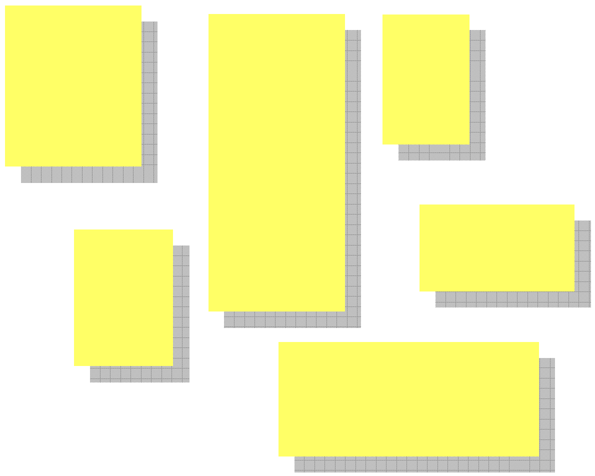
Performance Analysis, Scheduling and Synthesis of Embedded Systems

Kim G. Larsen
CISS – Aalborg University
DENMARK

DaNES

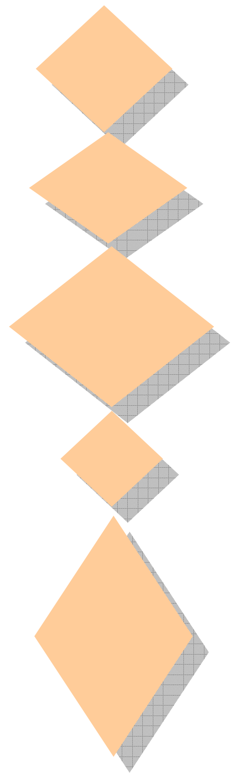
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Scheduling… in ES



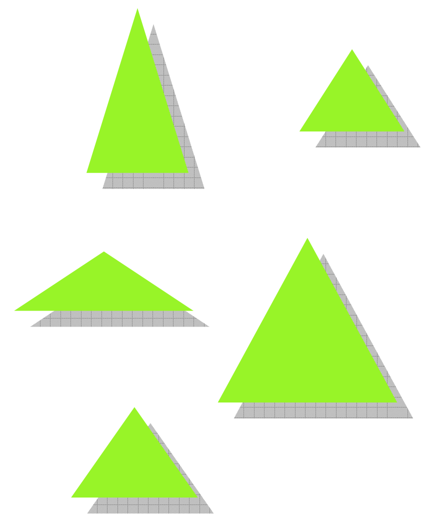
Tasks:

Computation times
Deadlines
Dependencies
Arrival patterns
uncertainties



Scheduling Principles (OS)

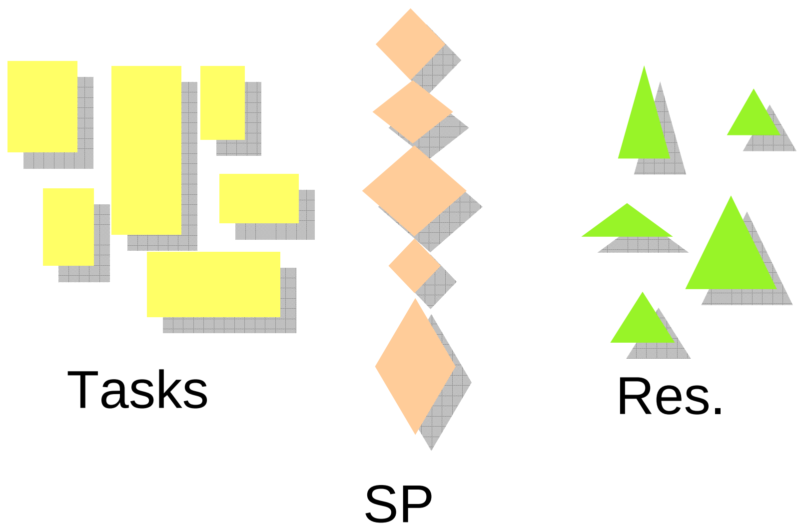
EDF, FPS, RMS, DVS, ..



Resources

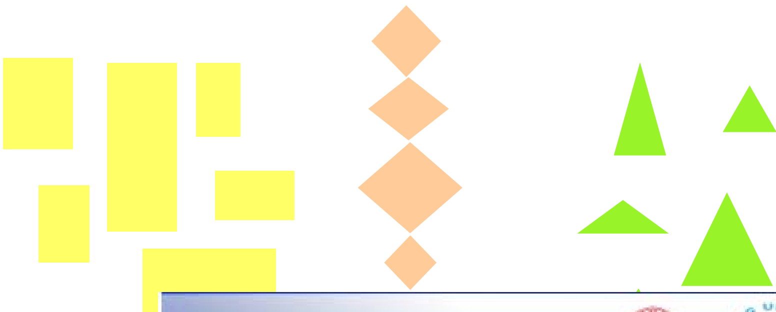
Execution platform
PE, Memory
Networks
Drivers
uncertainties

Issues

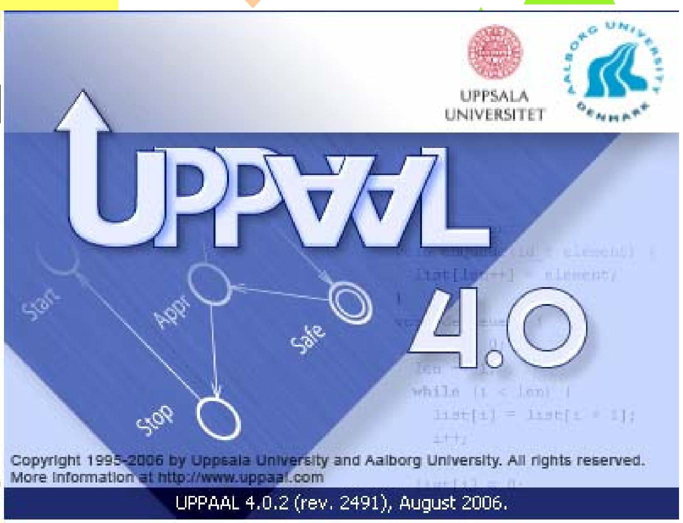


- Schedulability Analysis
  - Verify that given SP ensures deadlines.
- Performance Evaluation
  - Estimate resources (e.g. energy) required by given SP.
- Scheduling & Synthesis
  - Synthesize (optimal) SP ensuring given objective.
  - *Scheduling*: SP controls everything (including ex.time).
  - *Synthesis*: scheduling under uncertainties (e.g. execution time, availability of resources).

Approach – TA



Tasl



- Sch
  - Verify that given SP ensures deadlines.

- Performance Evaluation
  - Estimate resources (e.g. energy) required by given

CLASSIC

- Sc ... Synthesis

CORA

  - ... optimal) SP ... given objective.

TIGA

  - *Scheduling*: SP controls

TALK:

What can we do?

What can we do efficiently?

What can not be done?

What would we like to do?

Timed Automata

[Alur & Dill'89]

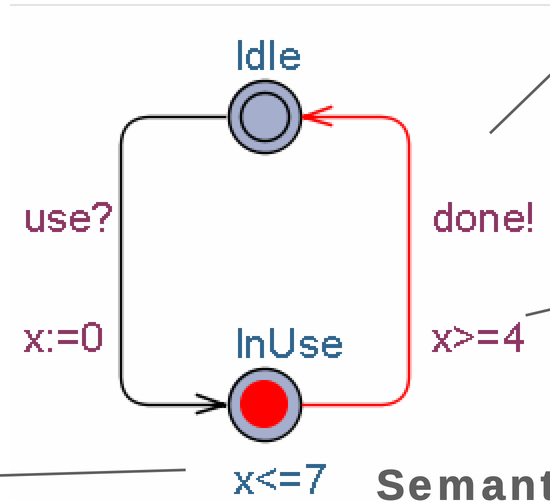
Resource

Synchronization

Reset

Guard

Invariant

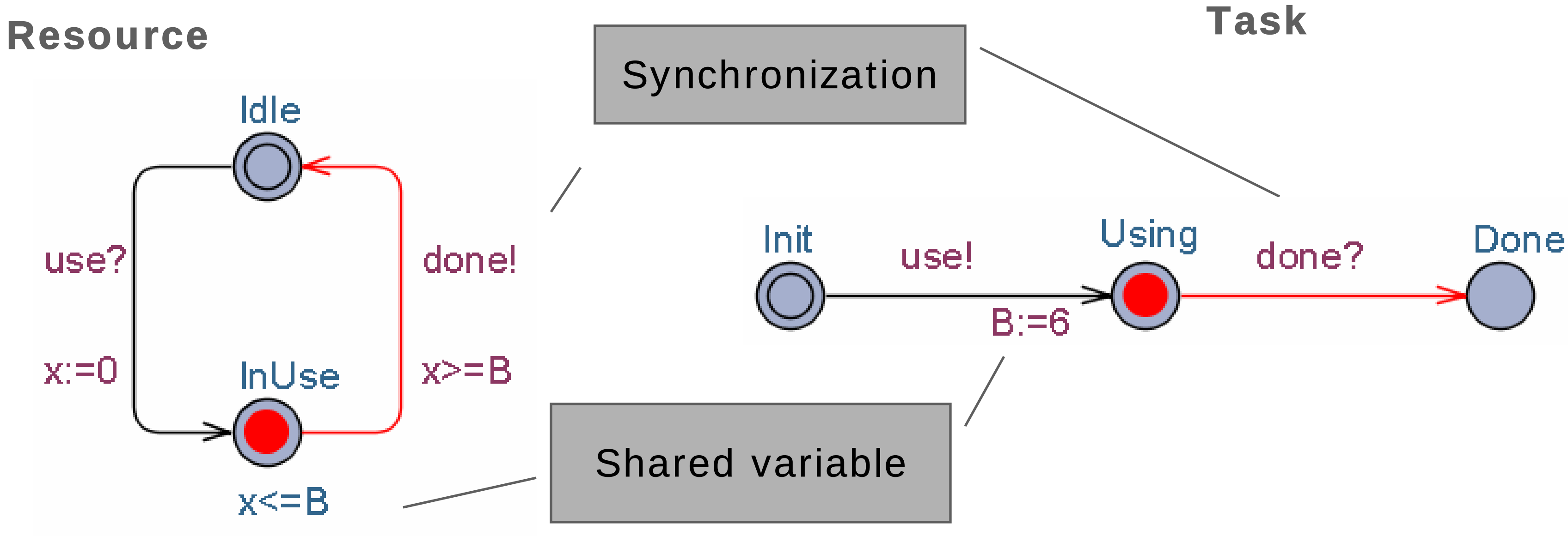


Semantics:

(Idle , x=0)

→ (Idle , x=2.5)	d(2.5)
→ (InUse , x=0)	use?
→ (InUse , x=5)	d(5)
→ (Idle , x=5)	done!
→ (Idle , x=8)	d(3)
→ (InUse , x=0)	use?

Composition



Semantics:

(Idle , Init , B=0, x=0)

→ (Idle , Init , B=0 , x=3.1415) d(3.1415)

→ (InUse , Using , B=6, x=0) use

→ (InUse , Using , B=6, x=6) d(6)

→ (Idle , Done , B=6 , x=6) done

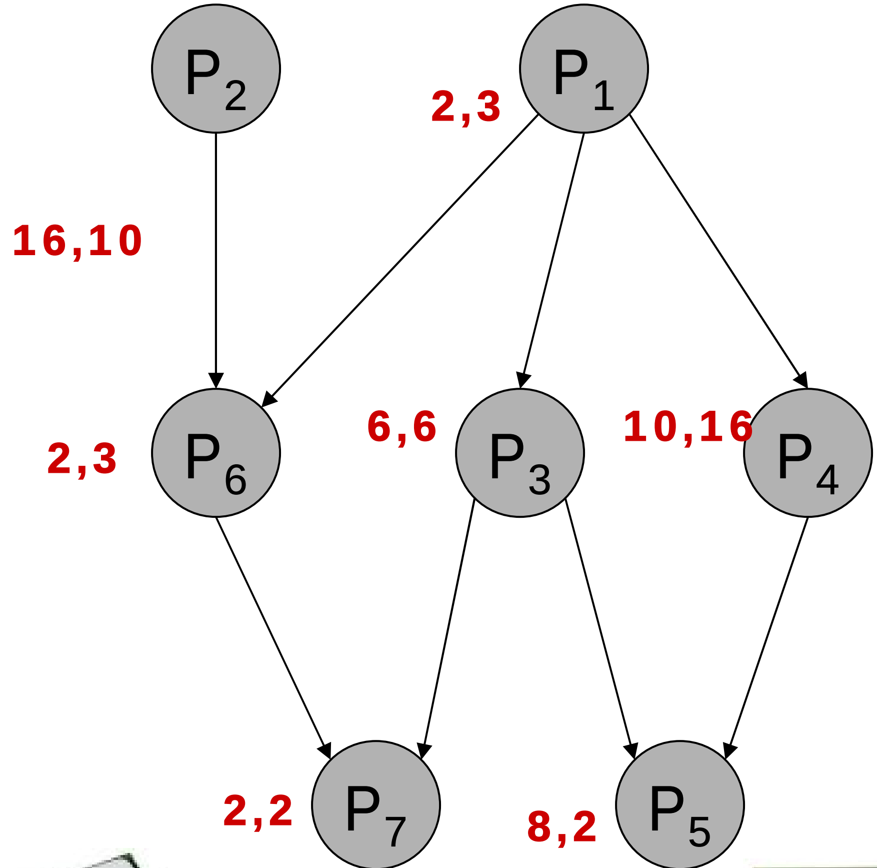
Task Graph Scheduling

Optimal Static Task Scheduling

- Task **P**$=\{P_1,.., P_m\}$
- Machines **M**$=\{M_1,..,M_n\}$
- Duration $\Delta : (\mathbf{P}\times\mathbf{M}) \rightarrow \mathbf{N}_\infty$
- $<$: p.o. on **P** (pred.)

- A task can be executed only if all predecessors have completed
- Each machine can process at most one task at a time
- Task cannot be preempted.

- Compute schedule with minimum completion-time!

