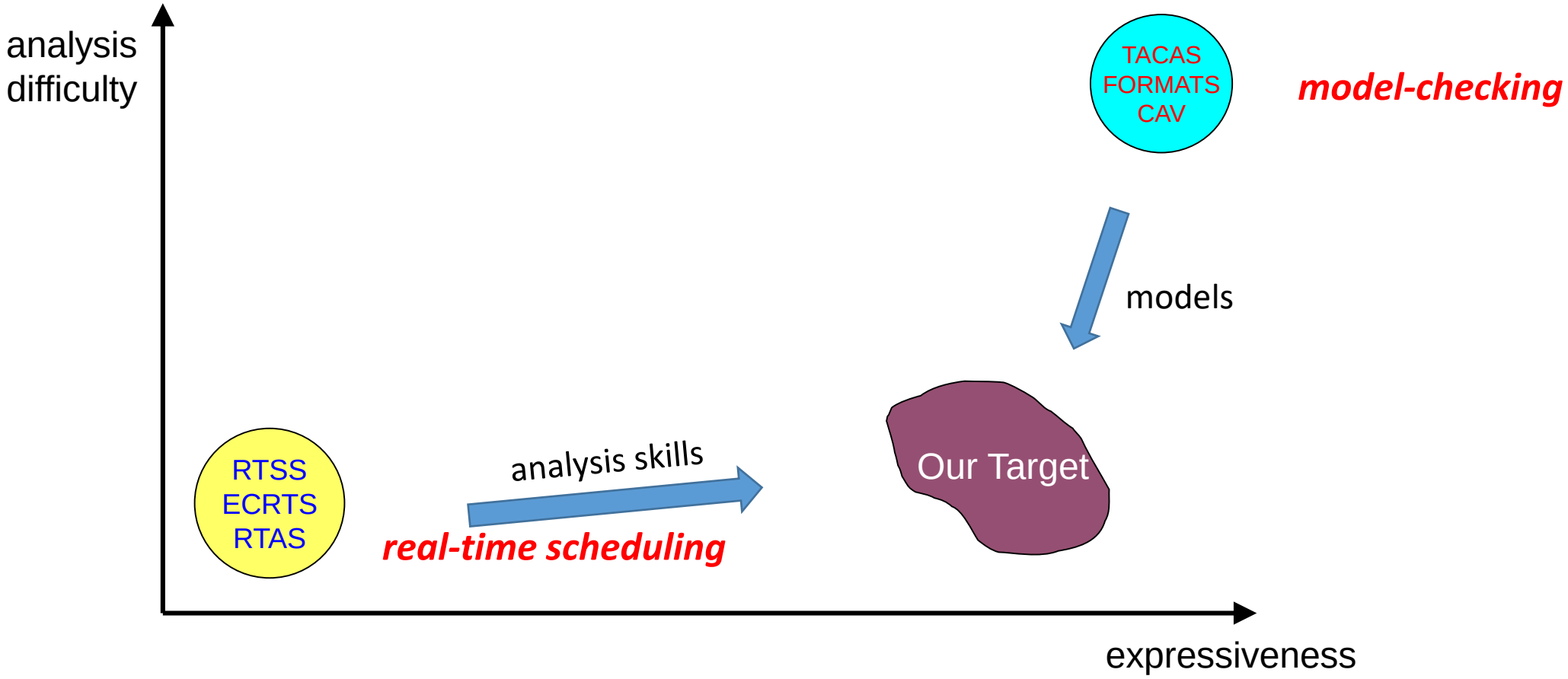




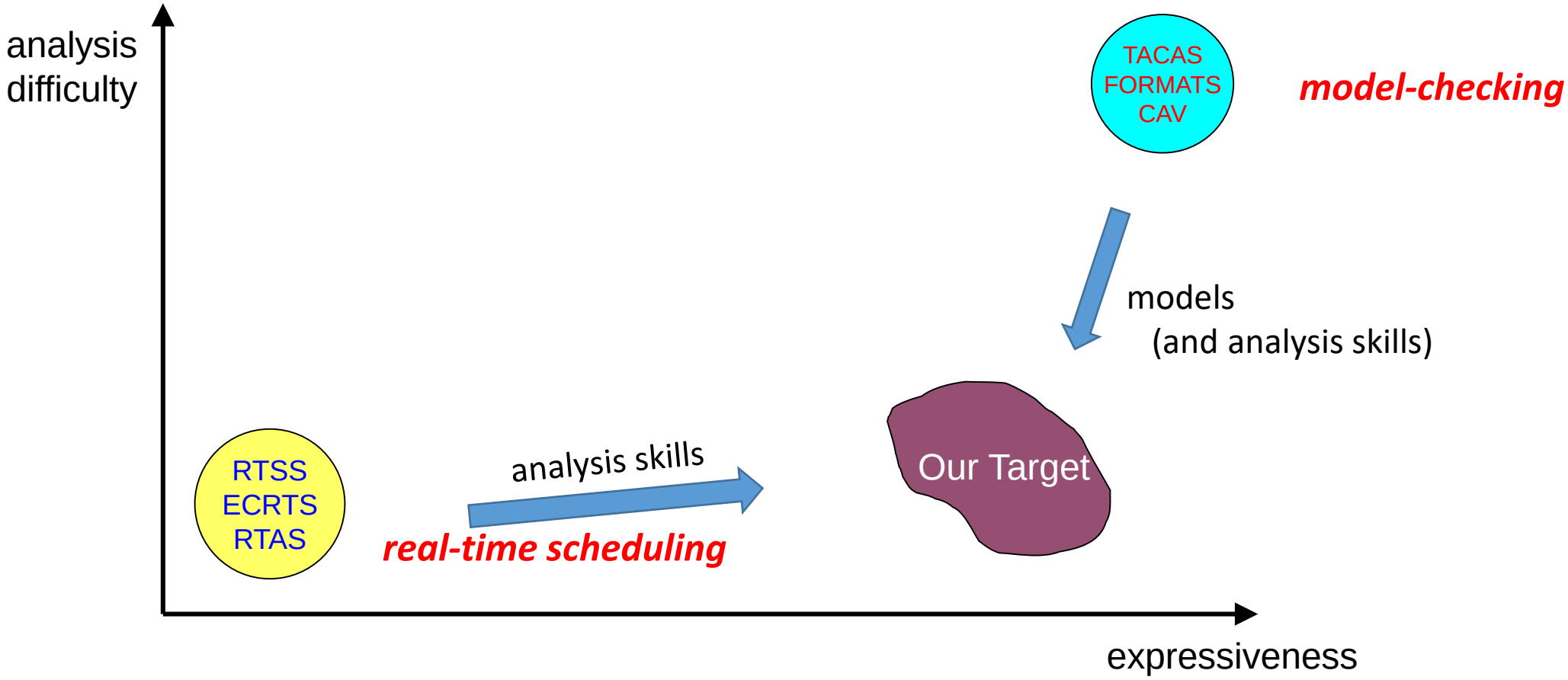
Real-Time Task Models



Real-Time Task Models

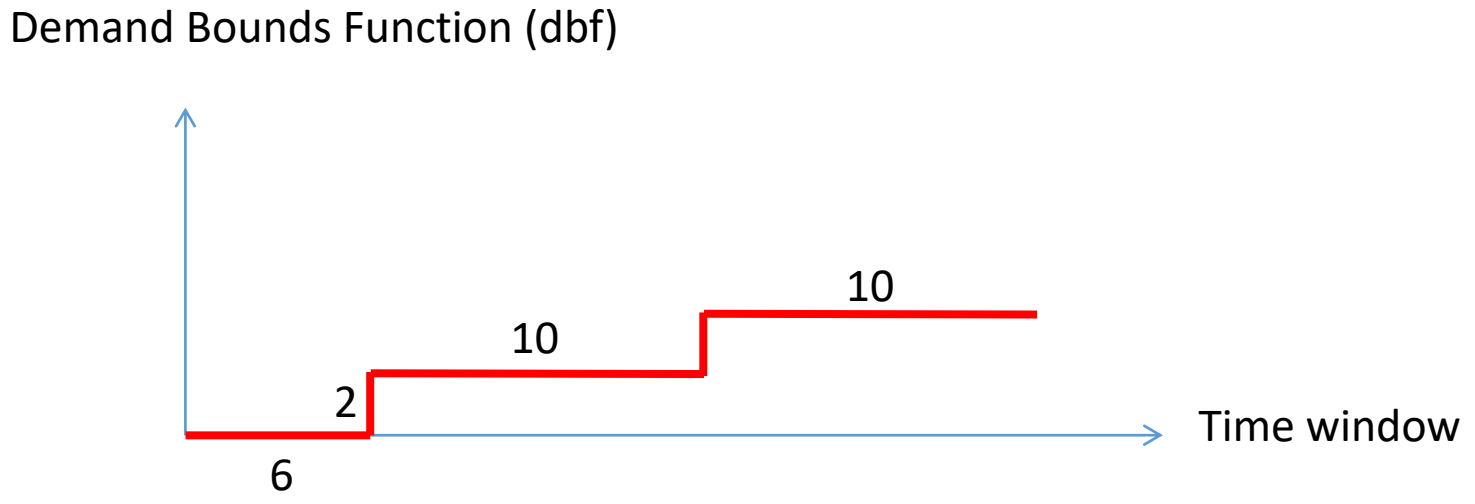


Real-Time Task Models



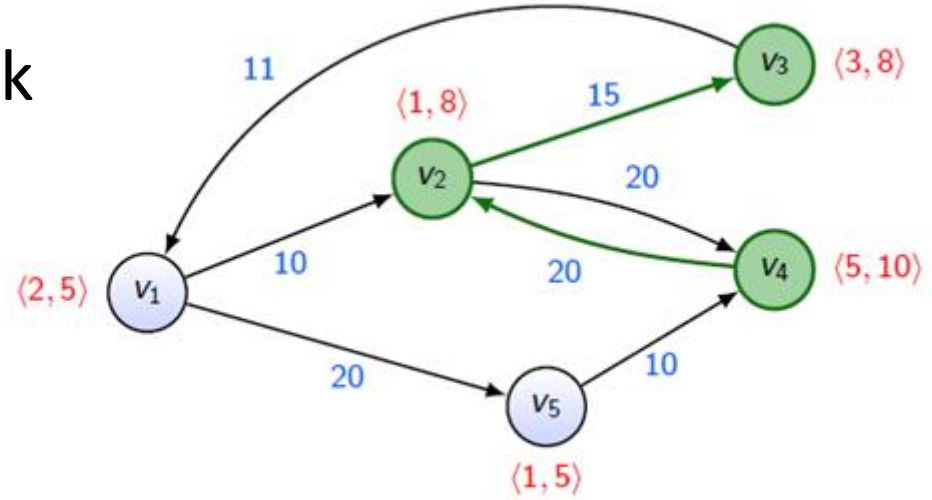
Demand Bound Function (DBF)

