Model-Based Schedulability Analysis

Gregor Gössler

February 26, 2002

Outline

1. Introduction: the limits of analytical schedulability analysis
2. Modeling Real-Time Systems
   - tasks
   - scheduling algorithms
3. Model-Based Schedulability Analysis
4. Conclusion

Reminder: analytical schedulability analysis

A scheduling policy encompasses 3 elements:

1. assumption about the problem: periodicity of processes, deadlines, use of resources, . . .
2. schedulability test
3. scheduling algorithm

Standard engineering practice.

Example: preemptive EDF

Preemptive EDF scheduling:

1. assumption: independent tasks of period T_i and execution time E_i sharing one processor and no other resource, rel. deadline = period
2. schedulable iff $\sum_i \frac{E_i}{T_i} \leqslant 1$
3. scheduling algorithm: earliest deadline first

And if we want to schedule tasks that share another, non-preemptable resource?

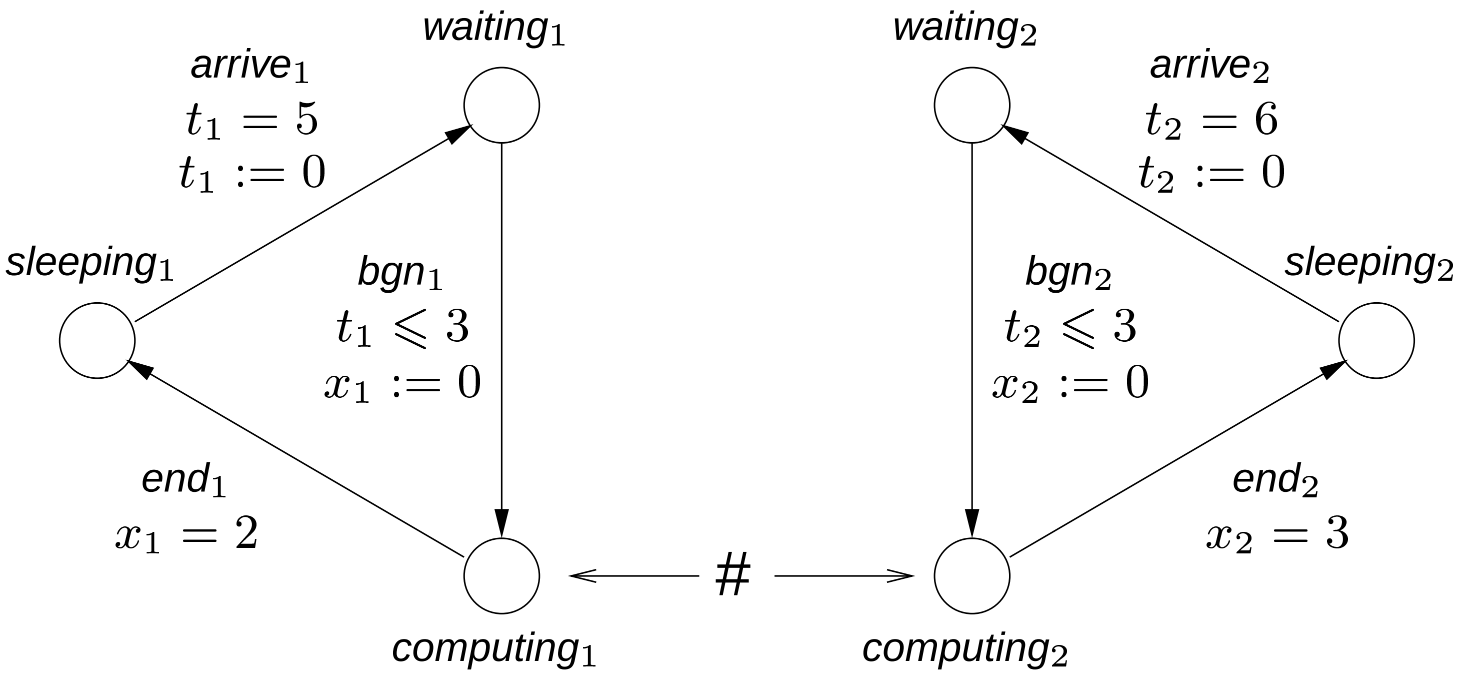
- generalize the scheduling policy (relax assumptions), or
- use model-based schedulability analysis.