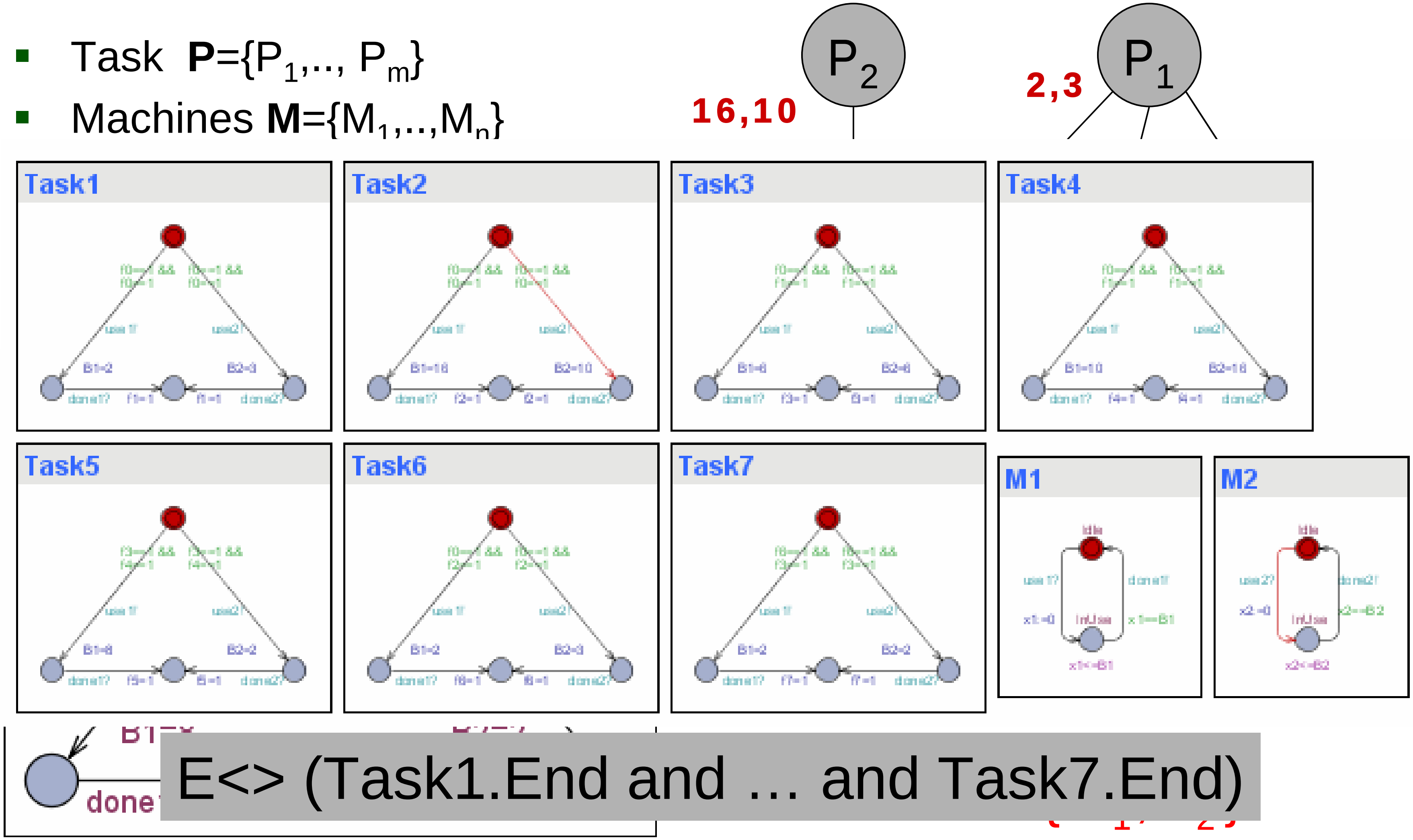


M $= \{M_1, M_2\}$

Task Graph Scheduling

Optimal Static Task Scheduling

- Task $\mathbf{P}=\{P_1,.., P_m\}$
- Machines $\mathbf{M}=\{M_1,..,M_n\}$



E<> (Task1.End and … and Task7.End)

Experimental Results

name	#tasks	#chains	# machines	optimal	TA
001	437	125	4	1178	1182
000	452	43	20	537	537
018	730	175	10	700	704
074	1007	66	12	891	894
021	1145	88	20	605	612
228	1187	293	8	1570	1574
071	1193	124	20	629	634
271	1348	127	12	1163	1164
237	1566	152	12	1340	1342
231	1664	101	16	t.o.	1137
235	1782	218	16	t.o.	1150
233	1980	207	19	1118	1121
294	2014	141	17	1257	1261
295	2168	965	18	1318	1322
292	2333	318	3	8009	8009
298	2399	303	10	2471	2473

AMETIST
advanced methods for timed systems

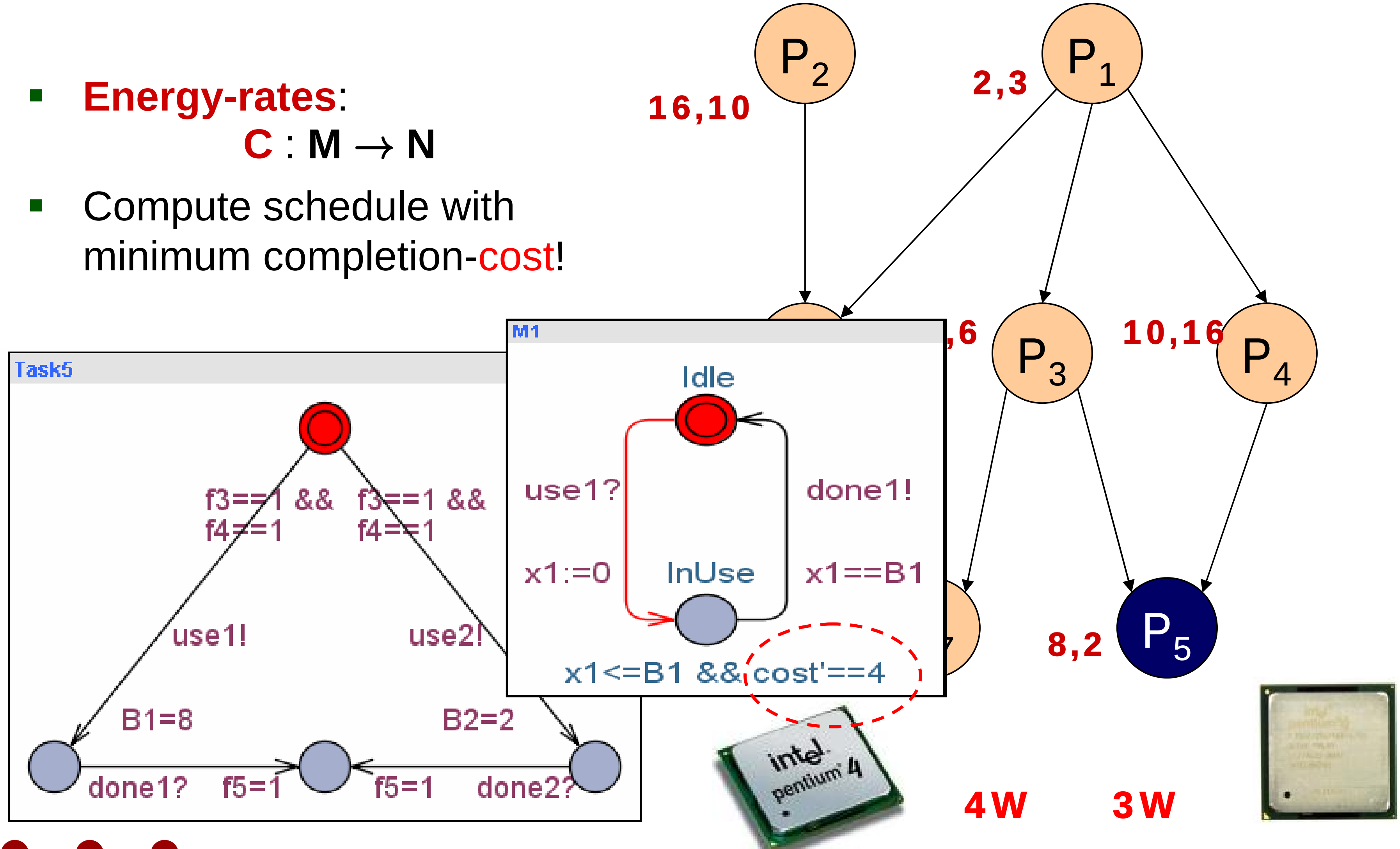
Symbolic A*
Brand-&-Bound
60 sec

Abdeddaïm, Kerbaa, Maler

Optimal Task Graph Scheduling

Power-Optimality

- **Energy-rates**:
  C : **M** $\rightarrow$ **N**
- Compute schedule with minimum completion-cost!

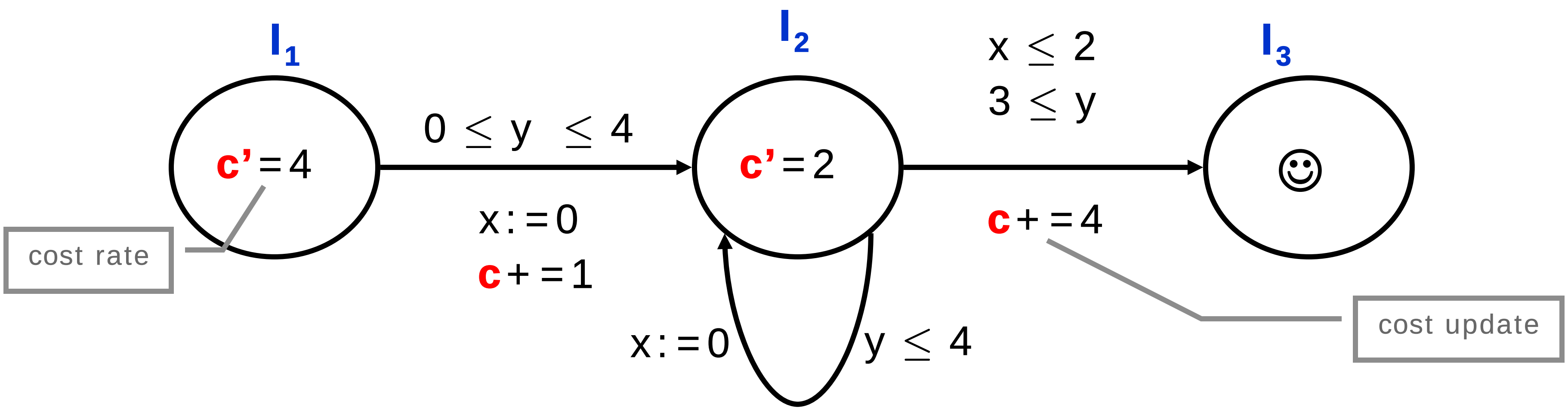


Priced Timed Automata

Behrmann, Fehnker, et all (HSCC'01)

Alur, Torre, Pappas (HSCC'01)

Timed Automata + **COST** variable

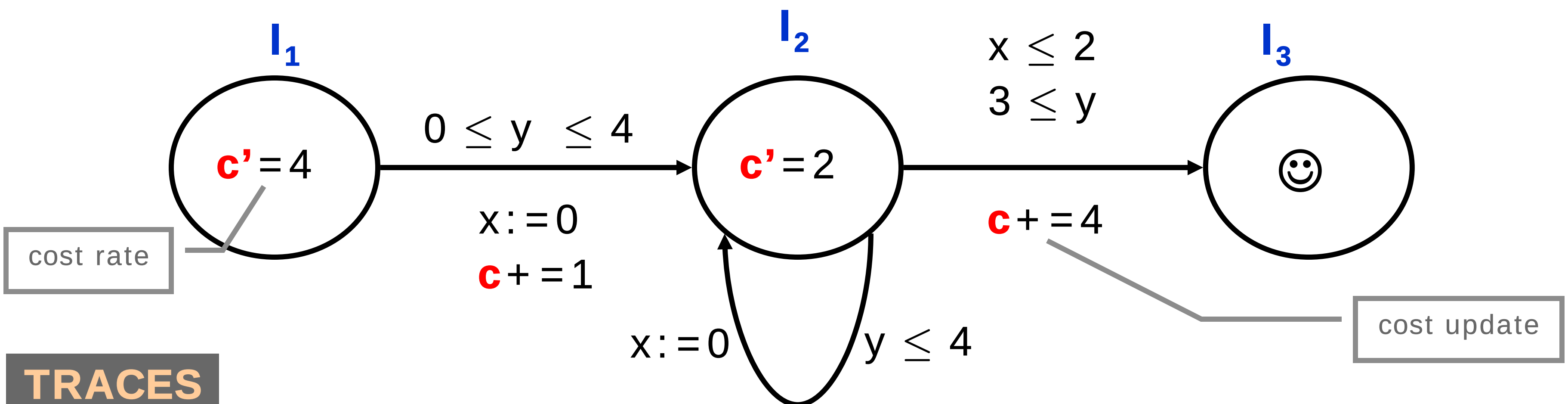


Priced Timed Automata

Behrmann, Fehnker, et all (HSCC'01)

Alur, Torre, Pappas (HSCC'01)

Timed Automata + **COST** variable



TRACES

$$(\mathbf{l_1}, x=y=0) \overset{\varepsilon(3)}{\underset{12}{\dashrightarrow}} (\mathbf{l_1}, x=y=3) \xrightarrow[1]{} (\mathbf{l_2}, x=0, y=3) \xrightarrow[4]{} (\mathbf{l_3}, \_, \_)$$

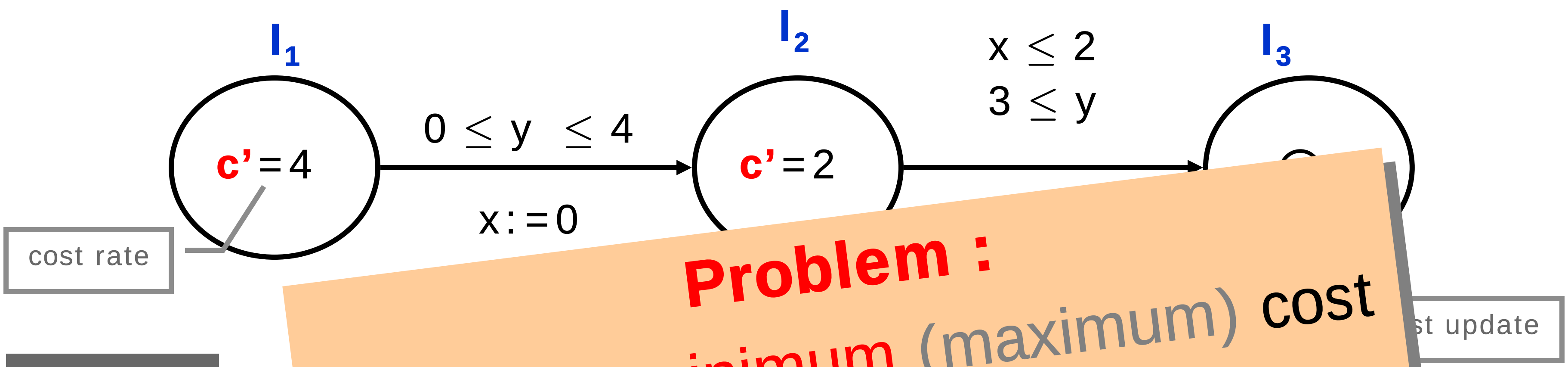
$\sum c = 17$

Priced Timed Automata

Behrmann, Fehnker, et all (HSCC'01)

Alur, Torre, Pappas (HSCC'01)