- DBF of a ***sporadic*** task
  - Period = 10
  - WCET = 2
  - Deadline = 6

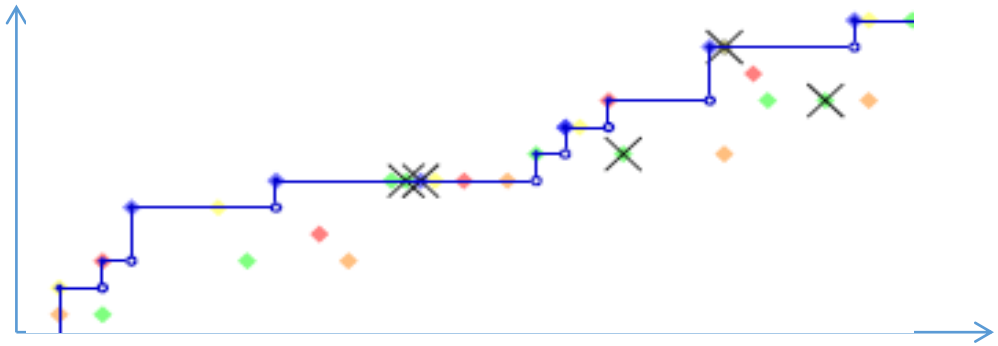


Demand Bound Function (DBF)

- DBF of a ***Digraph*** task

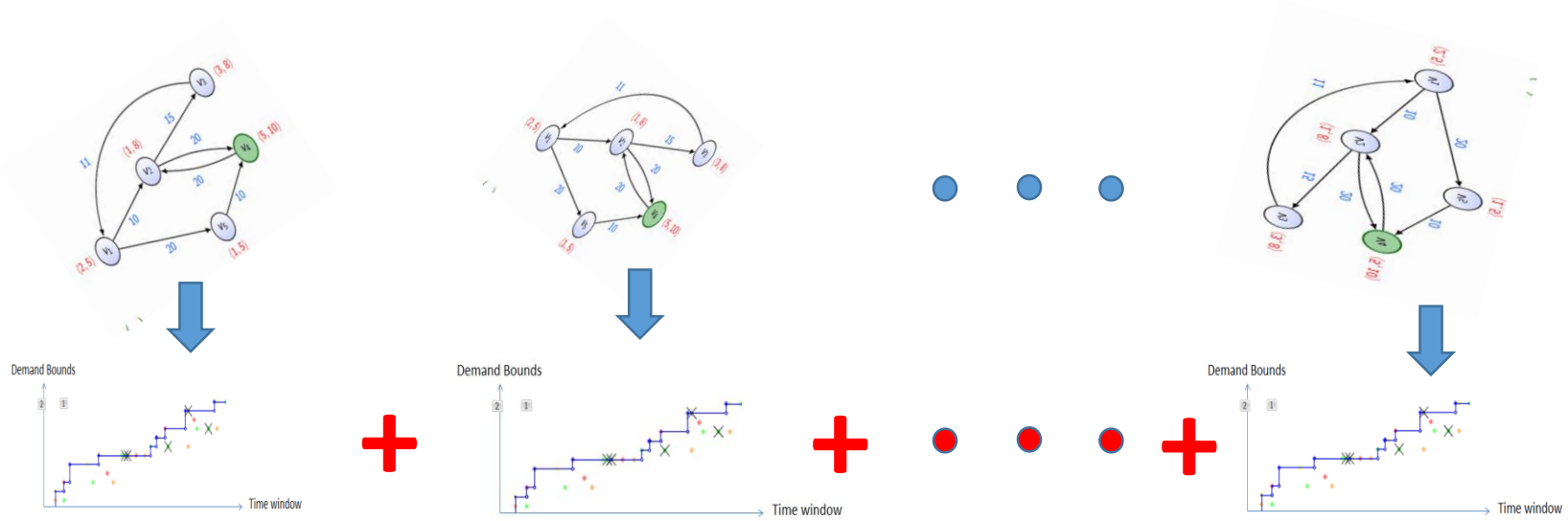


Demand Bounds Function (dbf)

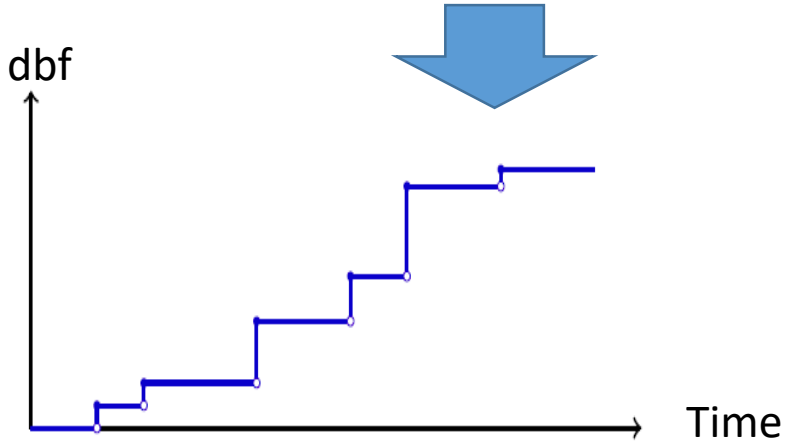


Time window

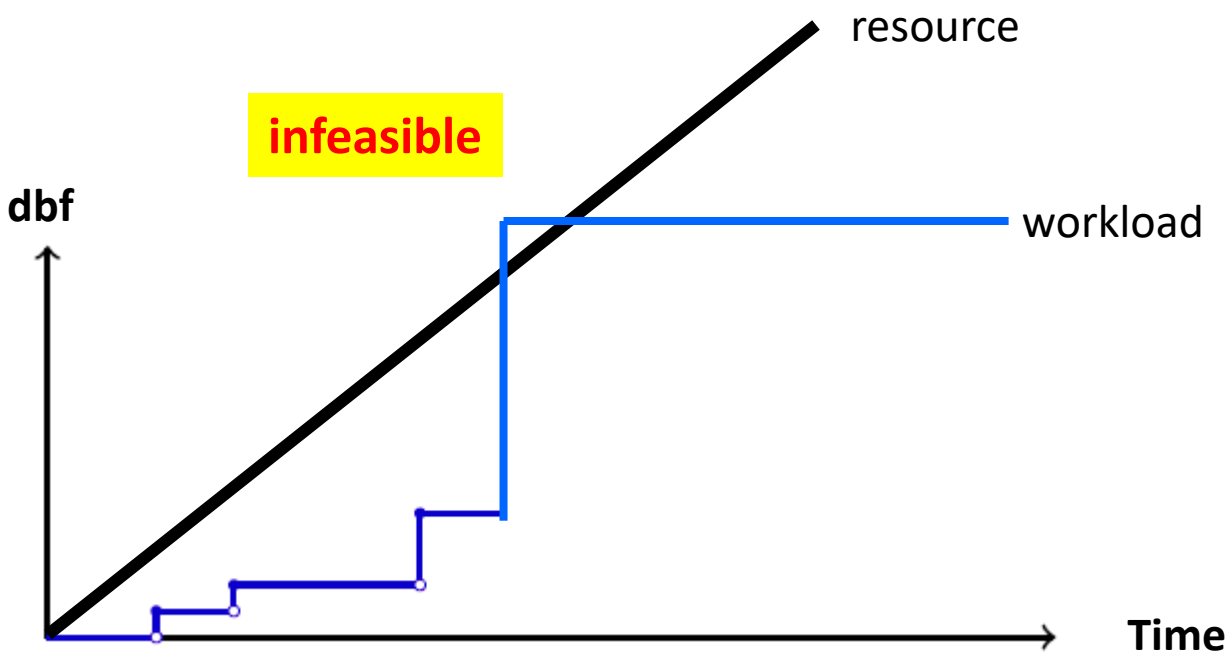
Demand Bound Function (DBF)



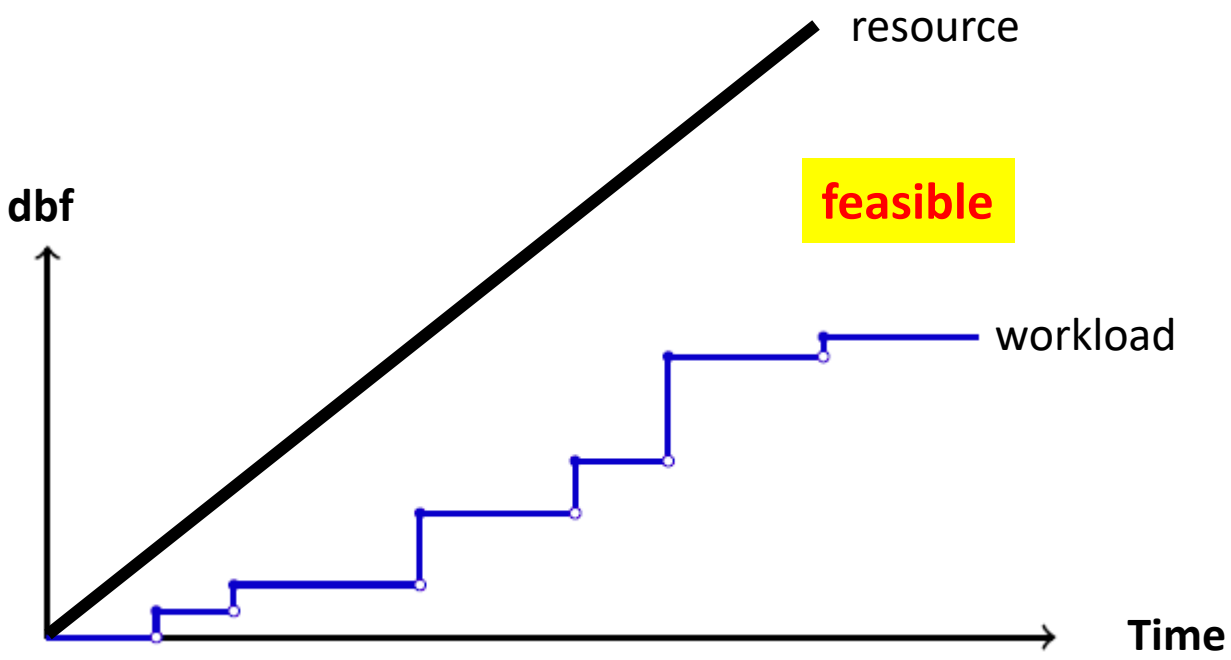
The system workload:



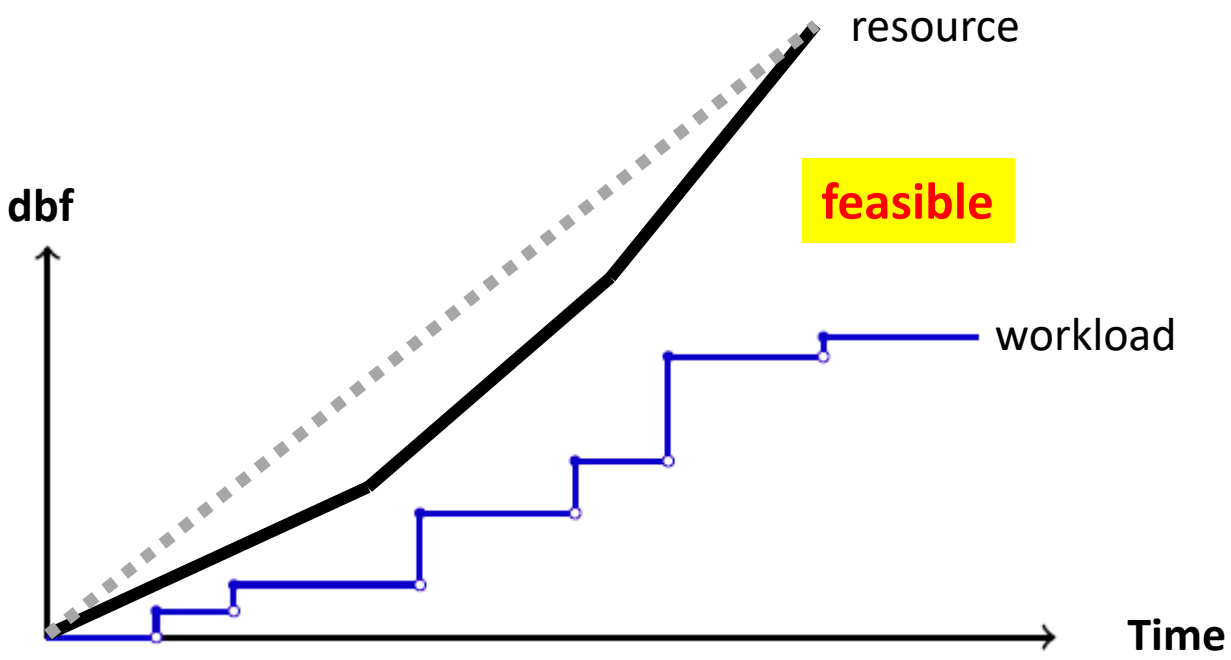
Demand Bound Function (DBF)