Model-based schedulability analysis

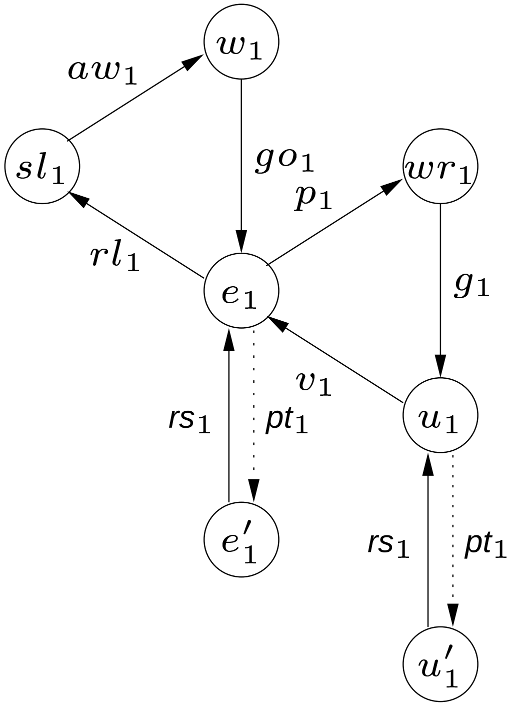
Idea:

- Model the problem (tasks and scheduling policy) in a general mathematical framework, e.g. timed automata with priorities.
- Formulate schedulability as a property of the model.
- Check if the model satisfies the property.

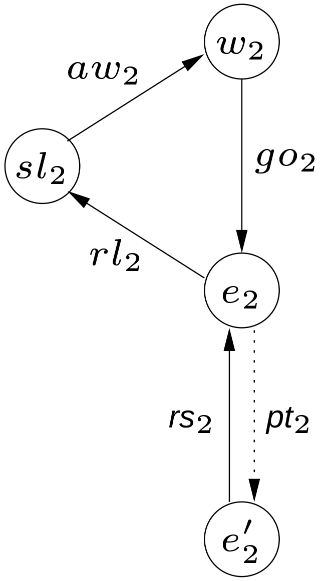
Example: two periodic processes



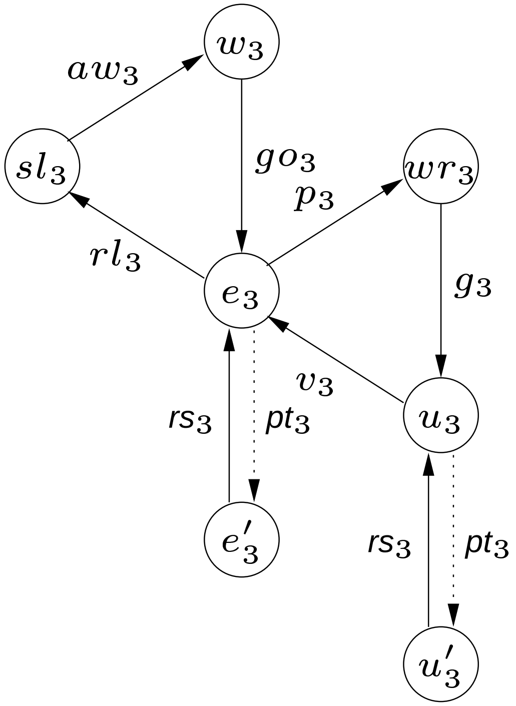
Example: three processes sharing two resources