Timed Automata + **COST** variable



TRACES

$(l_1, x=y=0)$

$(l_1, x=y=0)$ ε(

$(l_1, x=y=0) \xrightarrow[1]{} (l_2, x=0, y=0$ ... $\xrightarrow[4]{} (l_3, \_, \_)$ $\sum c=11$

$(l_3, \_, \_)$ $\sum c=16$

$\sum c=17$

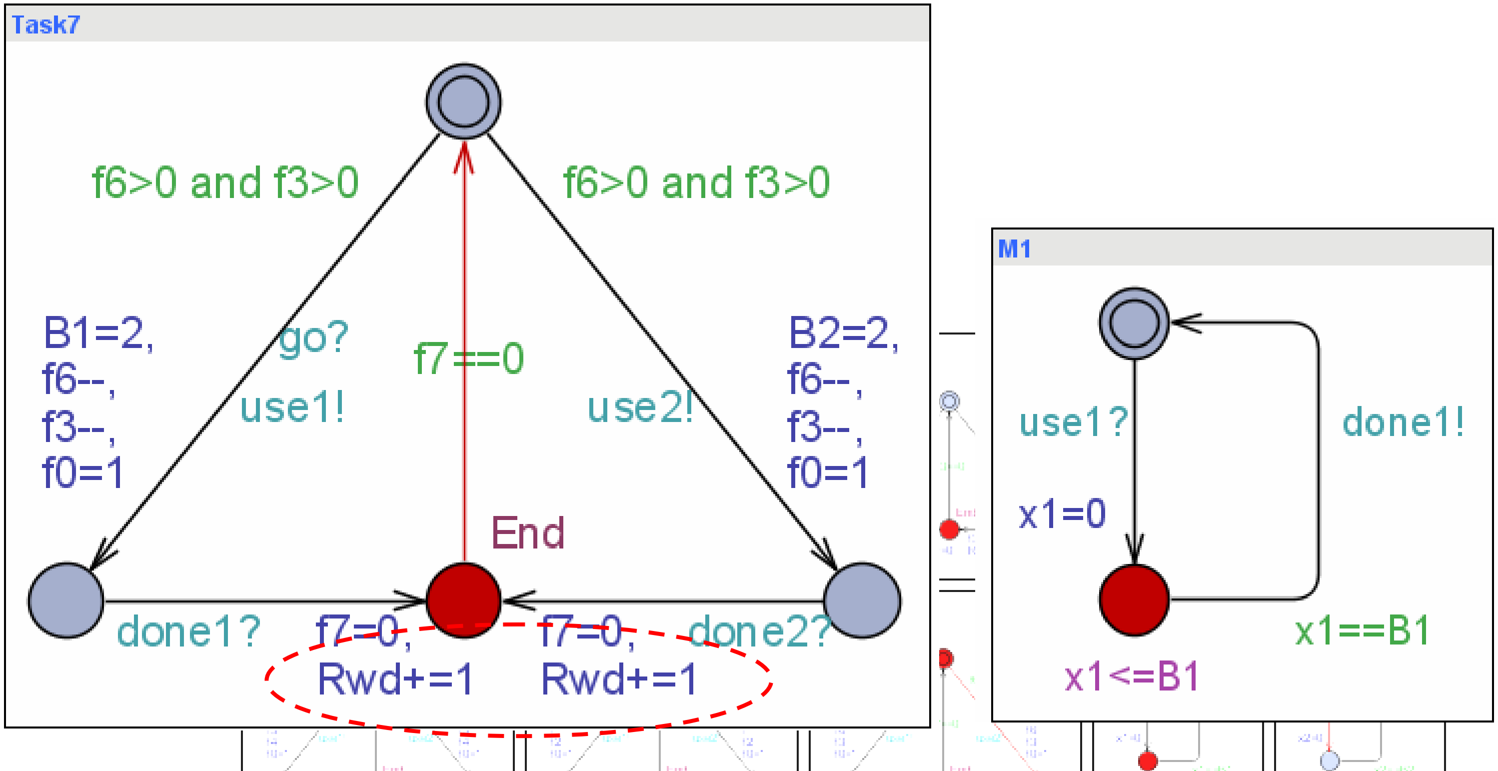
Problem :

Find the minimum (maximum) cost of reaching location l_3

Efficient Implementation:
CAV'01 and TACAS'04

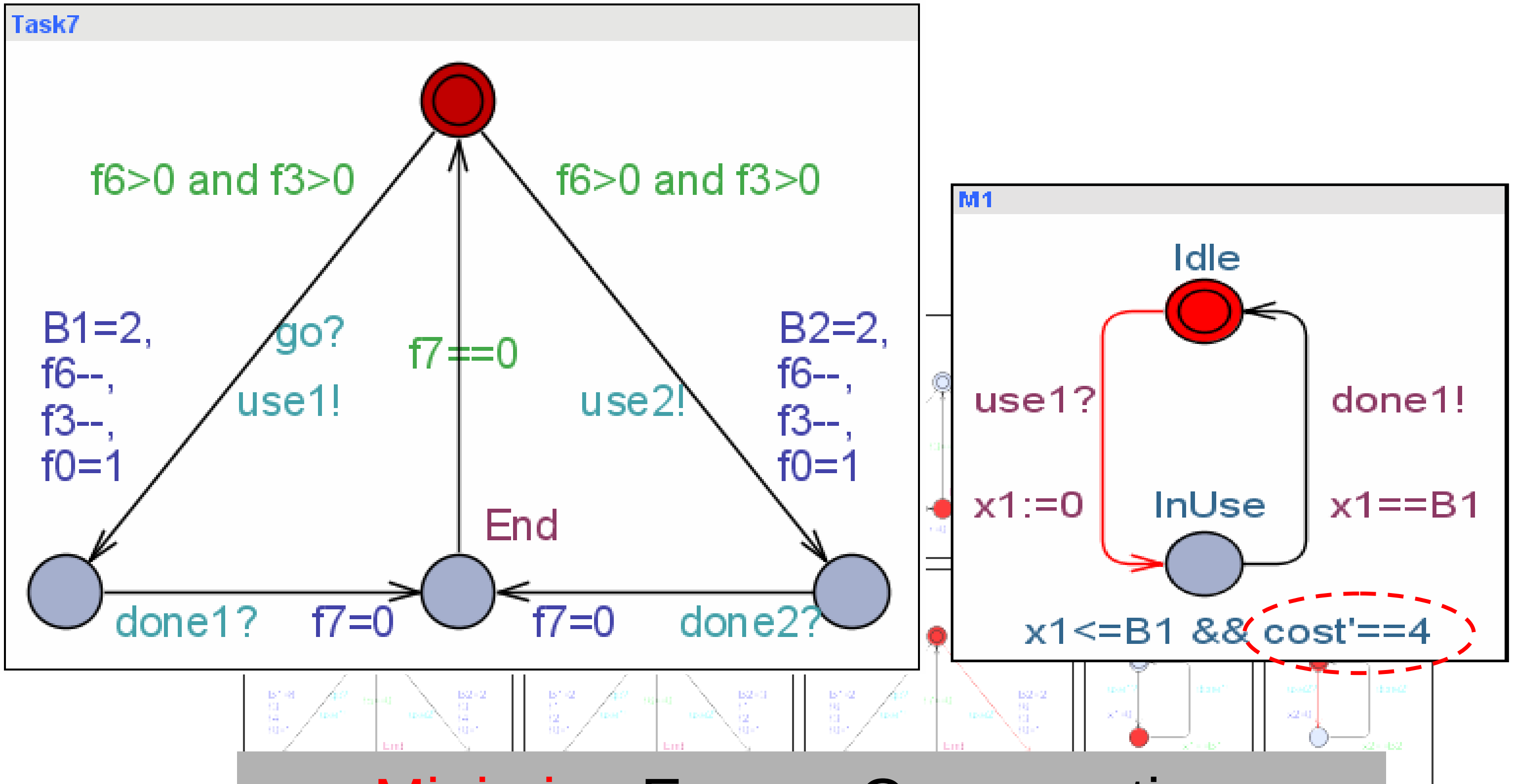
Competitive with MILP and commercial tool (Axxon)

Optimal Infinite Scheduling



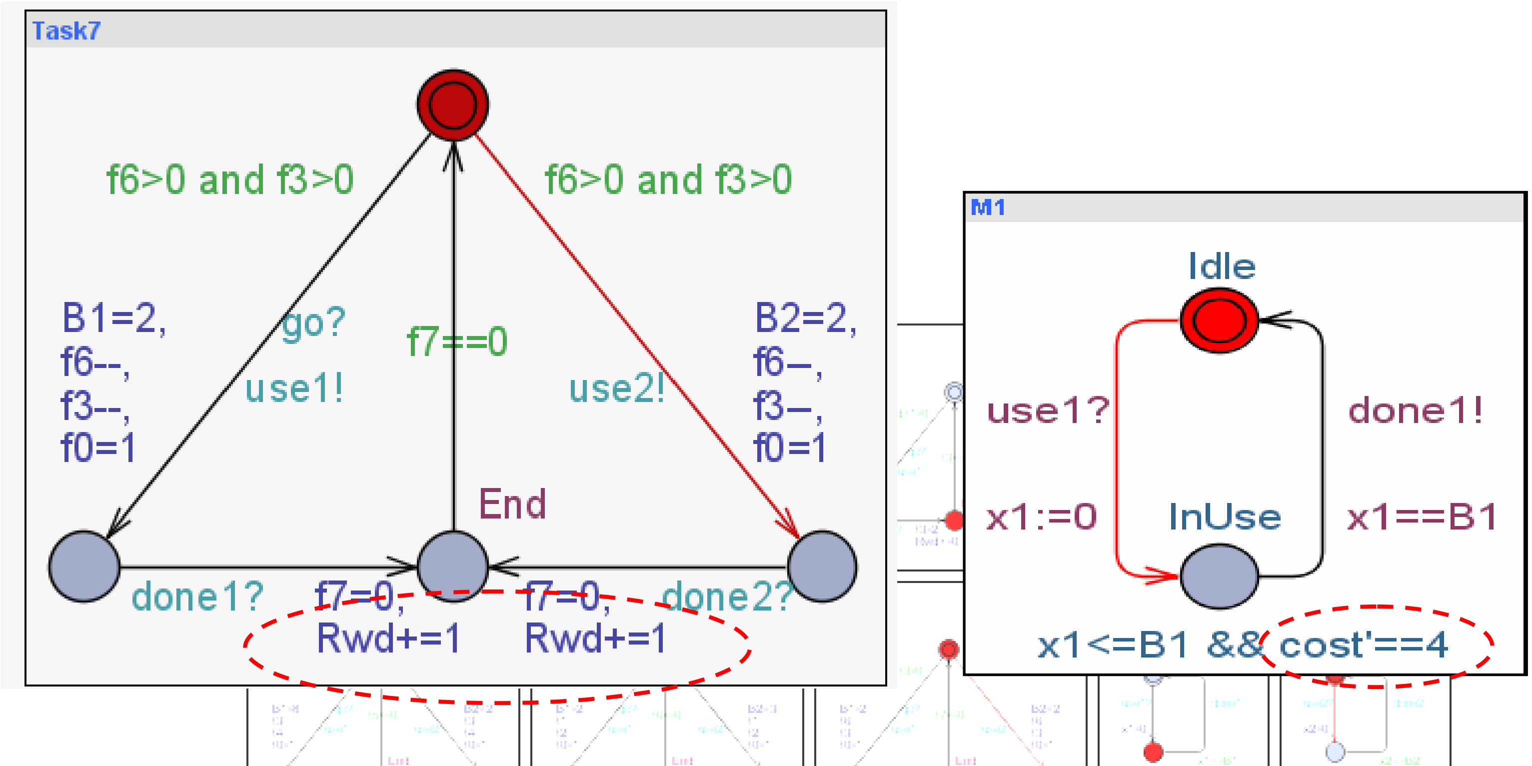
Maximize throughput:
i.e. maximize Reward / Time in the long run!

Optimal Infinite Scheduling



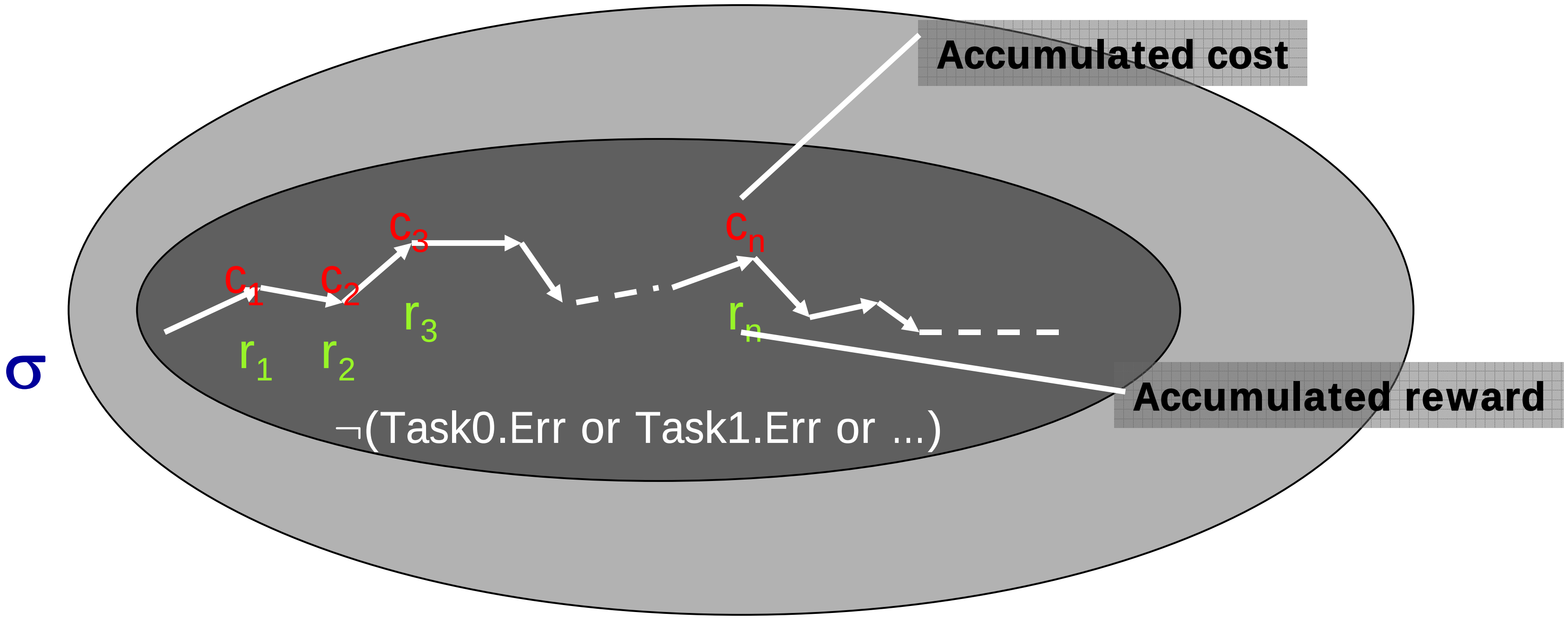
Minimize Energy Consumption:
i.e. minimize Cost / Time in the long run