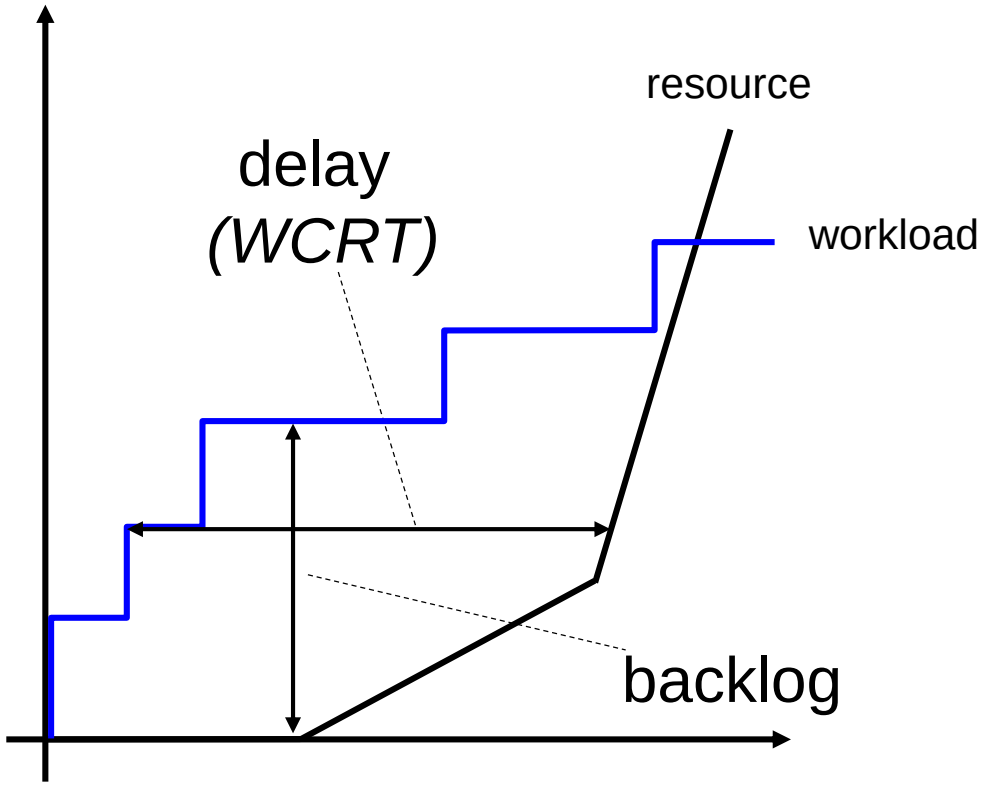
Demand Bound Function (DBF)



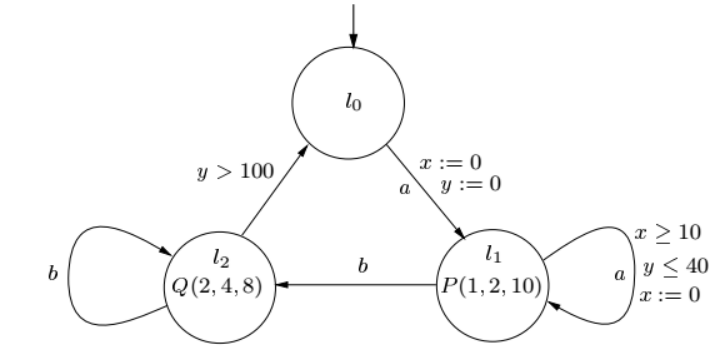
Demand Bound Function (DBF)

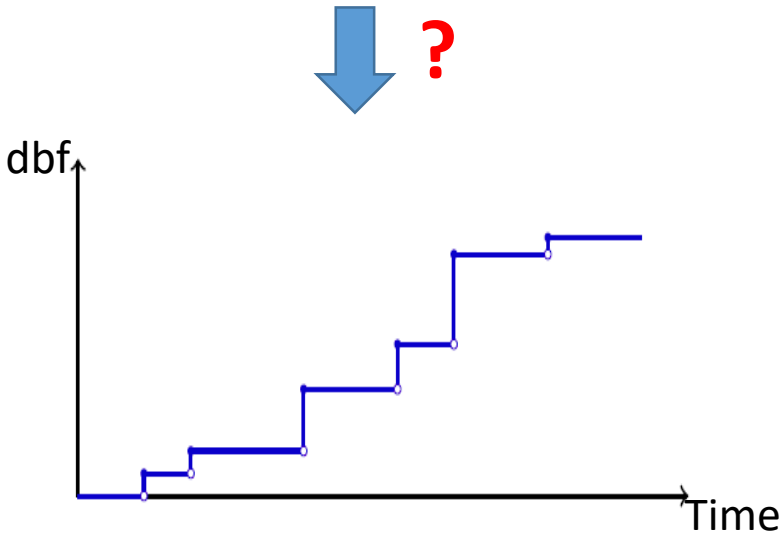


Request Bound Function (RBF) / Arrival Curve



Problem: Compute DBF/RBF of a TA (Efficiently)





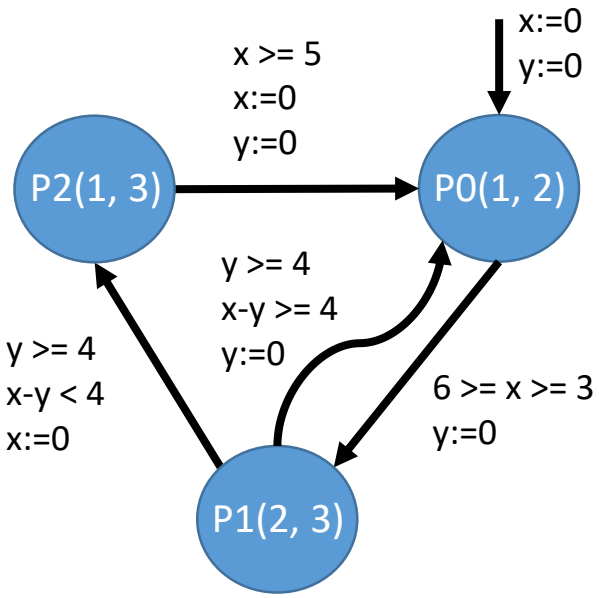
- If solved
  - The first step towards scalable schedulability analysis of general Task Automata models
  - Can efficiently check schedulability of a system composed of **independent** tasks each modeled as a Task Automaton

Outline

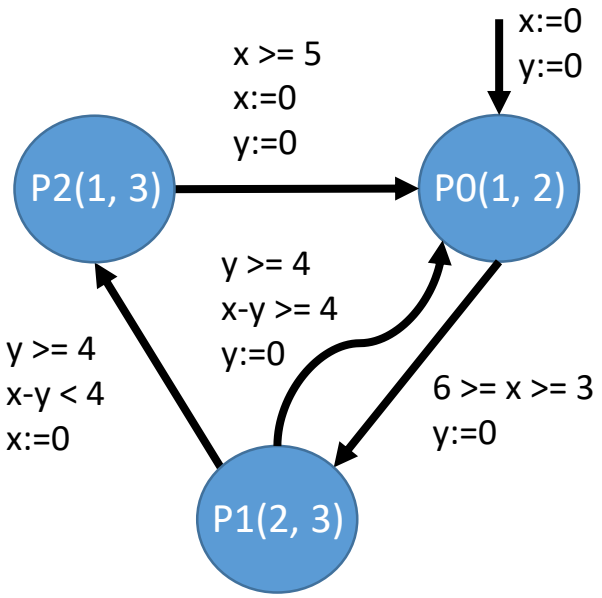
- Problem
  - What are Timed/Task Automata?
  - Why compute their DBF/RBF?