TS_1^{PMTN}

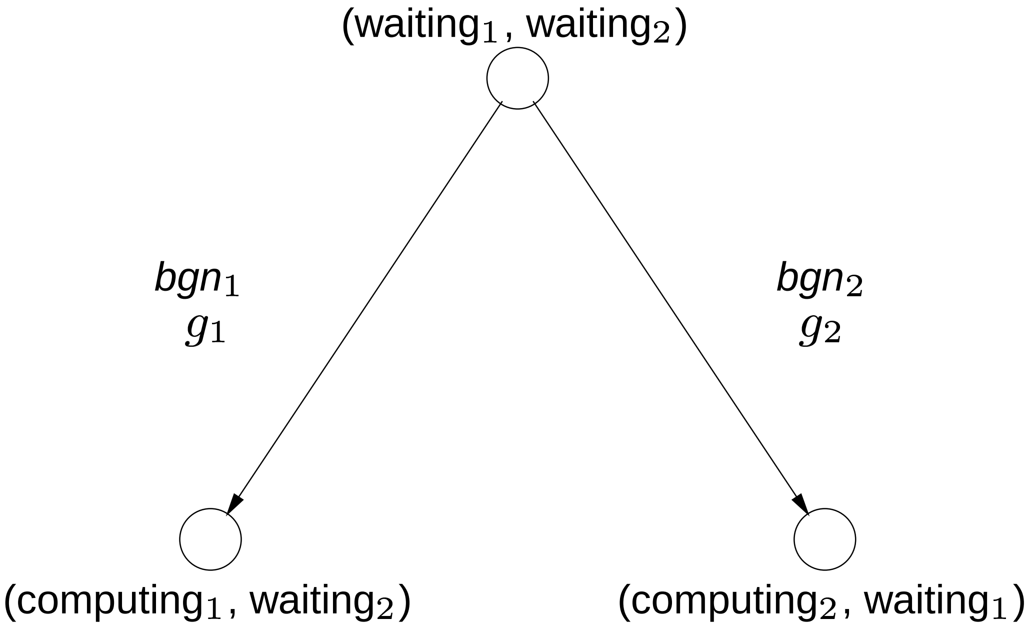


TS_2^{PMTN}

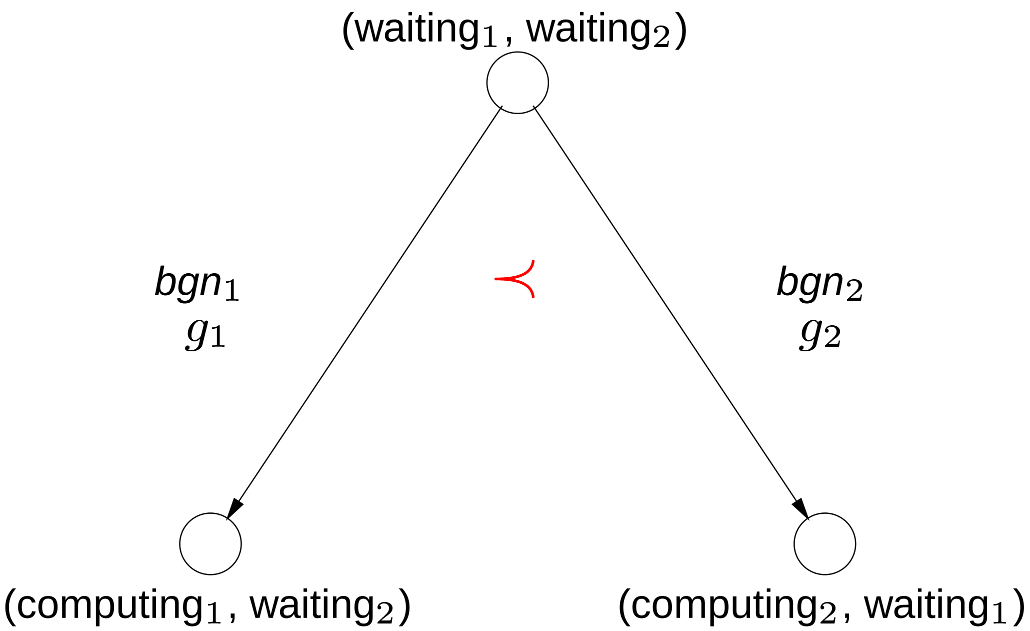


TS_3^{PMTN}

Dynamic priorities to model scheduling algorithms