Optimal Infinite Scheduling



Maximize throughput:
i.e. maximize Reward / Cost in the long run

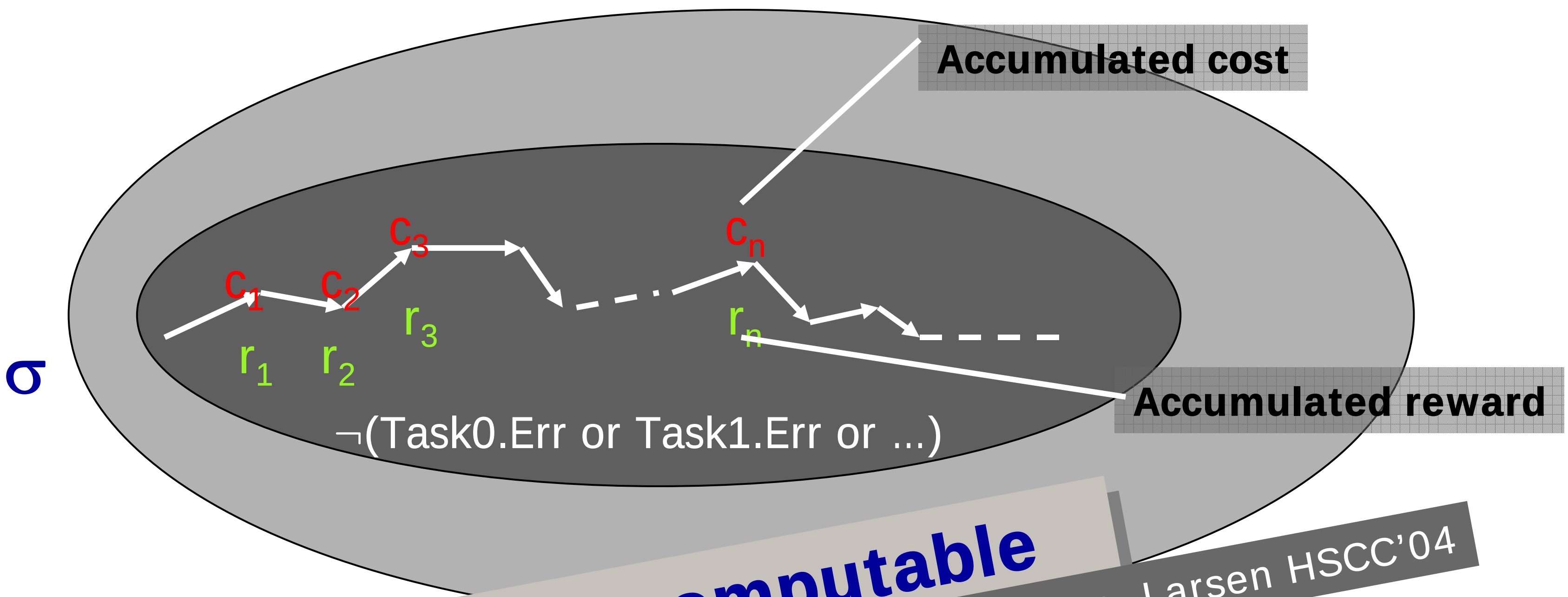
Cost Optimal Scheduling =
Optimal Infinite Path



Value of path σ: $val(\sigma) = \lim_{n\to\infty} c_n/r_n$

Optimal Schedule σ^*: $val(\sigma^*) = \inf_\sigma val(\sigma)$

Cost Optimal Scheduling = *Optimal Infinite Path*



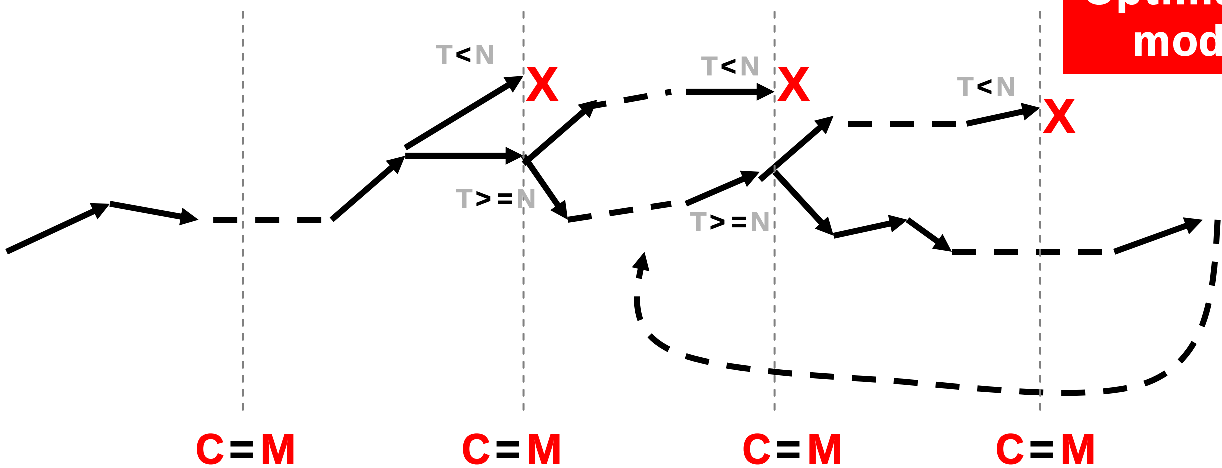
THEOREM: σ^* is computable

Bouyer, Brinksma, Larsen HSCC'04

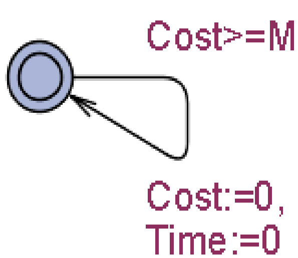
… path σ: val(σ) = $\lim_{n\to\infty} c_n/r_n$

Optimal Schedule σ^*: val(σ^*) = $\inf_\sigma$ val(σ)

Approximate Optimal Schedule



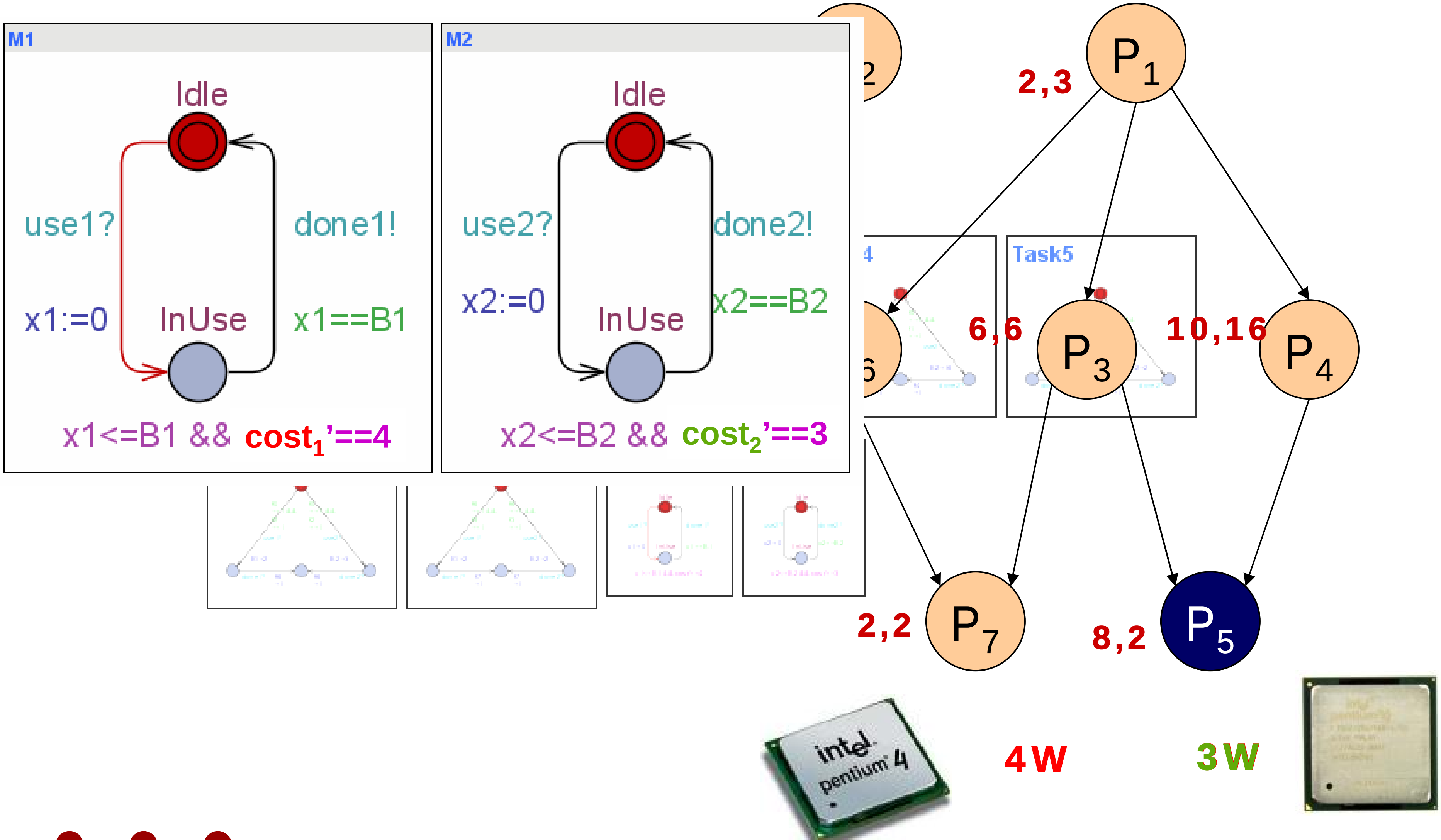
Optimal infinite schedule modulo cost-horizon



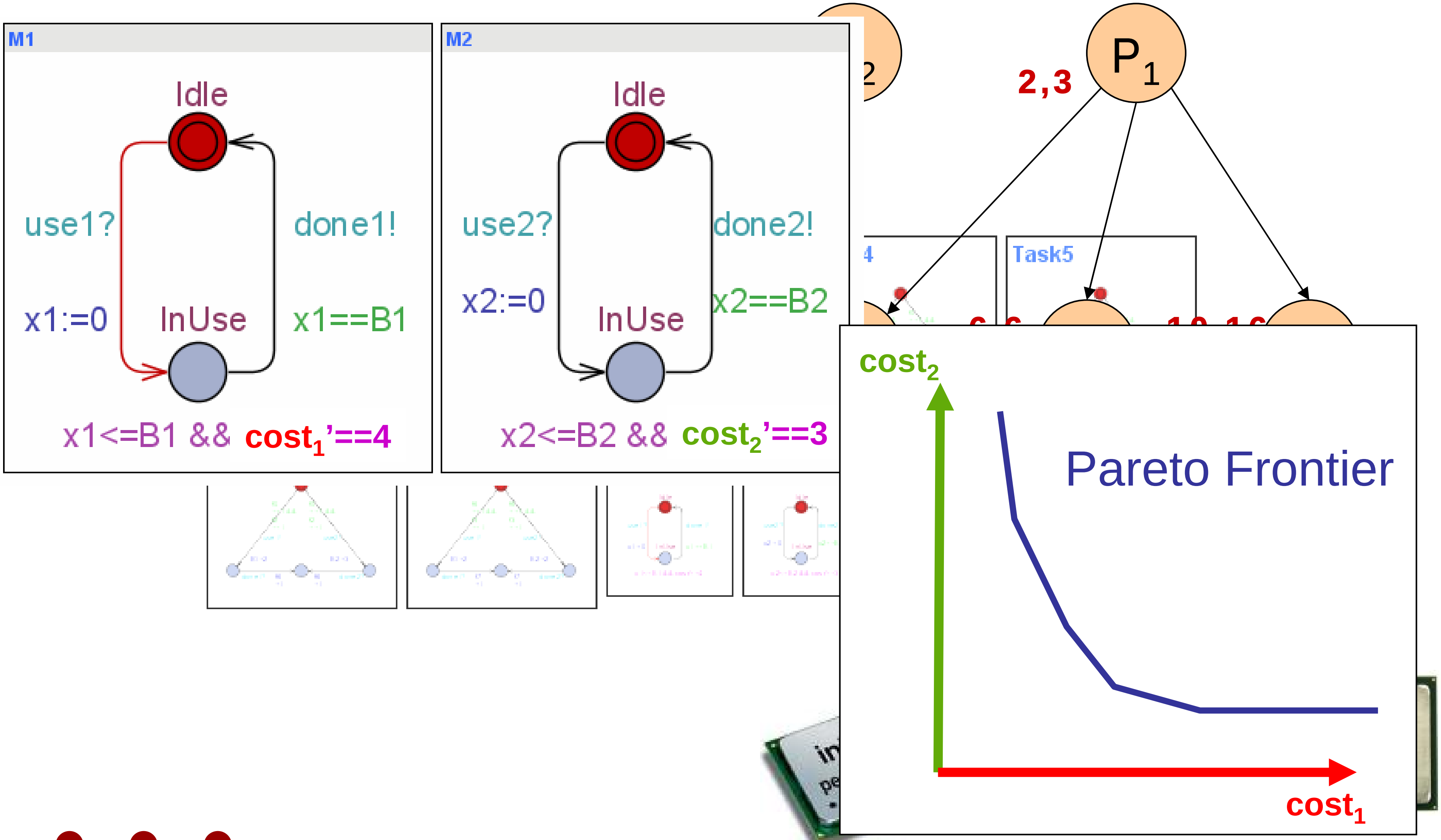
E[] (not (Task0.Error or Task1.Error or Task2.Error)
and
(cost>=M imply time >= N))
=
E[] ϕ(M ,N)