- A Starting Point

Semantics of Task Automata

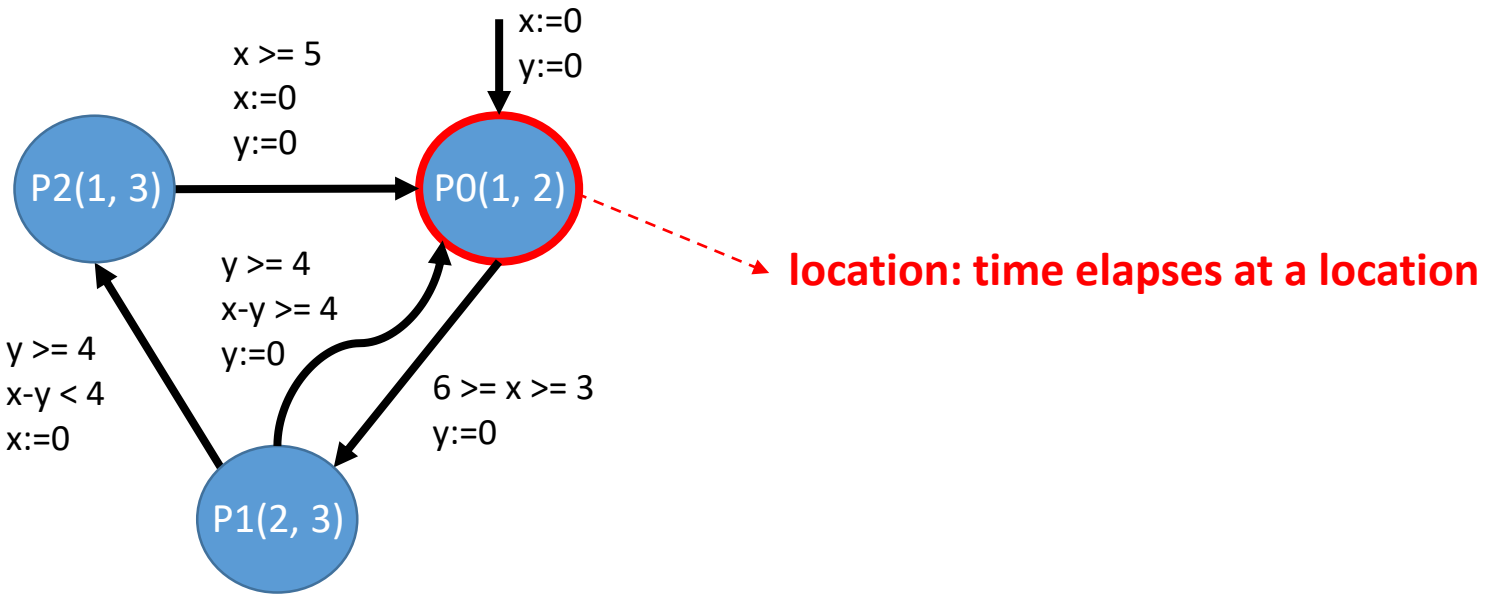


Semantics of Task Automata

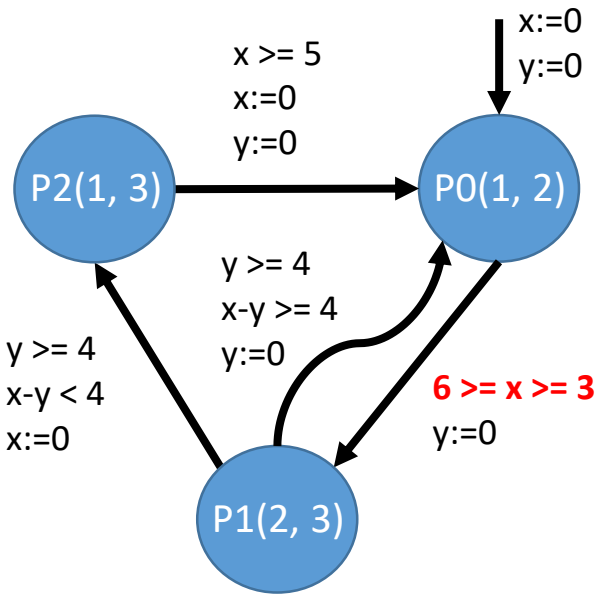


x, y: clocks

Semantics of Task Automata

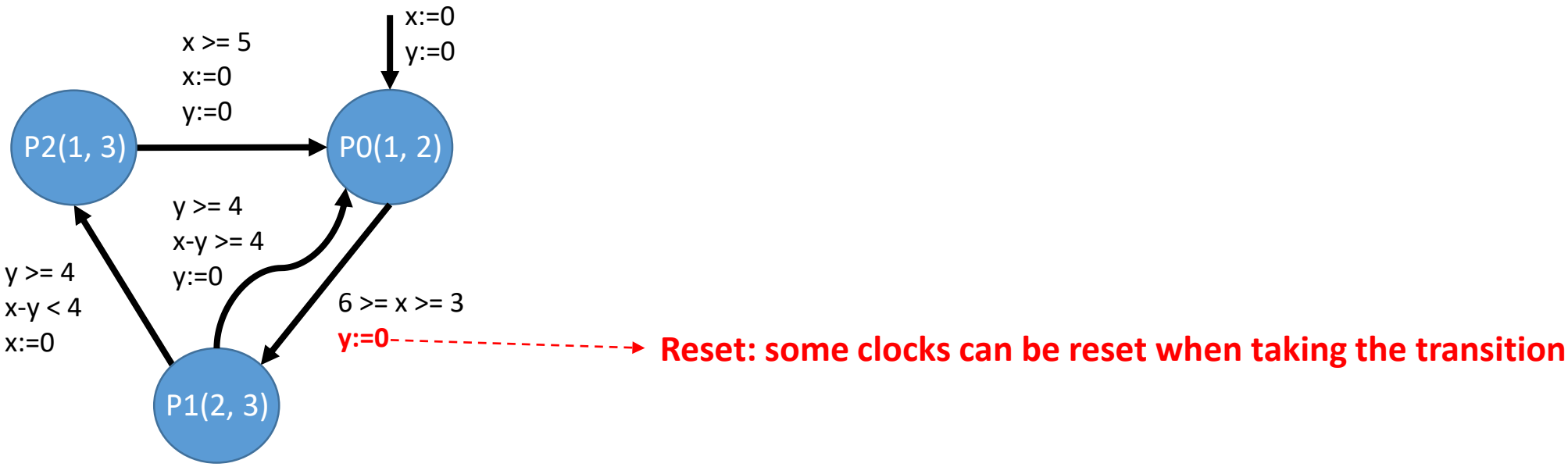


Semantics of Task Automata

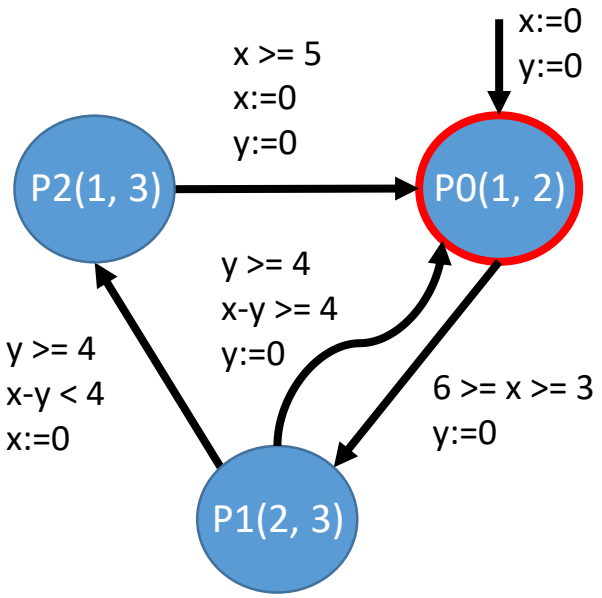


Guard: conditions on clock values to take the transition

Semantics of Task Automata



Semantics of Task Automata



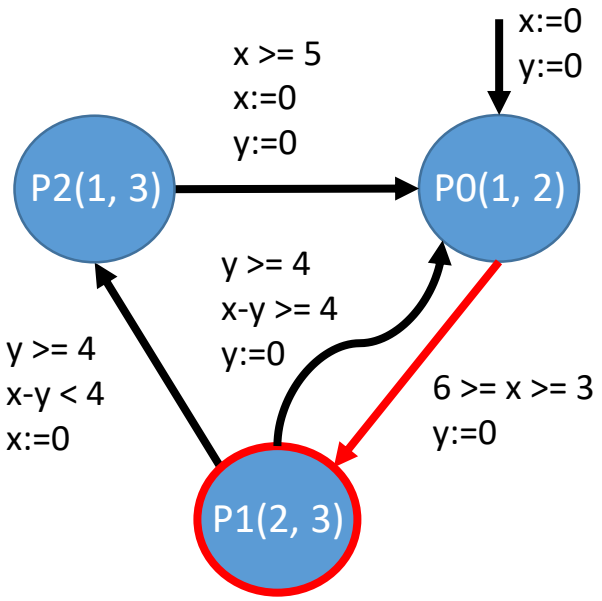
before

after

x=0
y=0

A possible run:

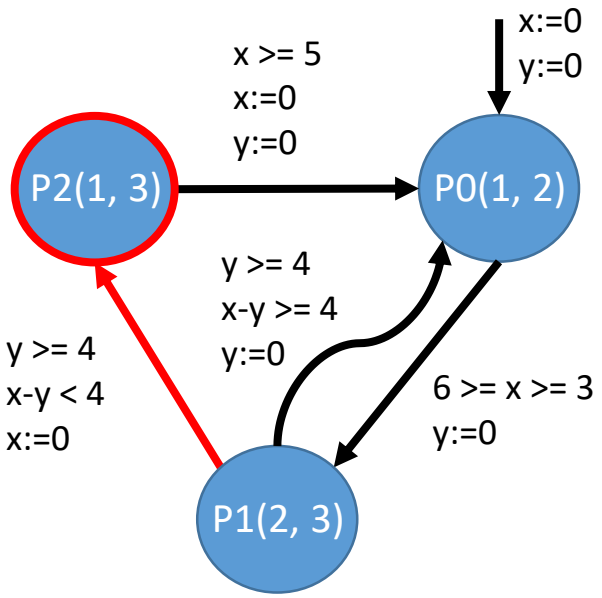
Semantics of Task Automata



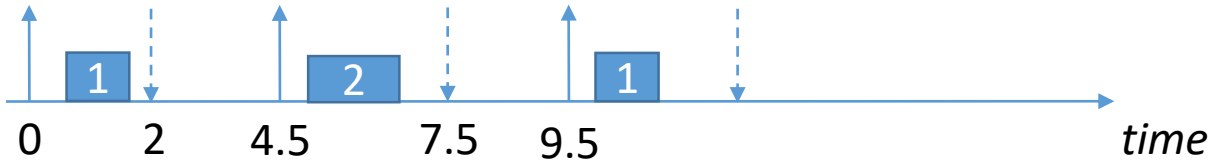
A possible run:

before		x=4.5 y=4.5
after	x=0 y=0	x=4.5 y=0

Semantics of Task Automata

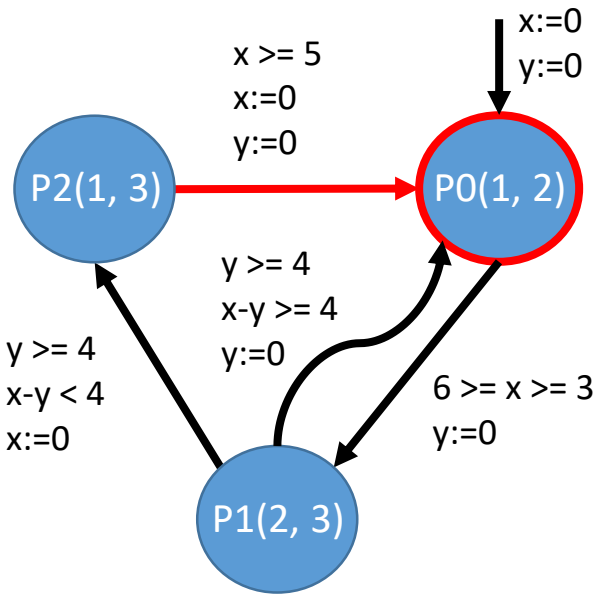


A possible run:

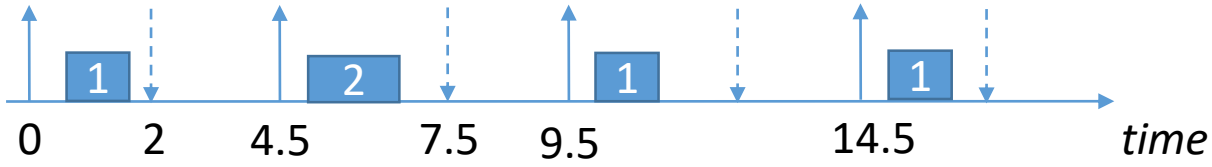


before		x=4.5 y=4.5	x=9.5 y=5
after	x=0 y=0	x=4.5 y=0	x=0 y=5

Semantics of Task Automata

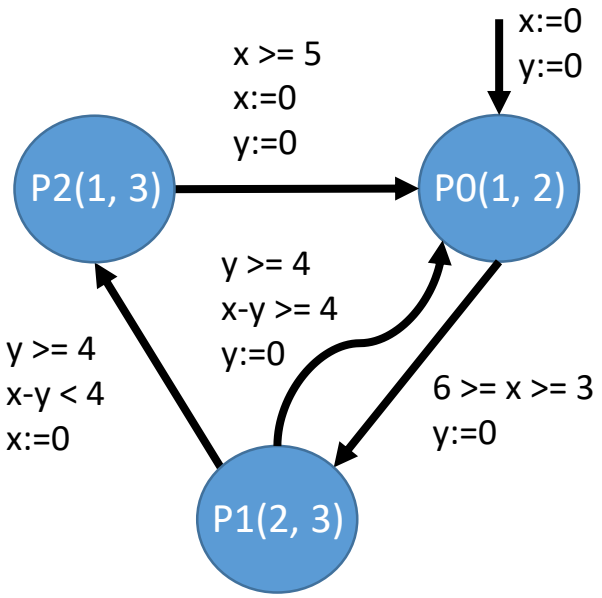


A possible run:

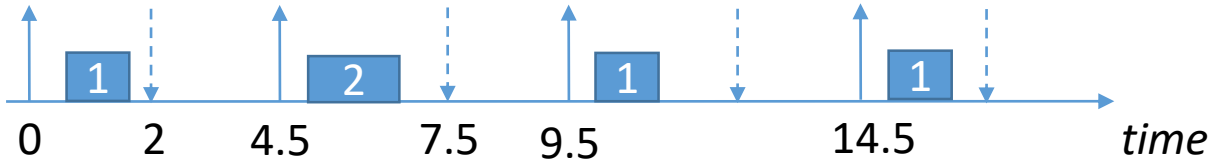


before		x=4.5 y=4.5	x=9.5 y=5	x=5 y=10
after	x=0 y=0	x=4.5 y=0	x=0 y=5	x=0 y=0

Semantics of Task Automata



A possible run:

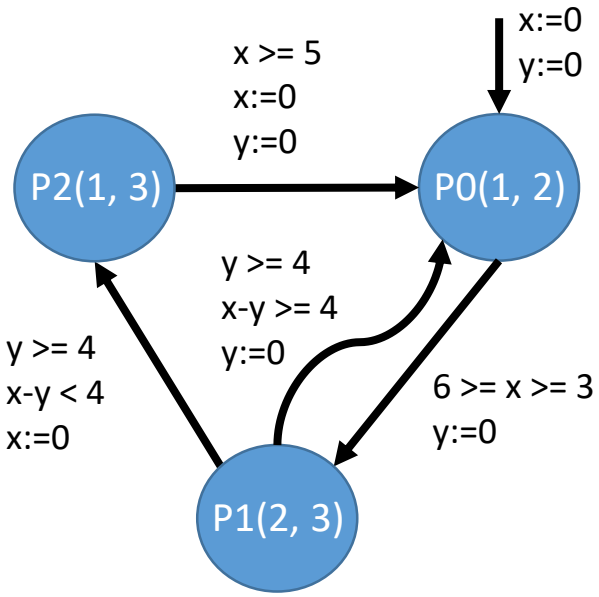


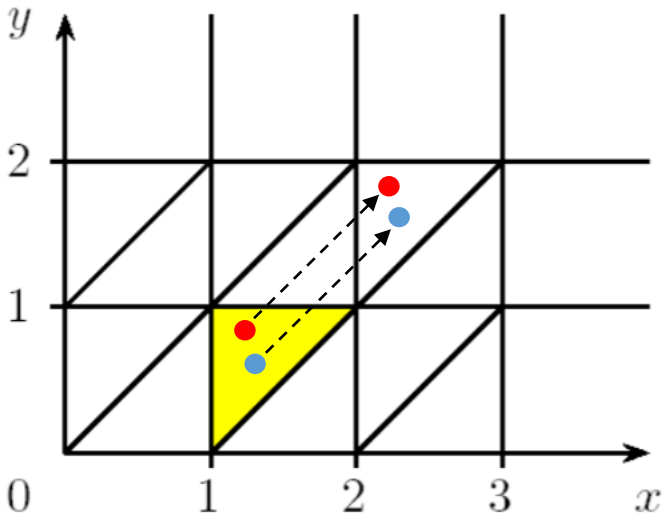
before		x=4.5 y=4.5	x=9.5 y=5	x=5 y=10
after	x=0 y=0	x=4.5 y=0	x=0 y=5	x=0 y=0