(waiting$_1$, waiting$_2$)
bgn$_1$
g_1
bgn$_2$
g_2
(computing$_1$, waiting$_2$)
(computing$_2$, waiting$_1$)

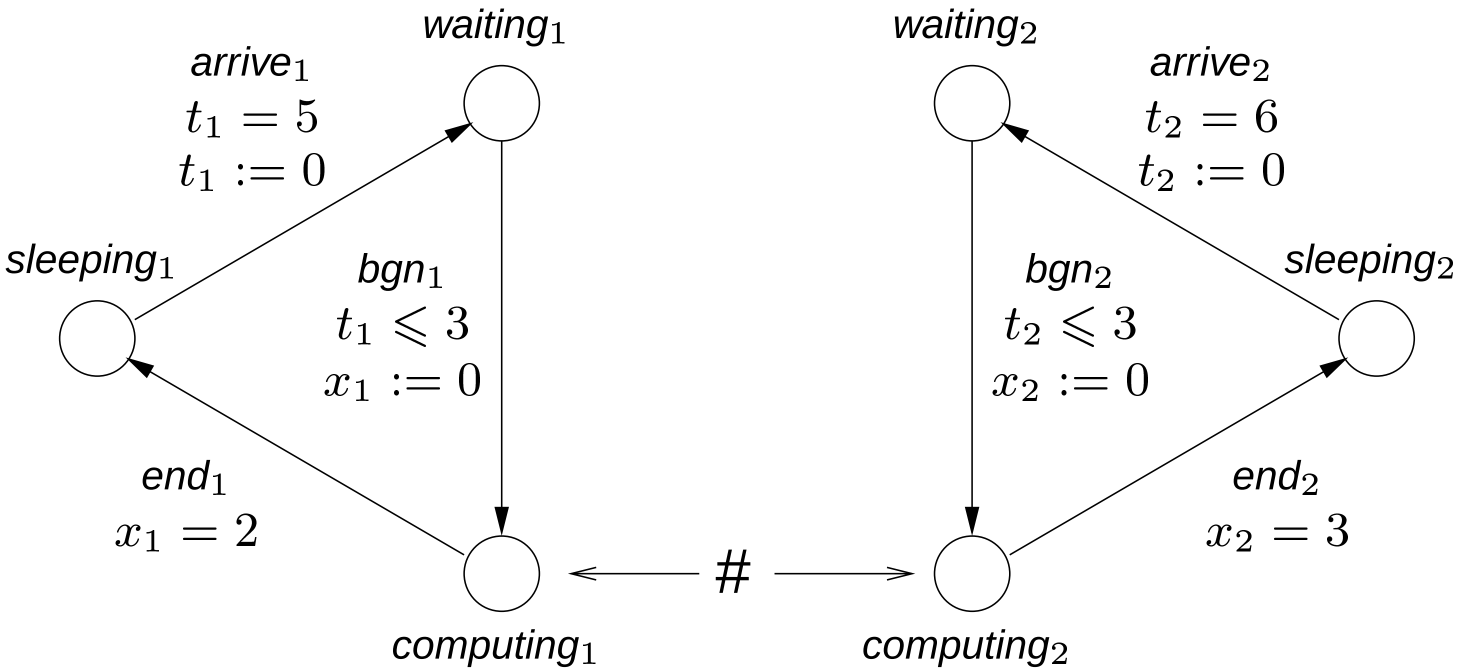
Dynamic priorities to model scheduling algorithms