$\sigma \models [] \phi(M,N)$ **imply val(σ)$\leq$ M/N**

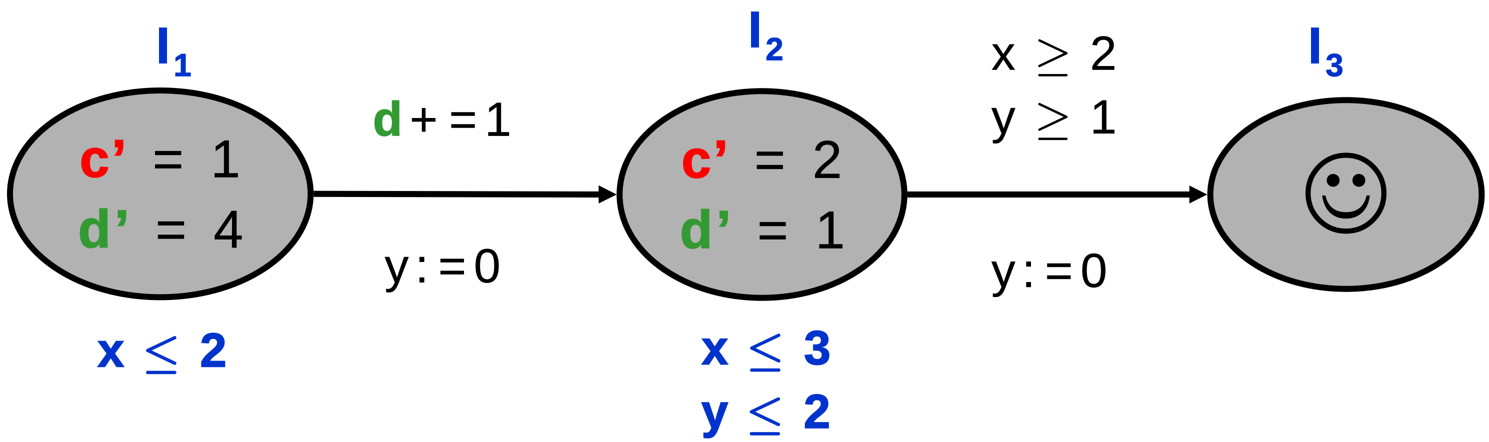
Multiple Objective Scheduling



Multiple Objective Scheduling



Multi Priced Timed Automata



PROBLEM:

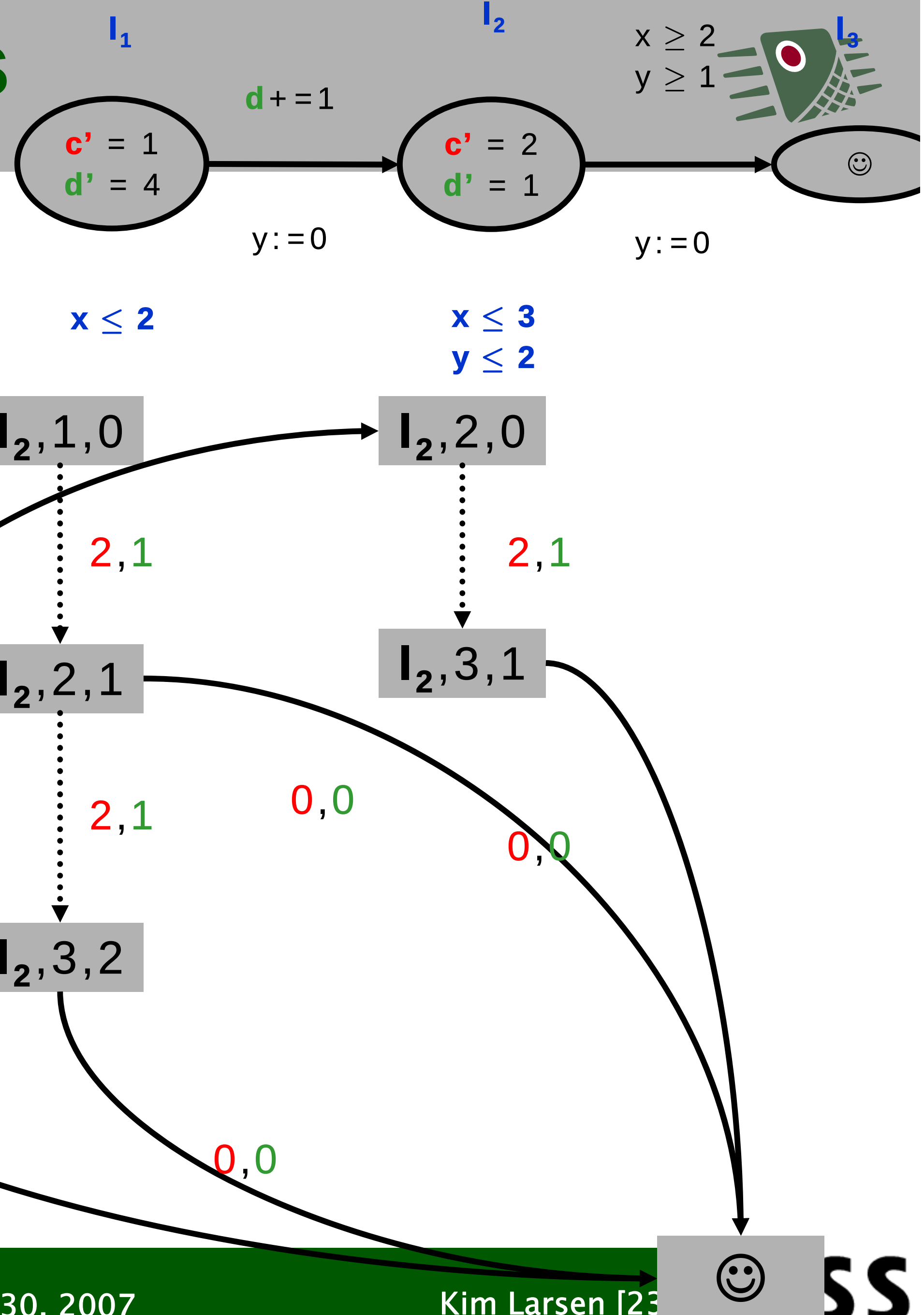
Reach l_3 in a way which minimizes **c** subject to **d** ≤ 4

SOLUTION:

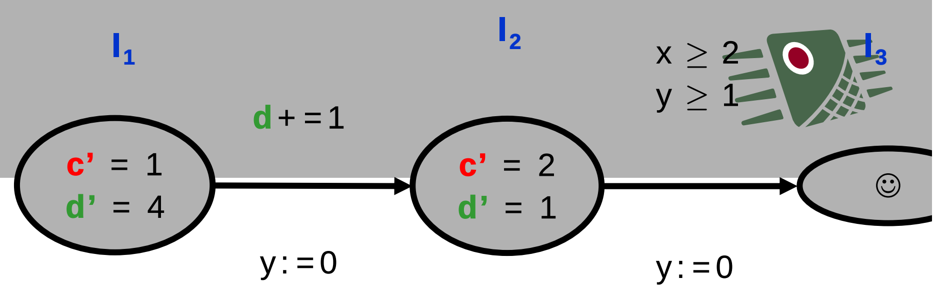
c = 11/3 →

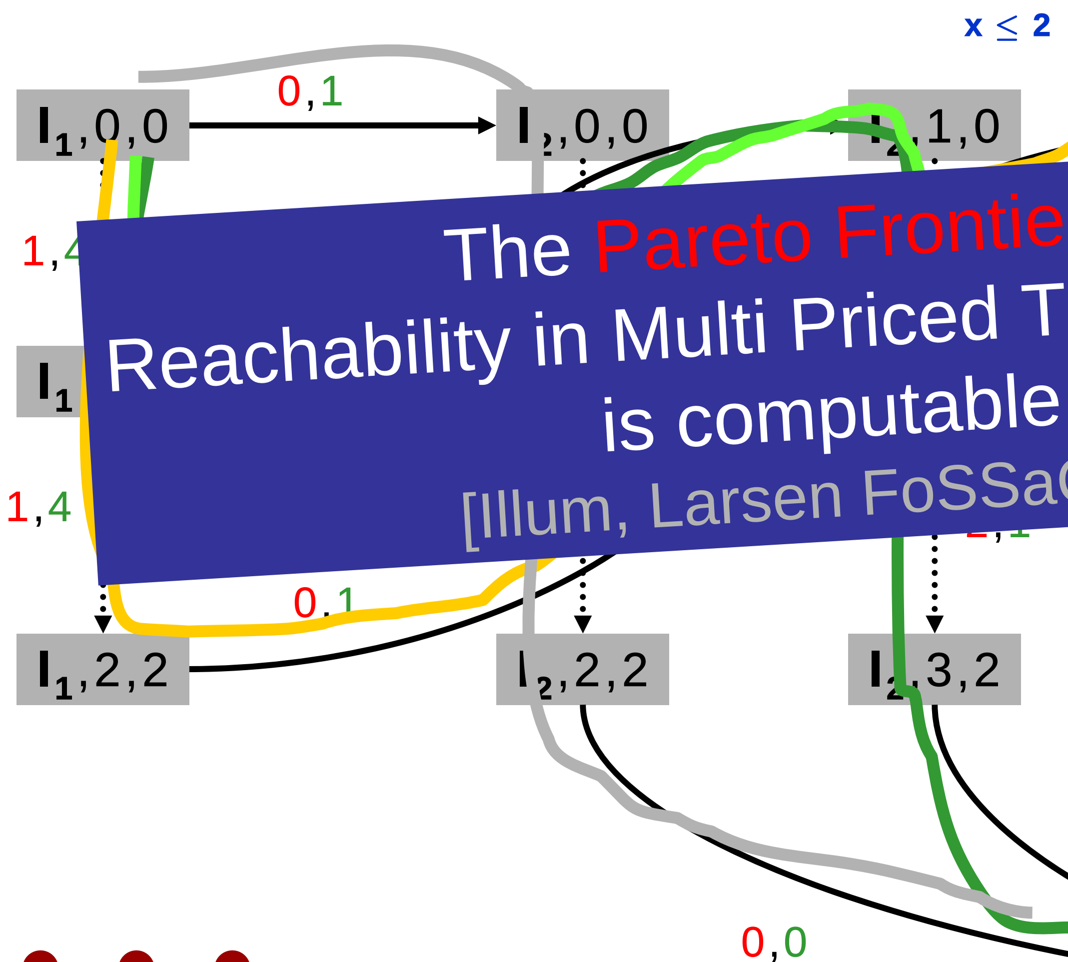
wait 1/3 in l_1; goto l_2;
wait 5/3 in l_2; goto l_3

Discrete Trajectories

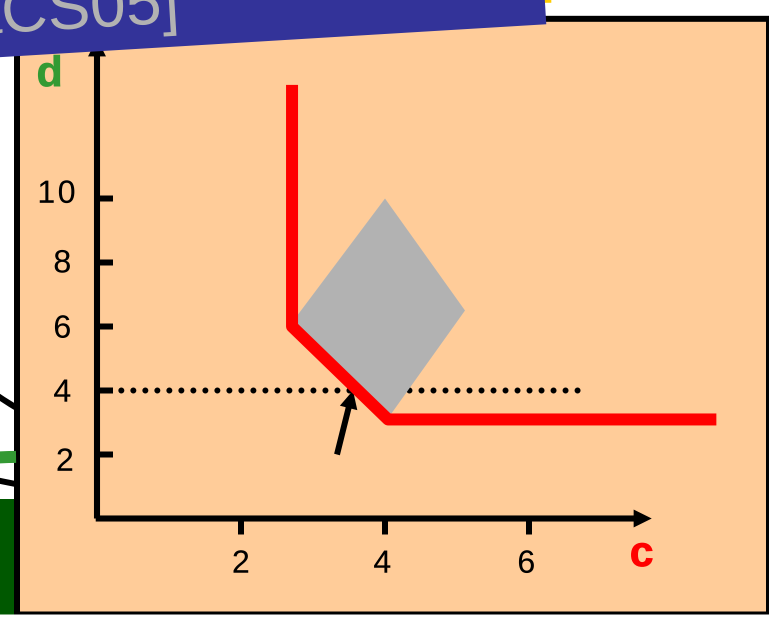


Discrete Trajectories



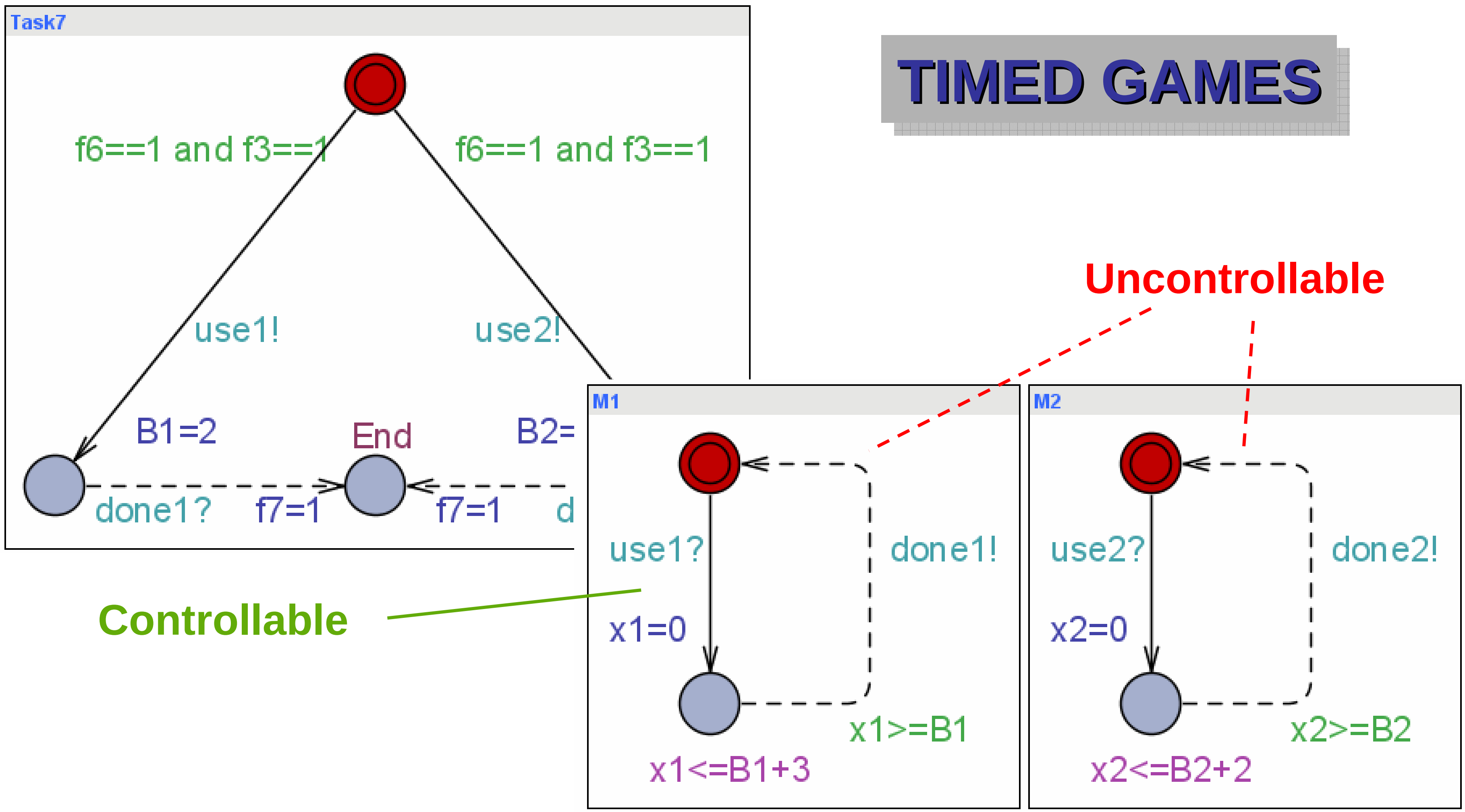


The Pareto Frontier for
Reachability in Multi Priced Timed Automata
is computable
[Illum, Larsen FoSSaCS05]