Problem: Infinitely many possible runs

Region

From Infinite to Finite State Space

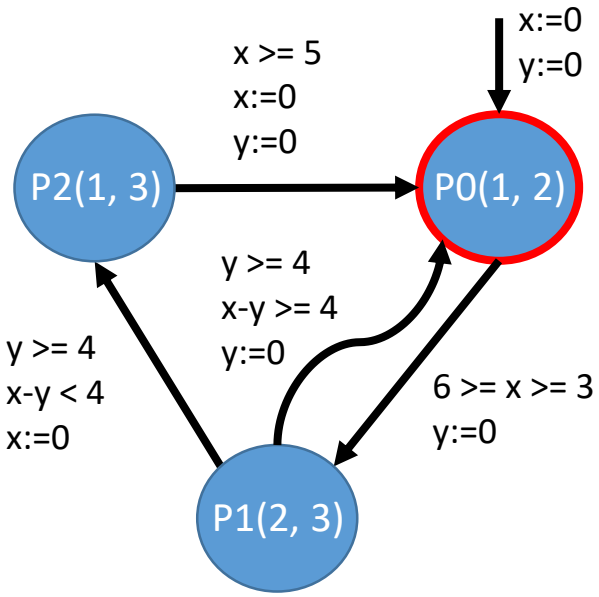


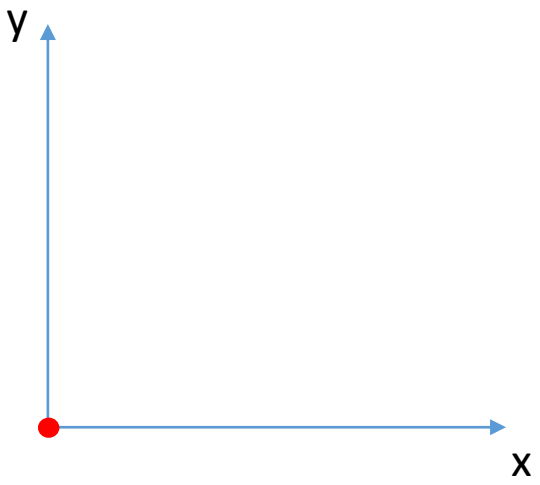


Region preserves reachability

Zone

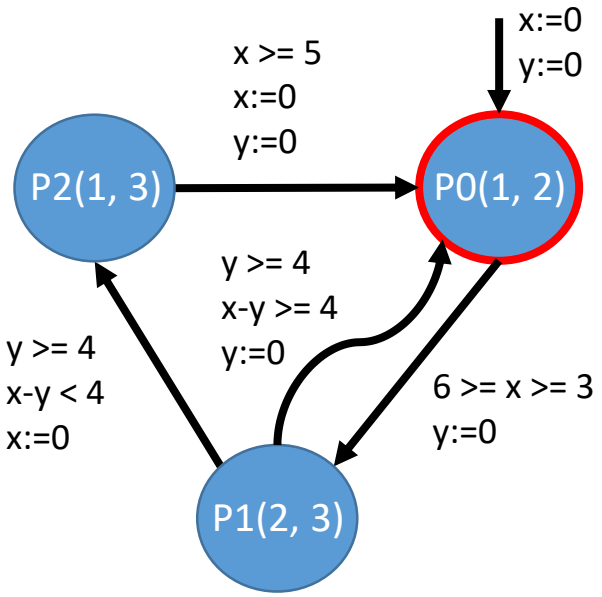
More compact state space representation

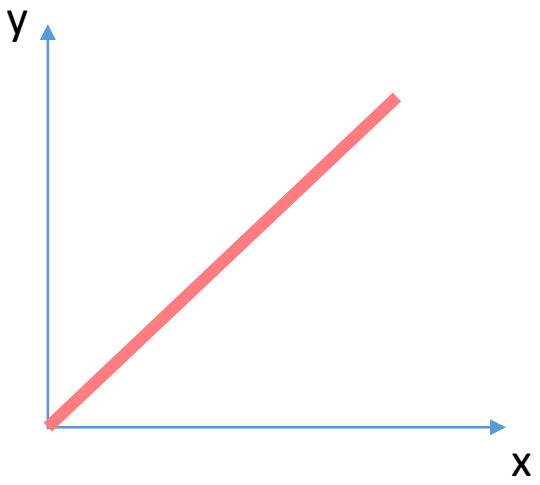




Zone

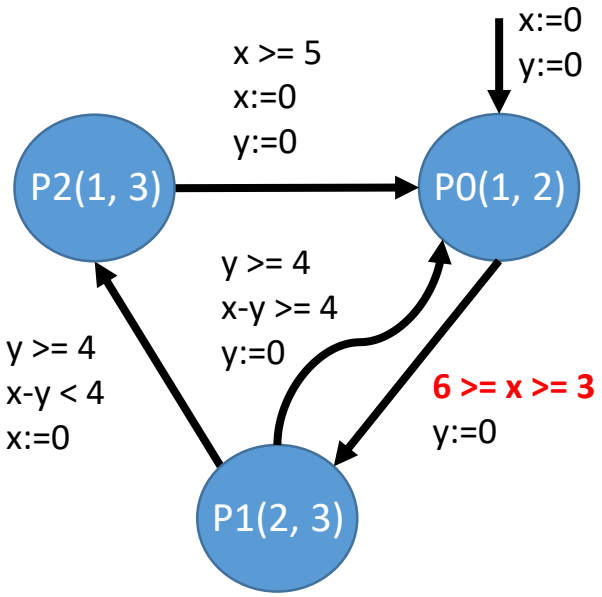
More compact state space representation

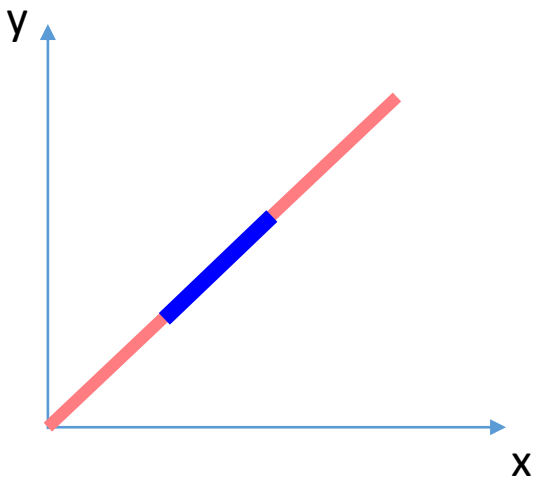




Zone

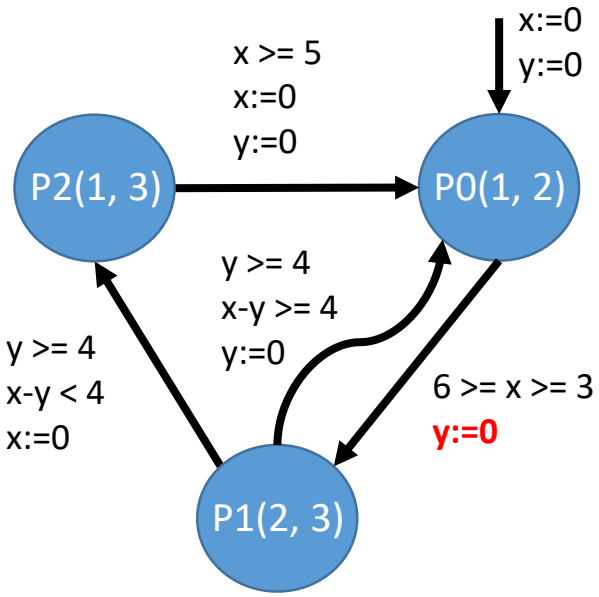
More compact state space representation

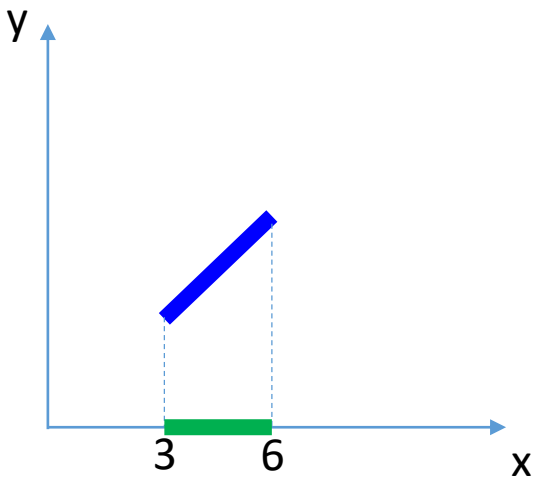




Zone

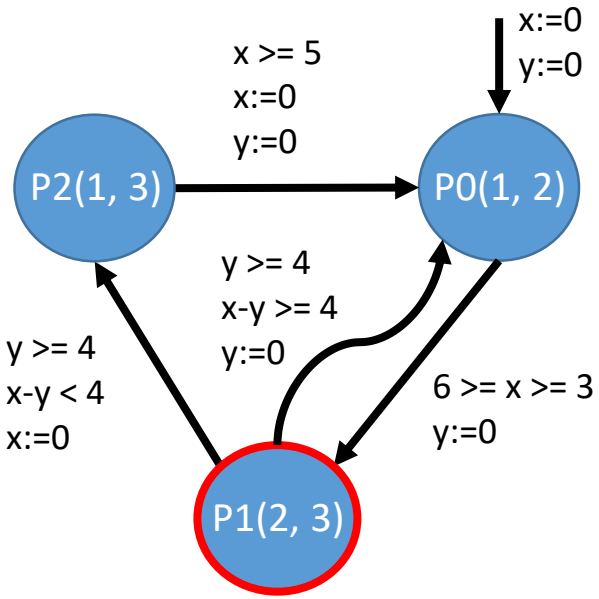
More compact state space representation

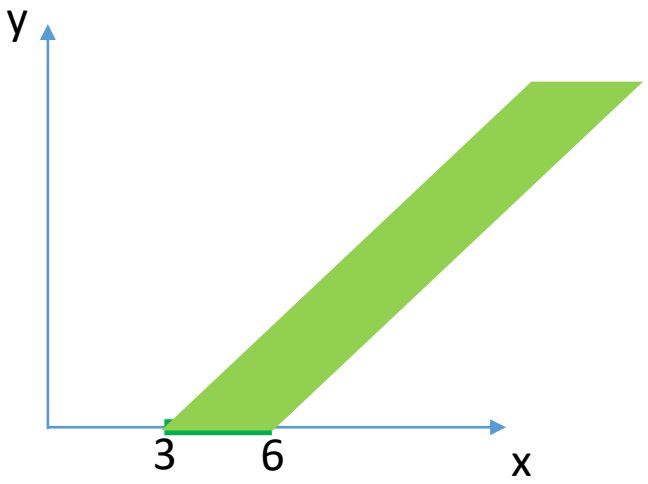




Zone

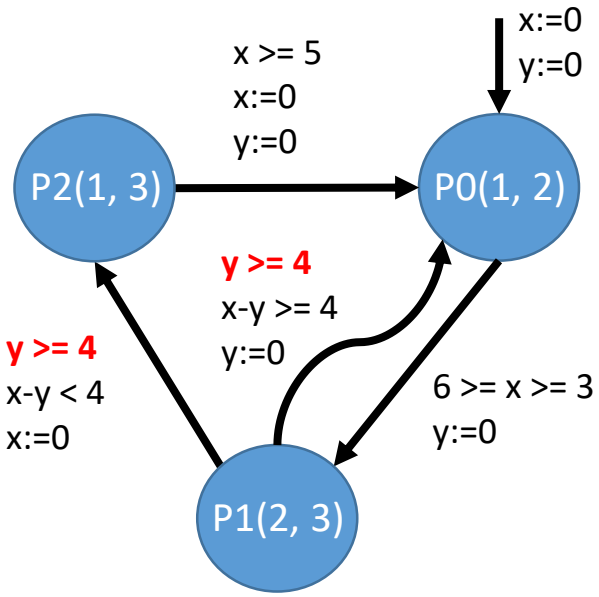
More compact state space representation

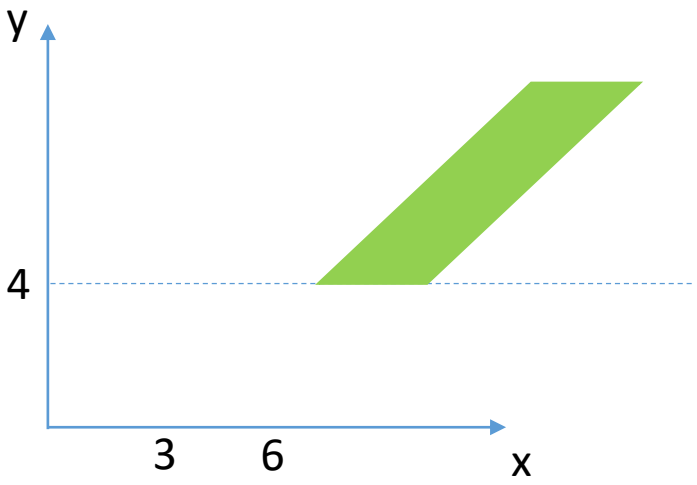




Zone

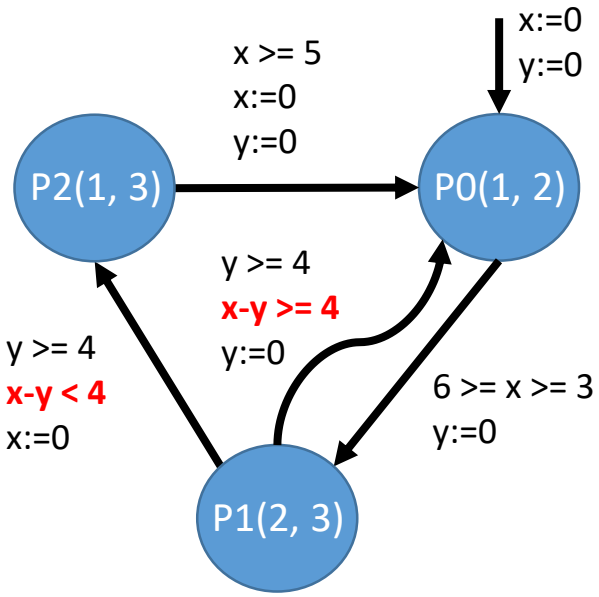
More compact state space representation

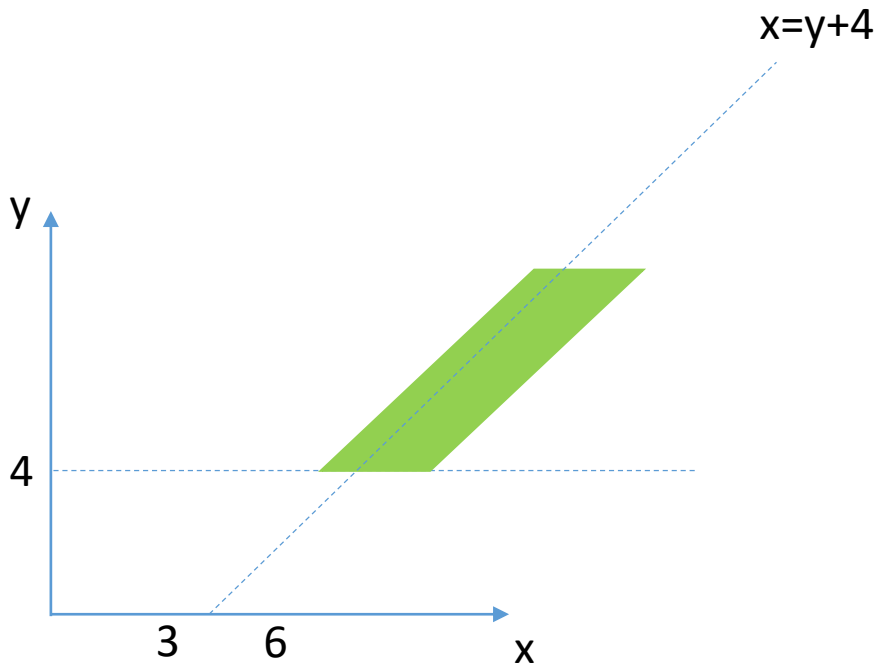




Zone

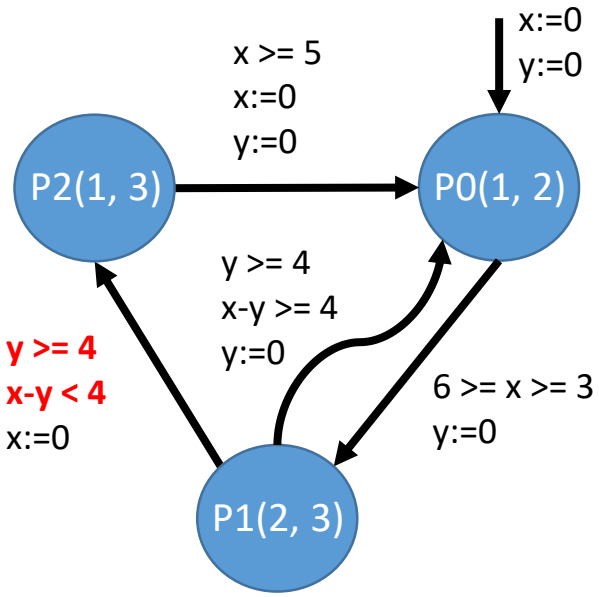
More compact state space representation