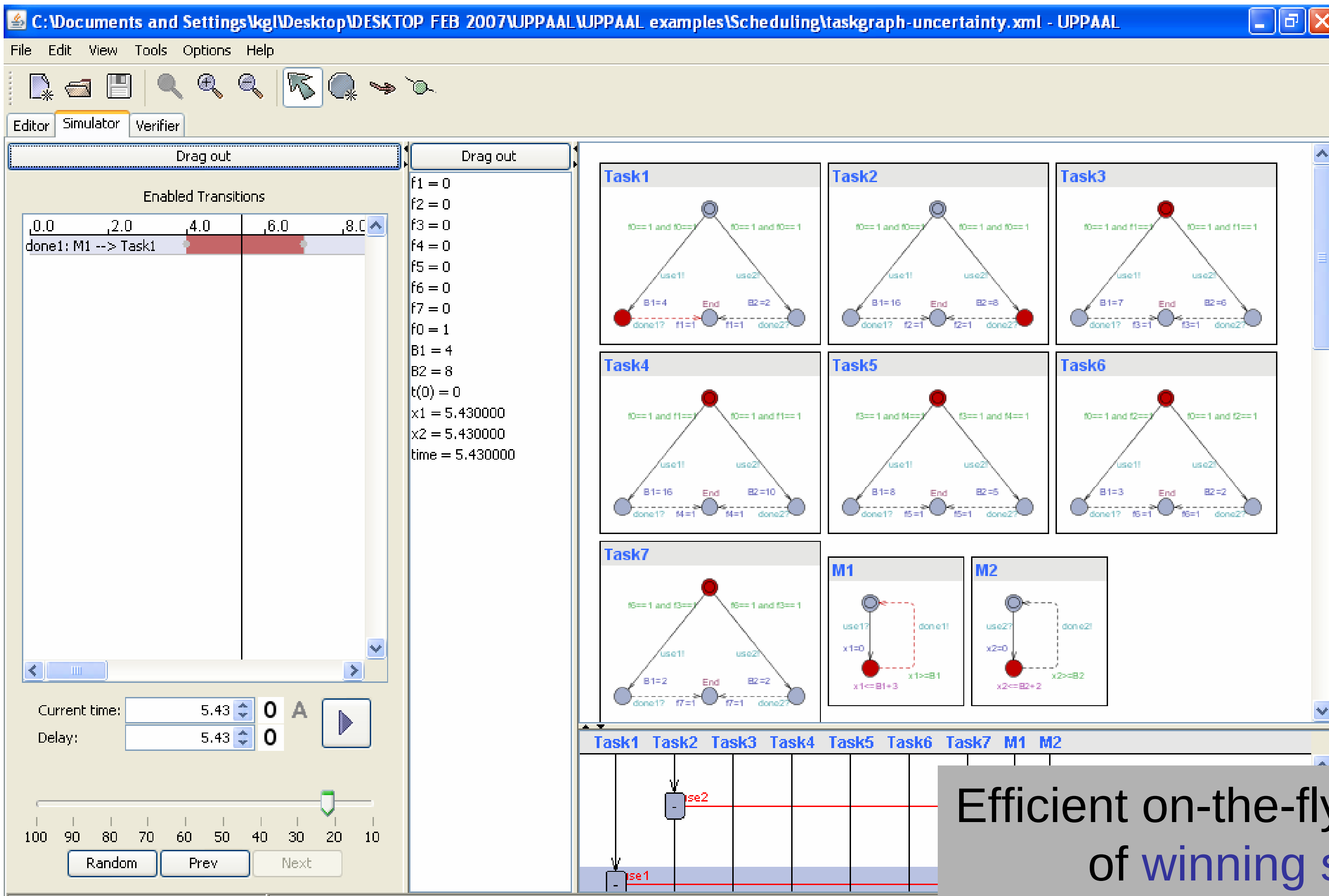
Synthesis =
Scheduling under uncertainty

TIMED GAMES



UPPAAL Tiga

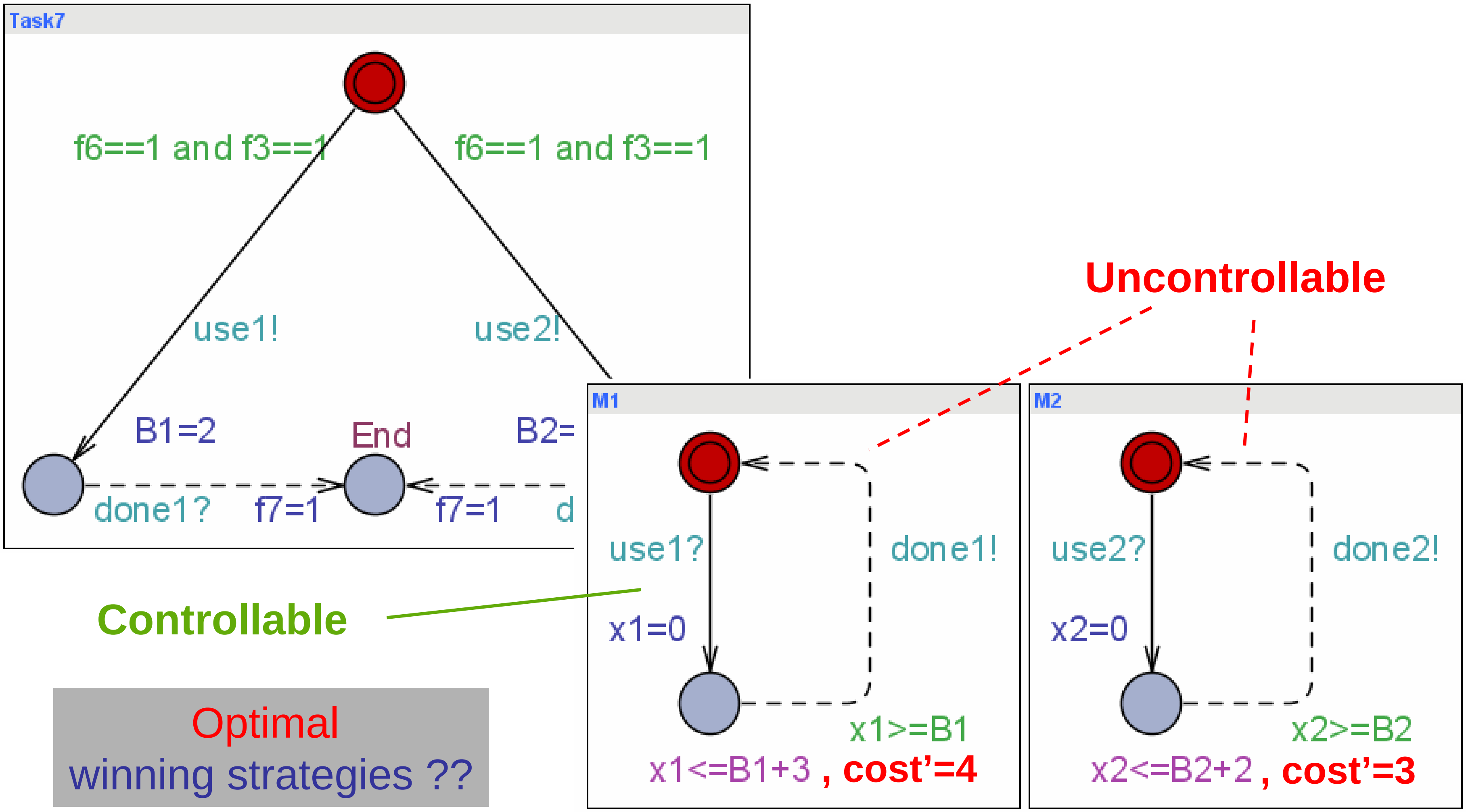
Synthesis of winning strategies for TIMED GAMES



CONCUR05,
CAV07,
FORMATS07

Efficient on-the-fly generation of winning strategies for safety & liveness objectives

Optimal Synthesis =
Priced Timed Games



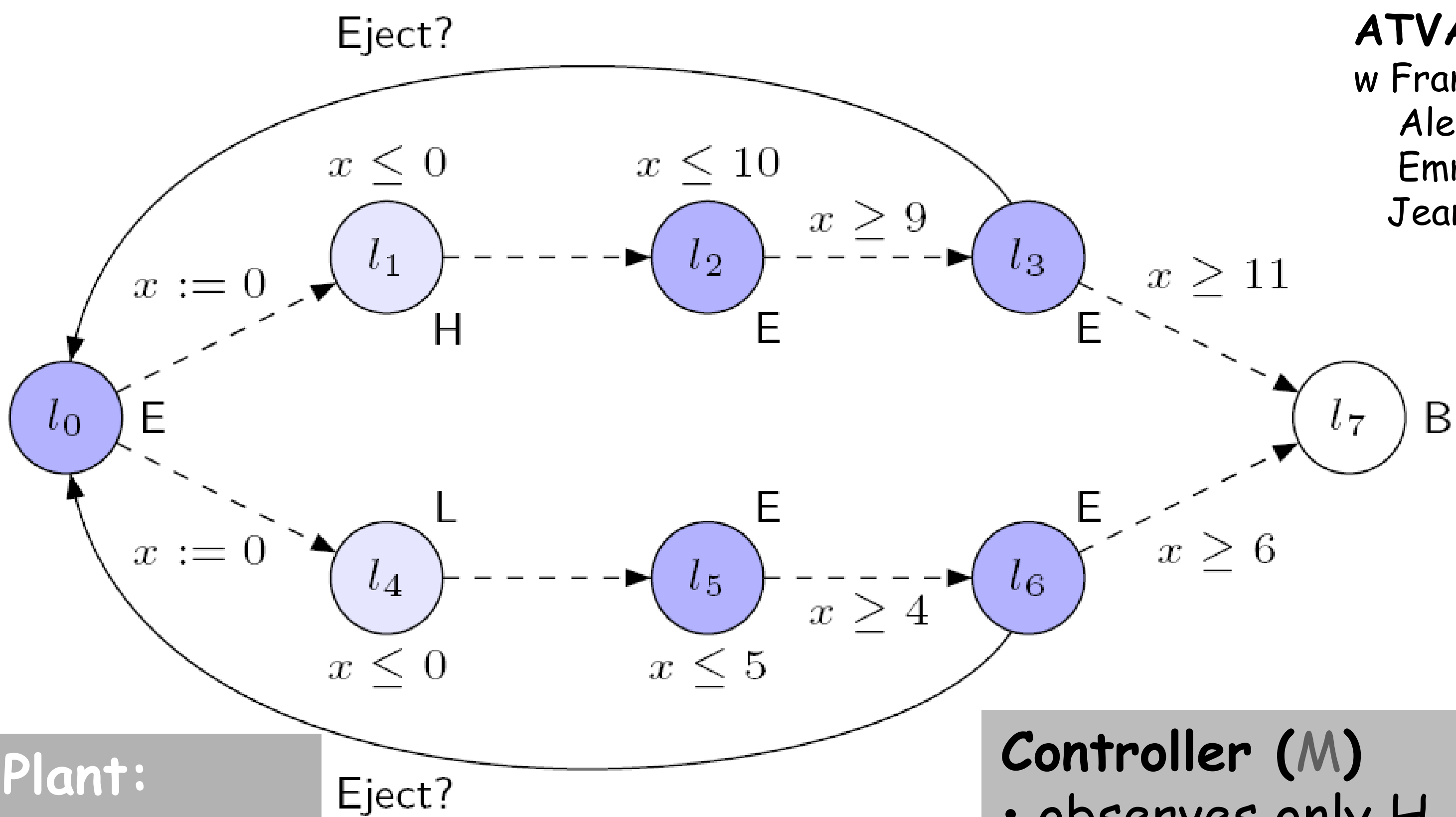
Optimal
winning strategies ??

Priced Timed Games

- Price Optimal Control (reachability):
  - Acyclic PTA [LTMM02]
  - Bounded length [ABM04]
  - Strong non-zeno cost-behaviour [BCFL04]
  - Undecidable with 3 clocks or more [BBR05, BBM06]
  - Decidable for PTGs with 1 clock [BLMR06]
- Priced Timed Safety Games
  - Conjectured to be undecidable in general.

Timed Games w Partial Observability

ATVA07
w Franck Cassez,
Alexandre David,
Emmanuel Fleury,
Jean-Francois Raskin



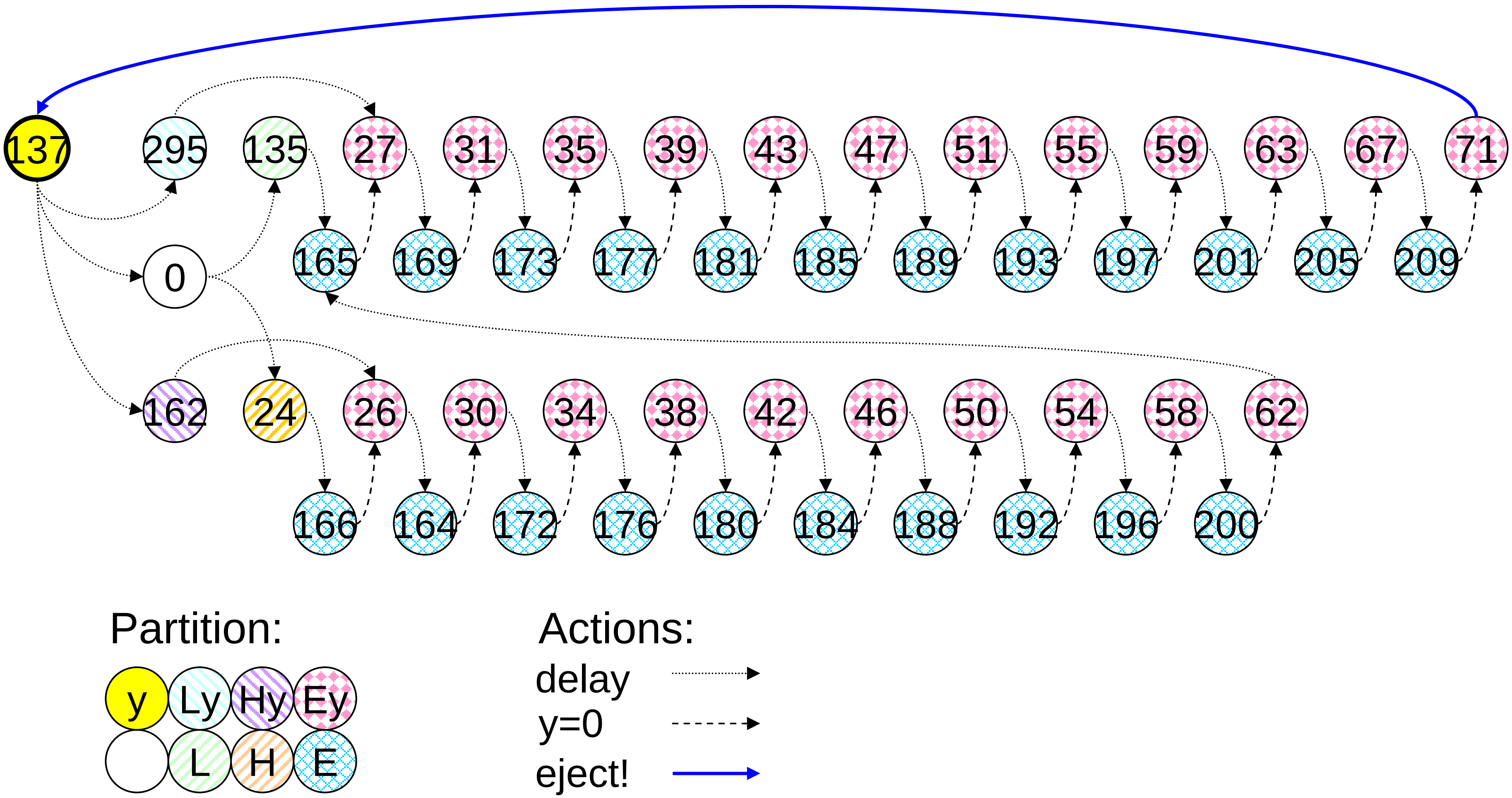
Plant:
Heavy & Light Bricks

Objective:
A[] ¬ B

Controller (M)
- observes only H, L, E or B
- has own clock y
- can test (y < M) or (y ≥ M)
- can always reset y

+ MEMORY

Memory-full Strategy



Formalizing the ARTS MPSoC Model in UPPAAL

Aske Brekling, Jens Ellebæk,
Kristian S. Knudsen, Jan Madsen,
Michael R. Hansen, Jacob Illum Rasmussen

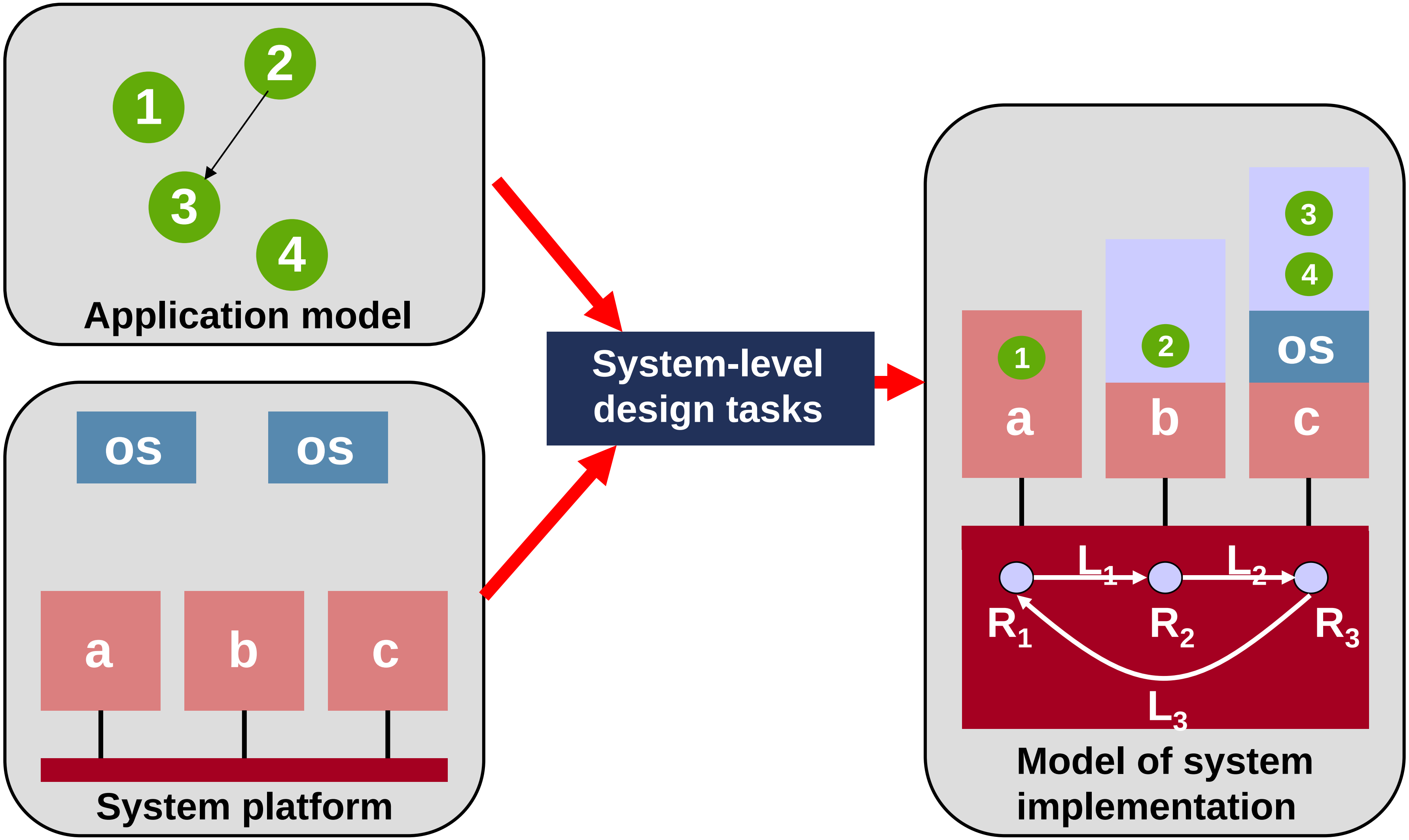
Embedded Systems Engineering Group

Informatics and Mathematical Modeling

Technical University of Denmark

DaNES

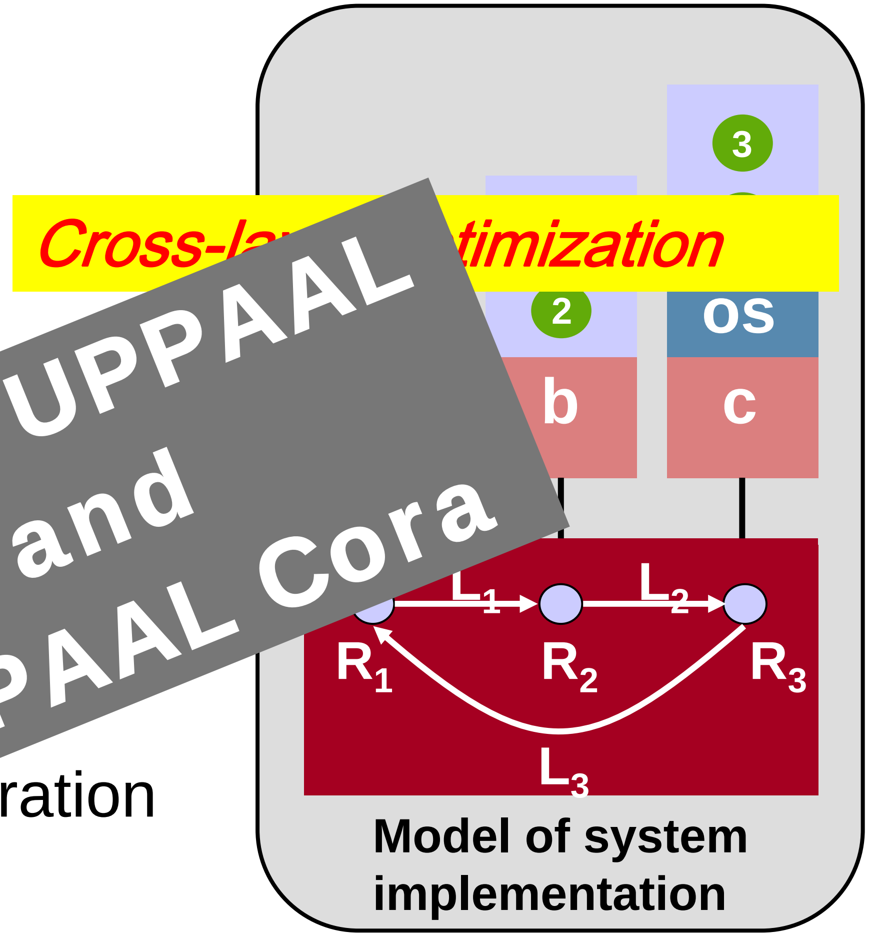
Motivation



2
1
3
4
Application model
OS
OS
a
b
c
System platform
System-level
design tasks
3
4
OS
2
1
a
b
c
L1
L2
R1
R2
R3
L3
Model of system
implementation

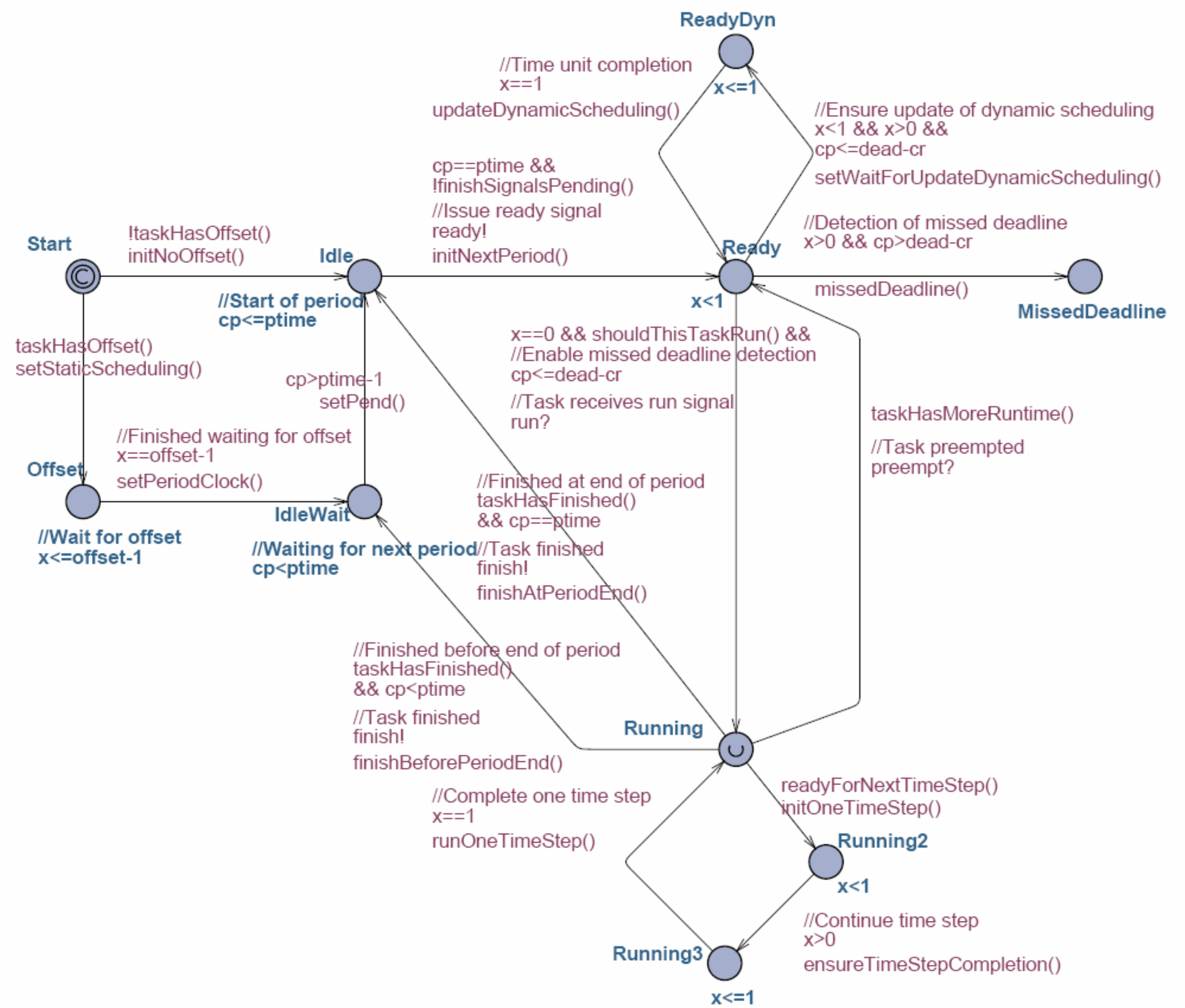
ARTS objectives

- System-level modeling framework
- Bridging,
  - Application
  - RTOS
  - Execution platform
    - Processing elemen
    - NoC
- Supporting
  - System-level an
  - Early design spac exploration



Using UPPAAL and UPPAAL Cora

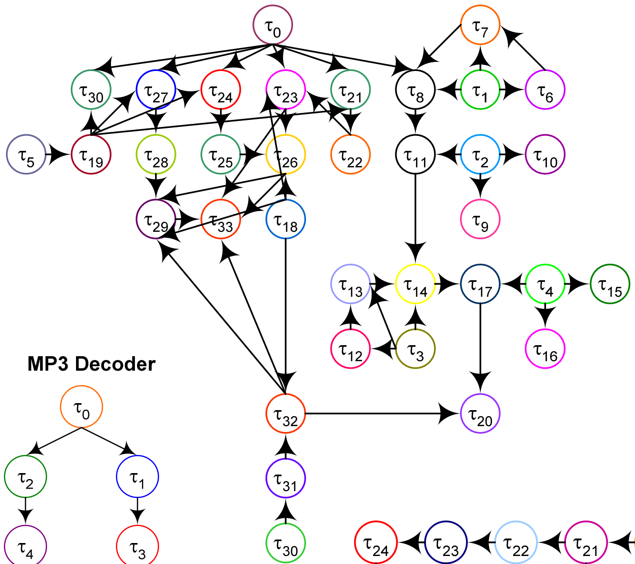
Timed Automata for a task



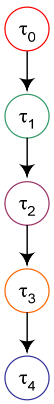
Handling realistic applications?