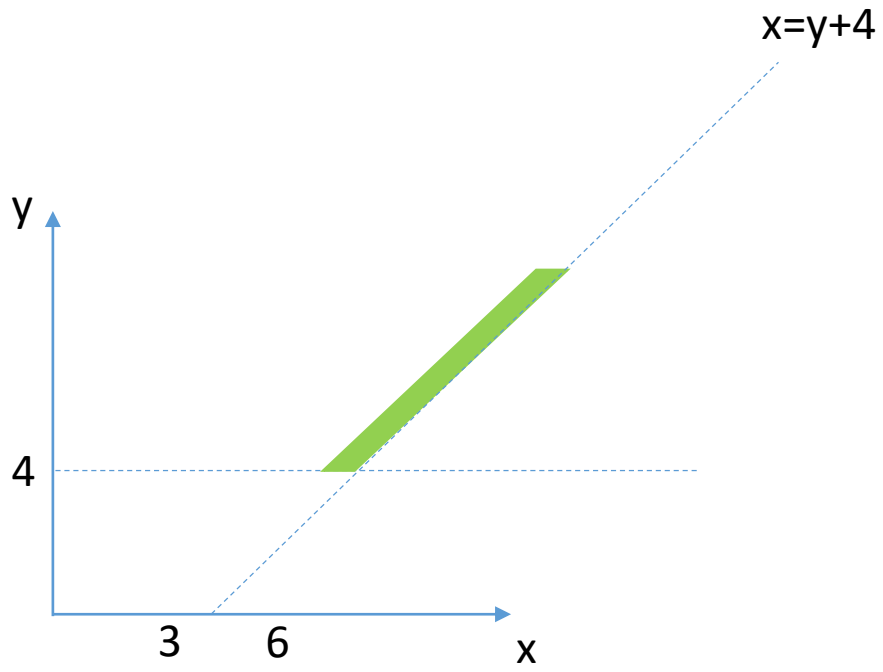




Zone

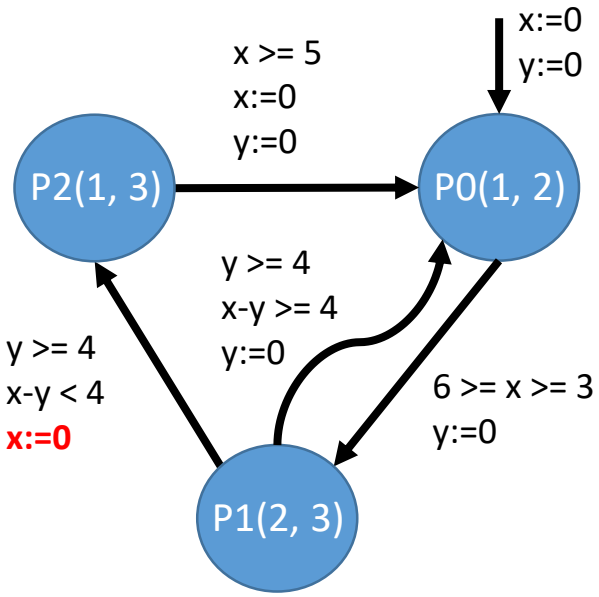
More compact state space representation

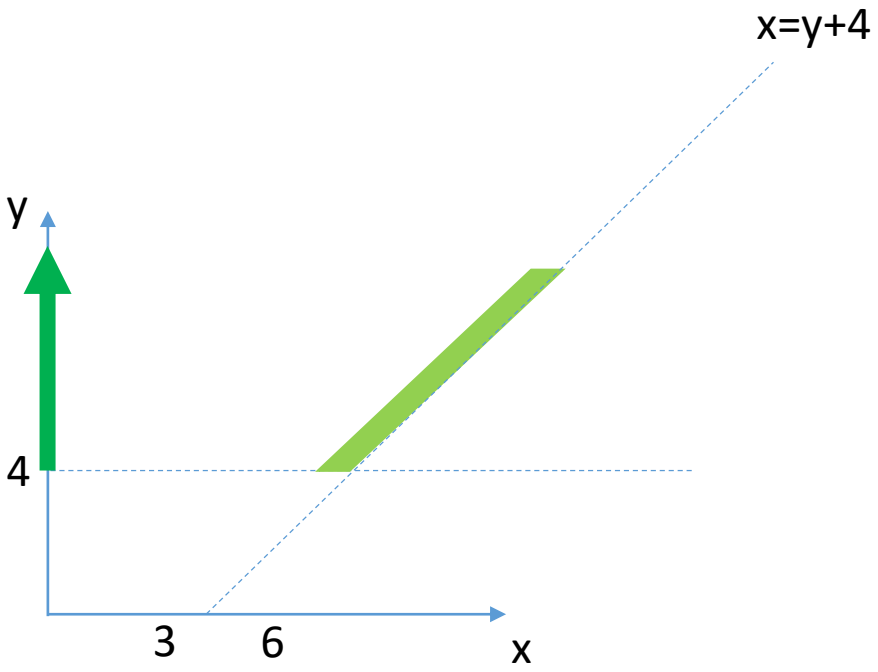




Zone

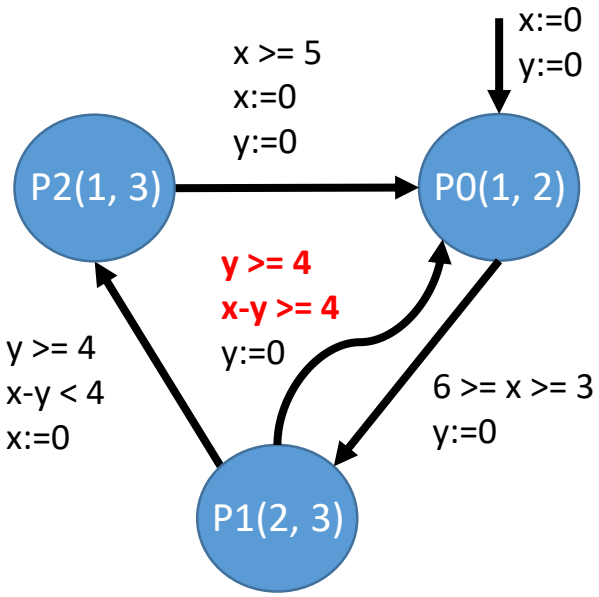
More compact state space representation

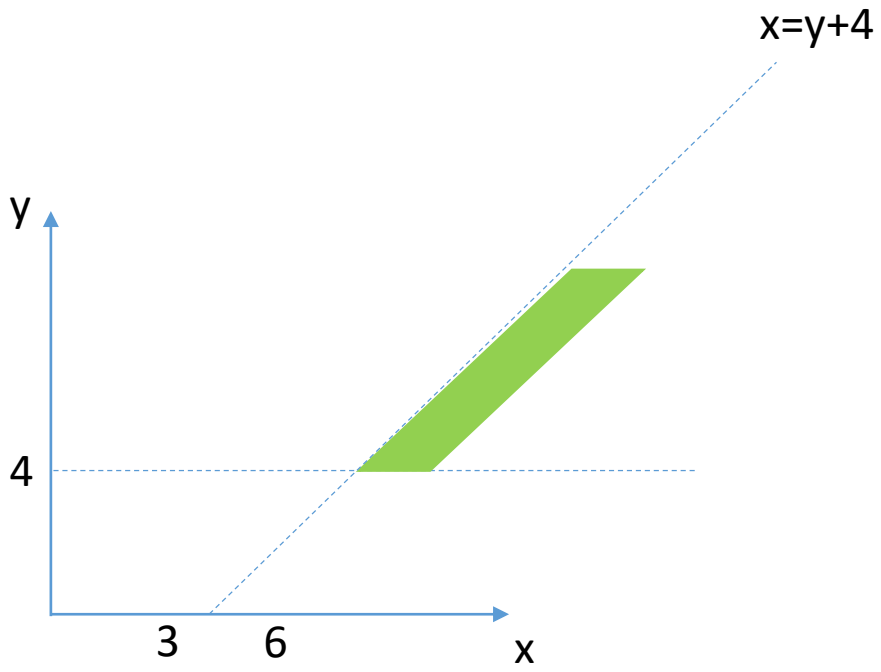




Zone

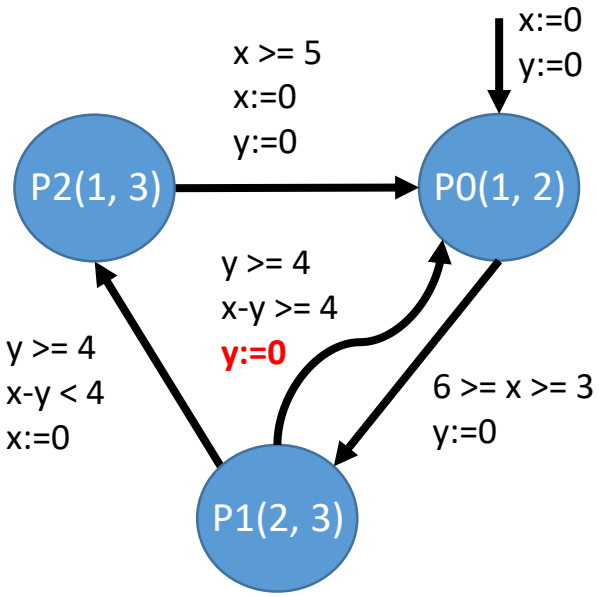
More compact state space representation

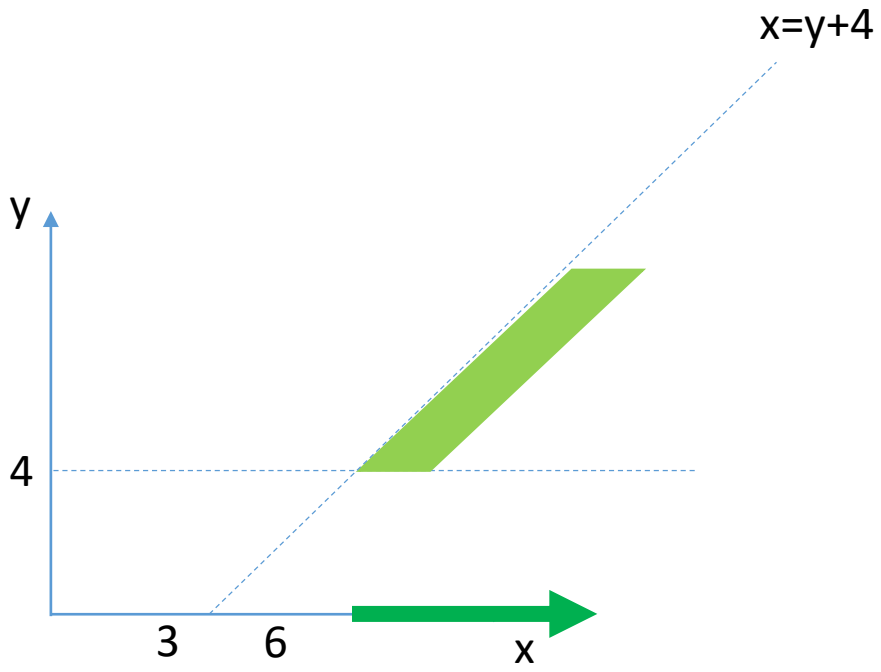




Zone

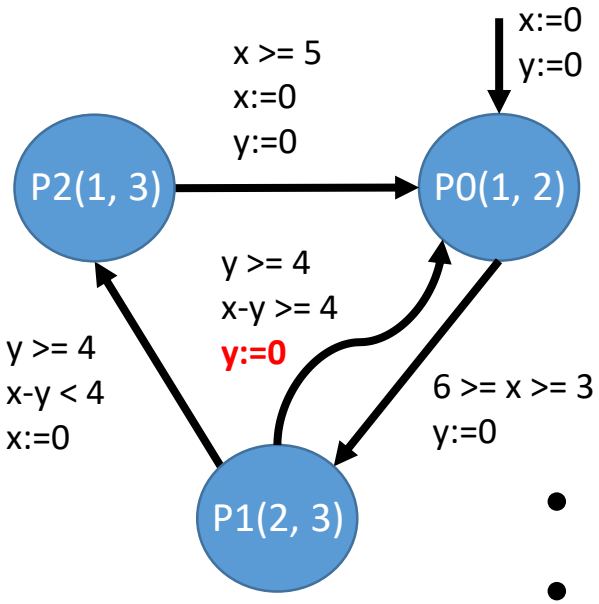
More compact state space representation

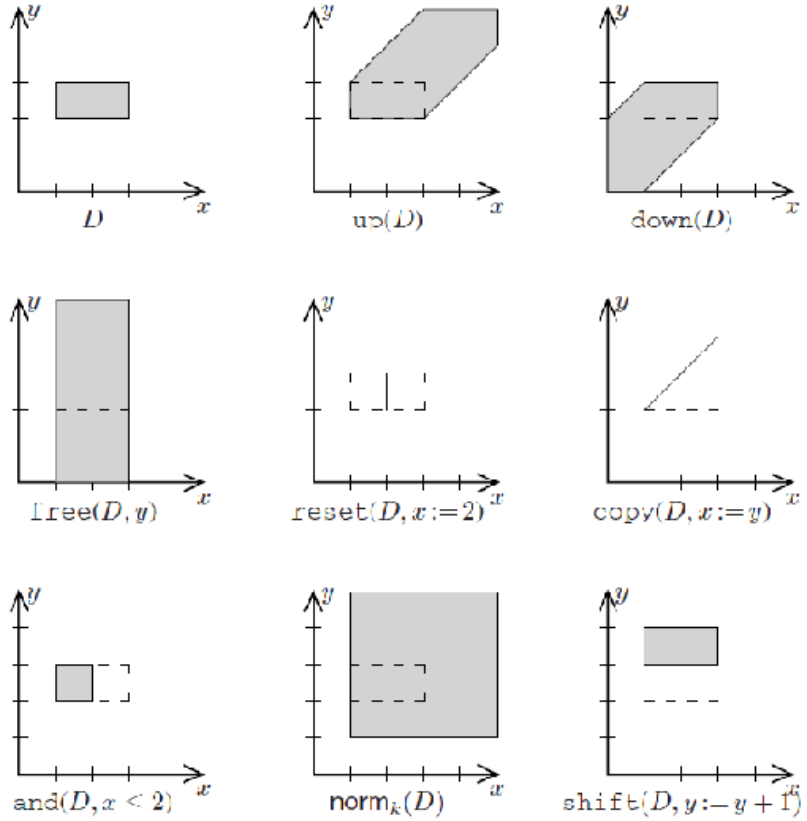




Zone

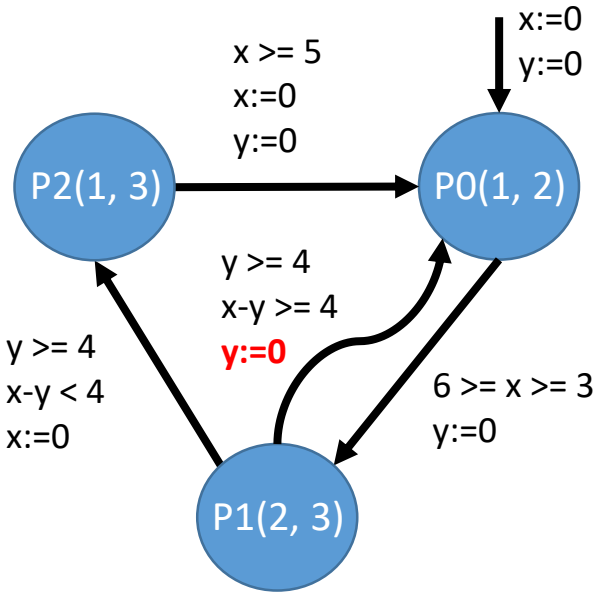
More compact state space representation



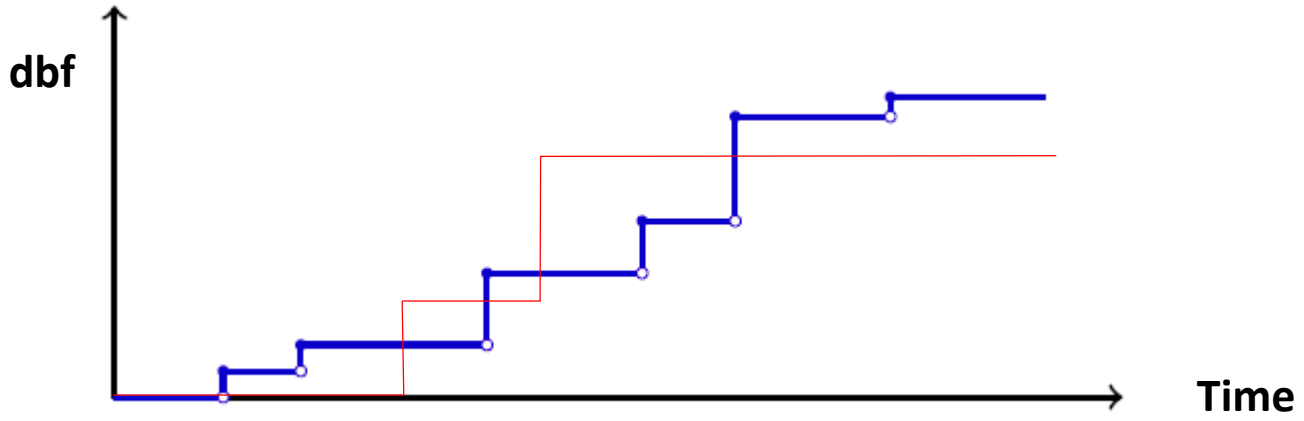


- Zone also preserves reachability
- A Zone is the solution set of a clock constraint
  - maximal set of clock assignments satisfying the constraint
  - Such sets can be efficiently represented and stored in memory as DBMs (Difference Bound Matrices)

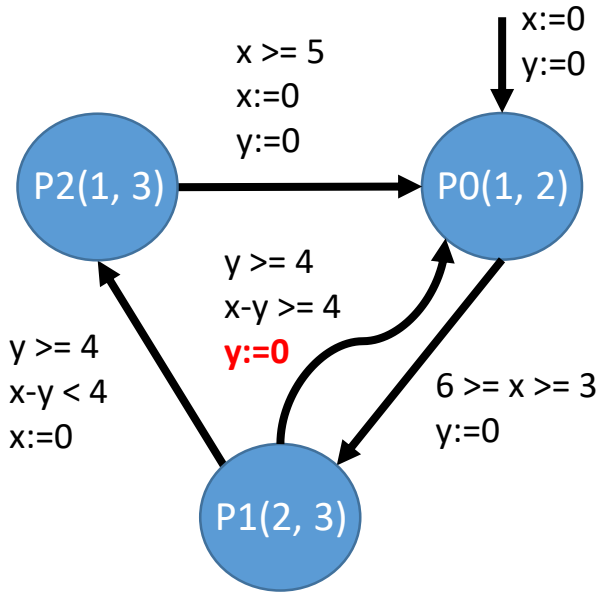
Computing DBF based on Zone



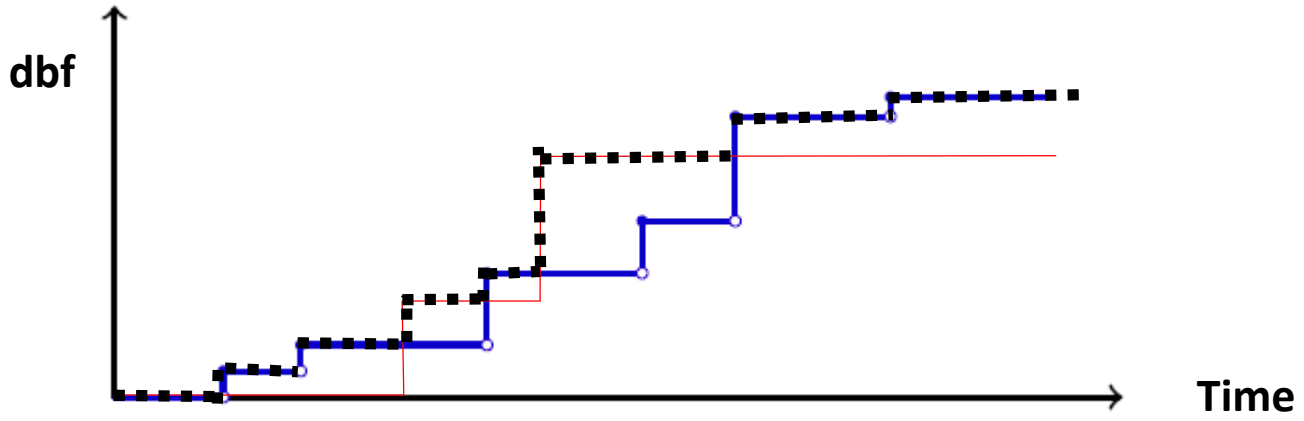
- Traverse the graph with Zone