Smart phone:

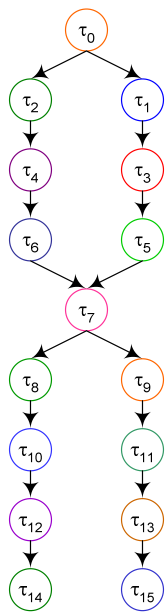
GSM Decoder



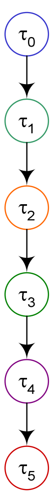
JPEG Encoder



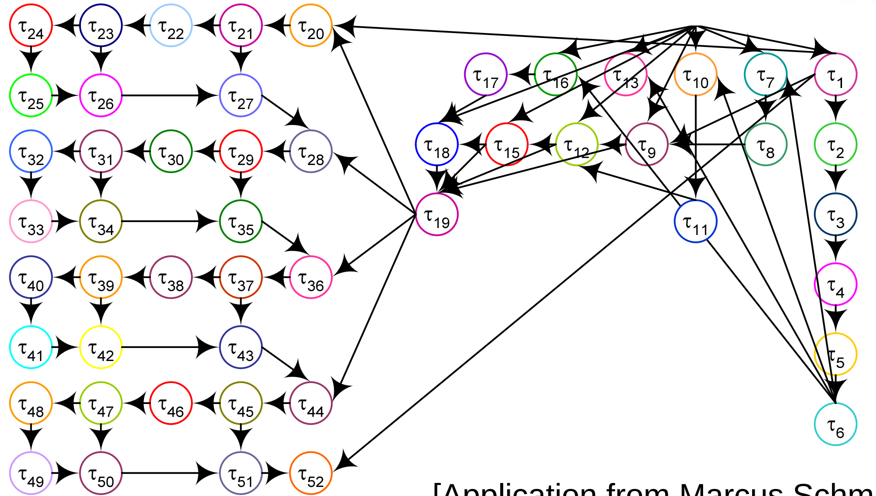
MP3 Decoder



JPEG Decoder

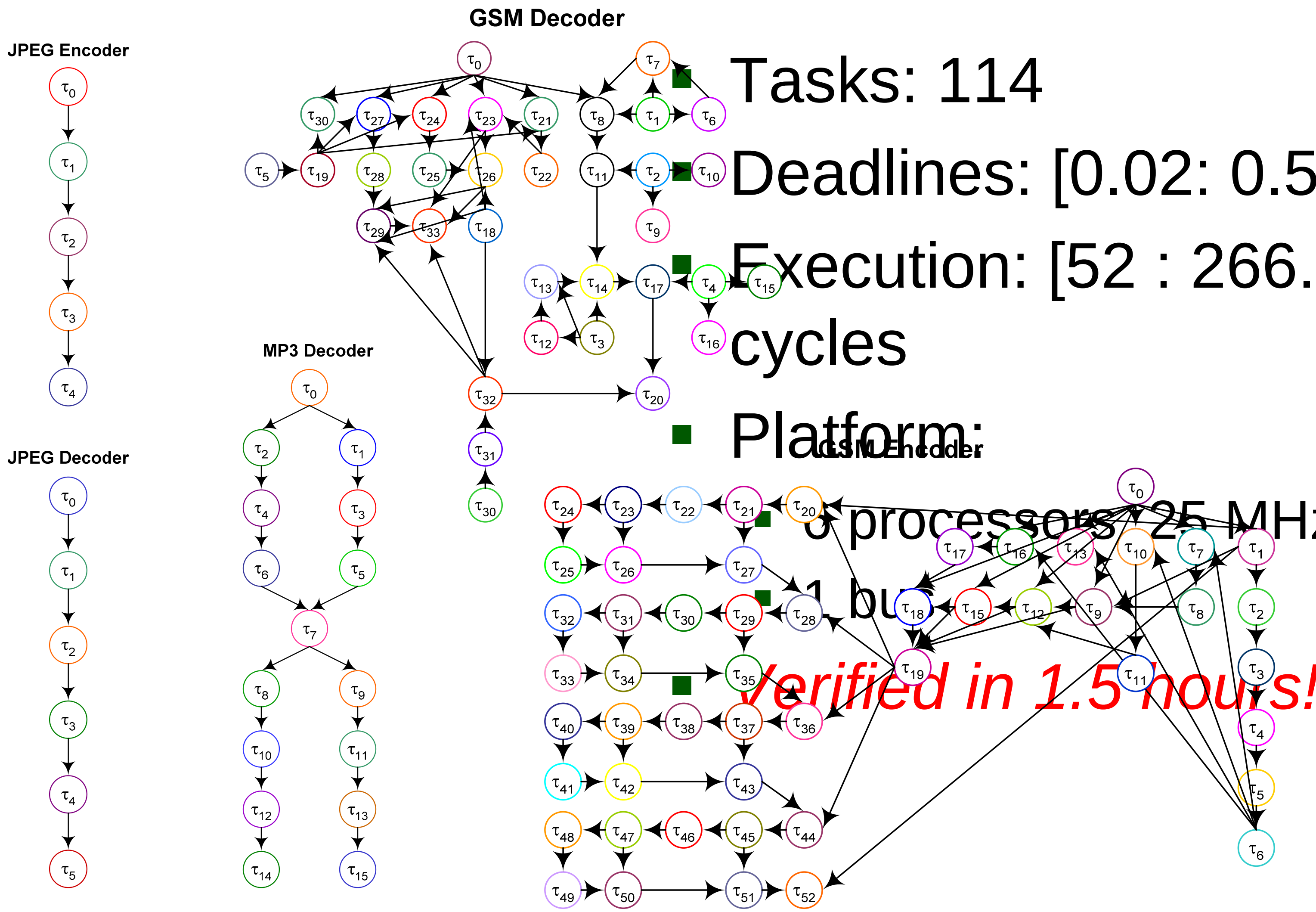


GSM Encoder



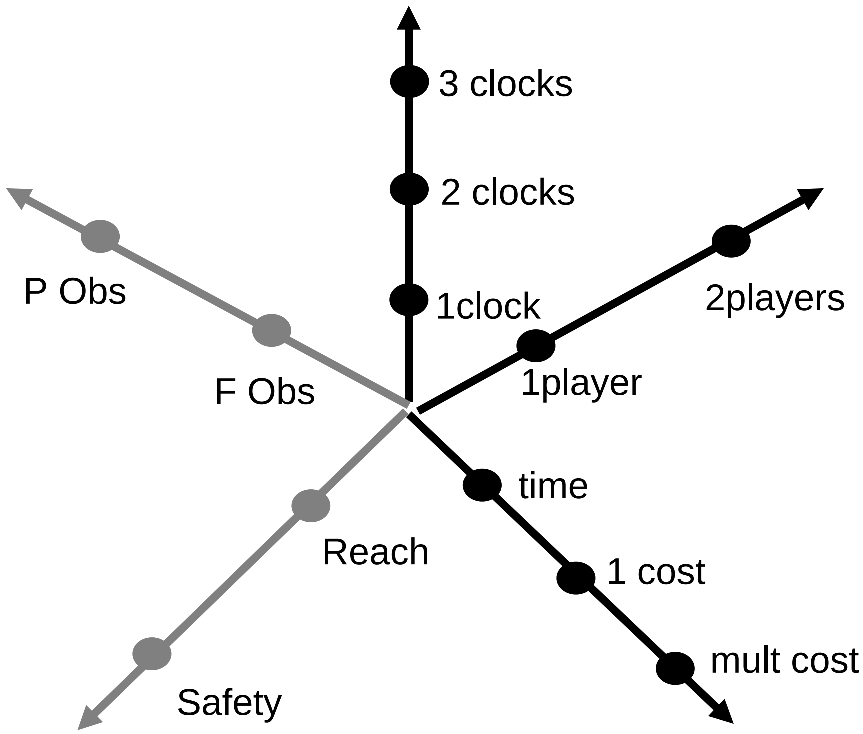
[Application from Marcus Schmitz, TU Linkoping]

Smart phone

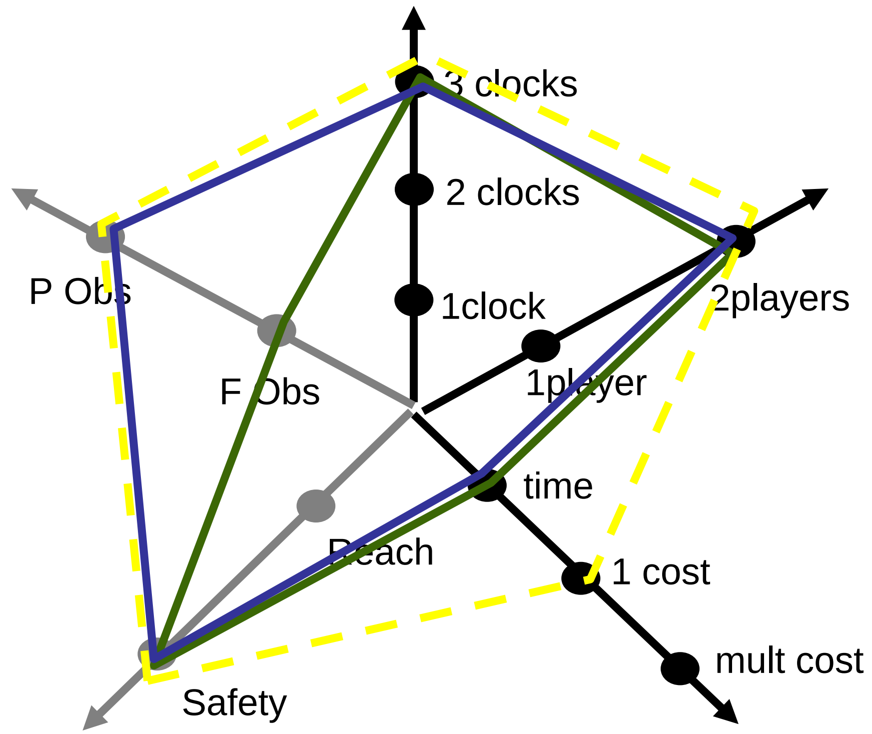


- Tasks: 114
- Deadlines: [0.02: 0.5] sec
- Execution: [52 : 266.687] cycles
- Platform:
  - 6 processors, 25 MHz
  - 1 bus
- *Verified in 1.5 hours!*

Optimal Combinations

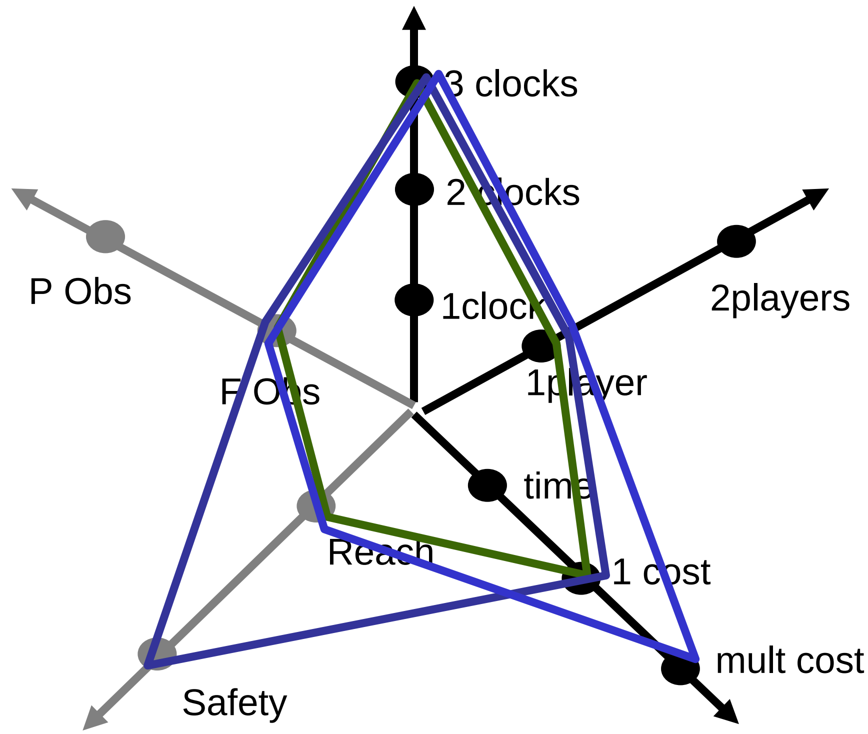


Optimal Combinations



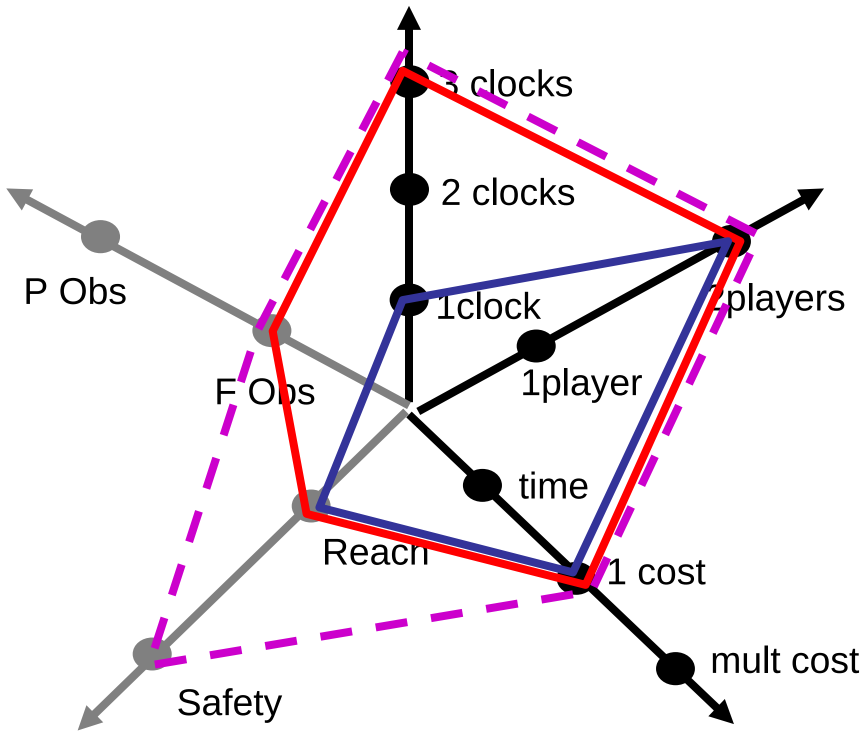
TIGA

Optimal Combinations

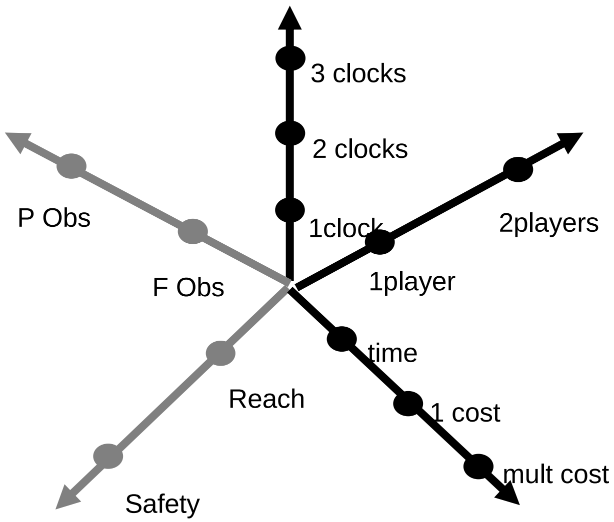


CORA

Optimal Combinations



Conclusion



- Identification of all Pareto optimal combinations!
  - Safety for PTG?
  - Reachability for 2PTG?
  - Safety for MPTA?
  - Safety for 1PTG?
- Efficient realizations
  - TG w PO?
  - Safety for PTA?
  - Reachability for 1PTG?
- Dealing with undecidability.

Thanks for your attention!

Please do not hesitate to contact me:

kgl@cs.aau.dk