- Split a Zone upon “branches”
- Each traversal trace generates a “demand trace”
  - the “worst case” must be on the border of Zones
- DBF = Maximization of all “demand functions”



Computing DBF based on Zone

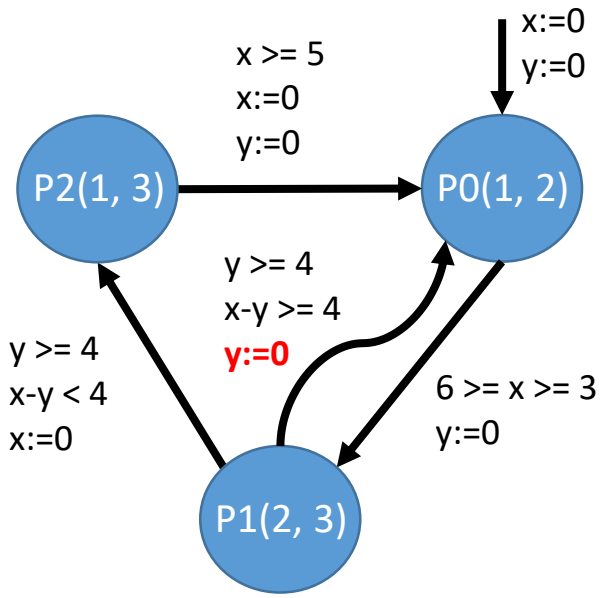


- Traverse the graph with Zone
- Split a Zone upon “branches”
- Each traversal trace generates a “demand trace”
  - the “worst case” must be on the border of Zones
- DBF = Maximization of all “demand functions”

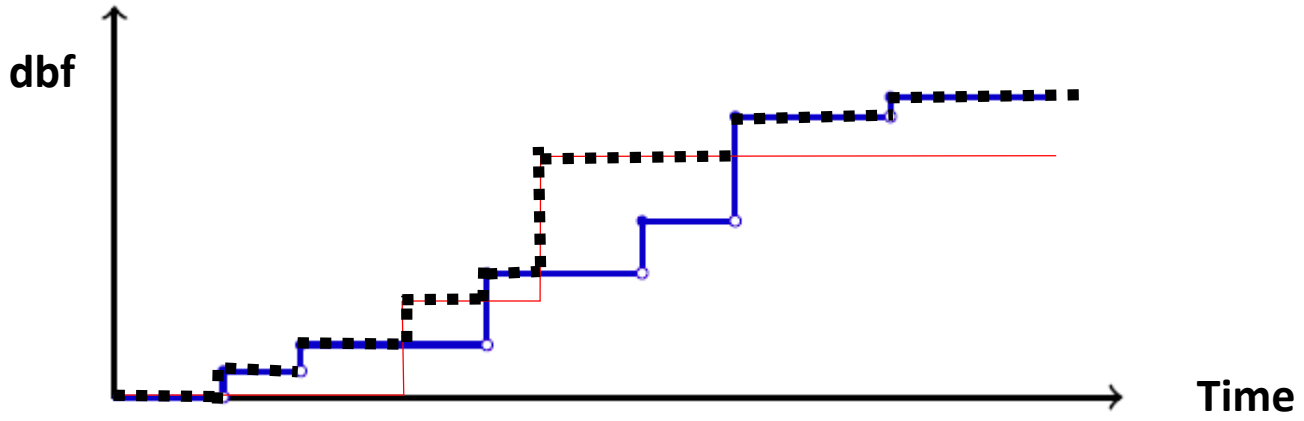


Computing DBF based on Zone

Exponential



- Traverse the graph with Zone
- Split a Zone upon “branches”
- Each traversal trace generates a “demand trace”
  - the “worst case” must be on the border of Zones
- DBF = Maximization of all “demand functions”



Summary

- Ultimate goal
  - Utilizing efficient abstraction and analysis techniques in scheduling theory to analyze the TA systems
- As the first step
  - Computing the DBF/RBF of an individual TA
- Starting point
  - Zone is a good starting point as state-space representation
    - Efficient storage and operations are known