(waiting$_1$, waiting$_2$)
bgn_1
g_1
$\prec$
bgn_2
g_2
(computing$_1$, waiting$_2$)
(computing$_2$, waiting$_1$)

priority function	guard of bgn_1 restricted to
$true \Rightarrow bgn_1 \prec bgn_2$	$g_1 \;\wedge\; \neg g_2$
$\mathcal{C} \Rightarrow bgn_1 \prec bgn_2$	$g_1 \;\wedge\; \neg(\mathcal{C} \wedge g_2)$

Modeling Scheduling Algorithms

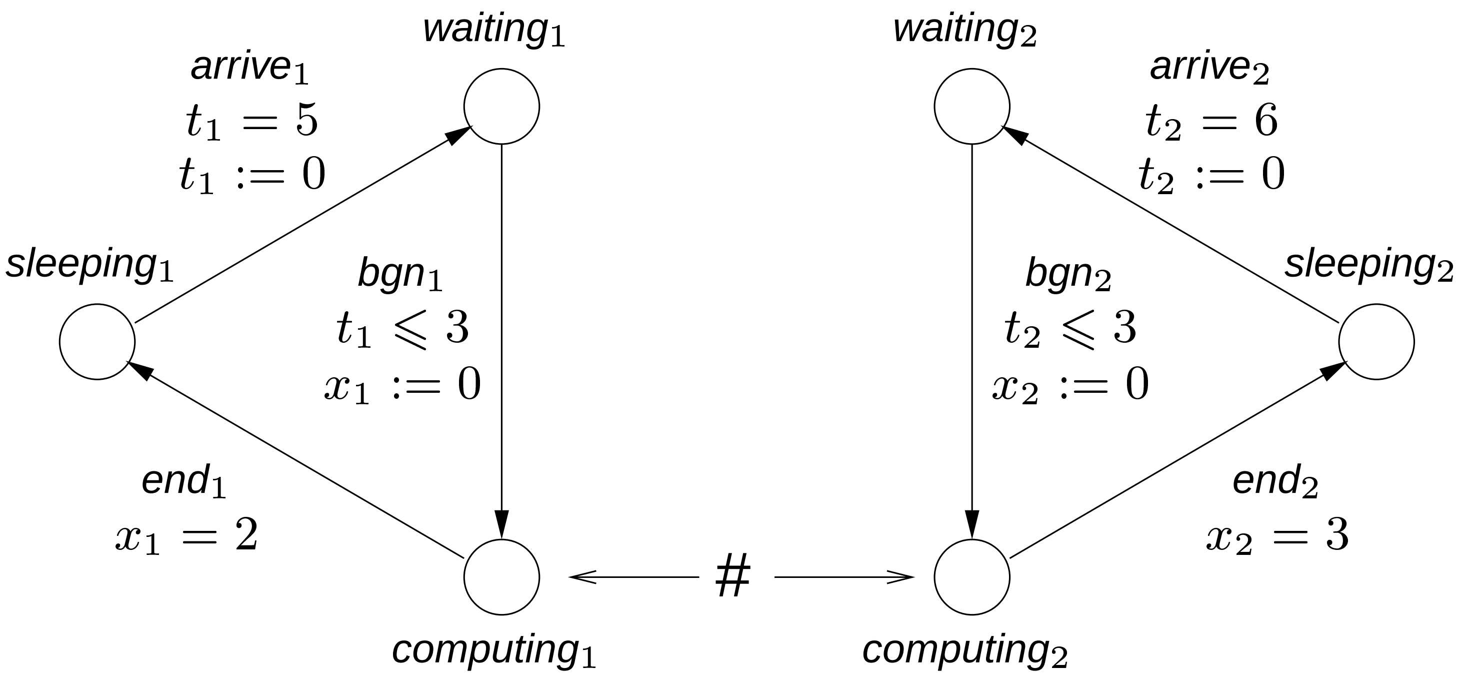
Example: RMS



RMS scheduling: *true* $\Rightarrow$ *bgn*$_2 \prec$ *bgn*$_1$

Modeling Scheduling Algorithms

Example: EDF

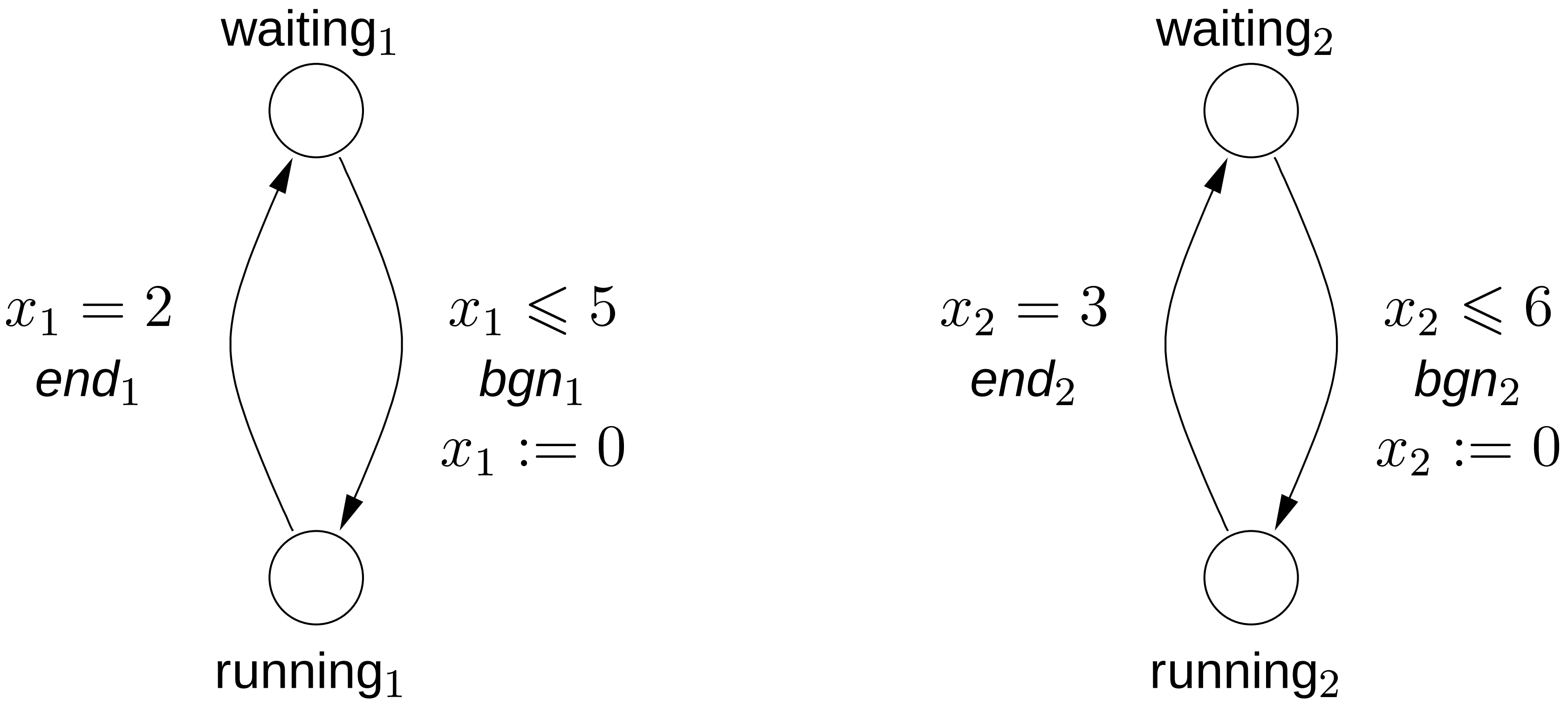


EDF scheduling: $5 - t_1 > 6 - t_2 \;\Rightarrow\;$ *bgn*$_1 \prec$ *bgn*$_2$

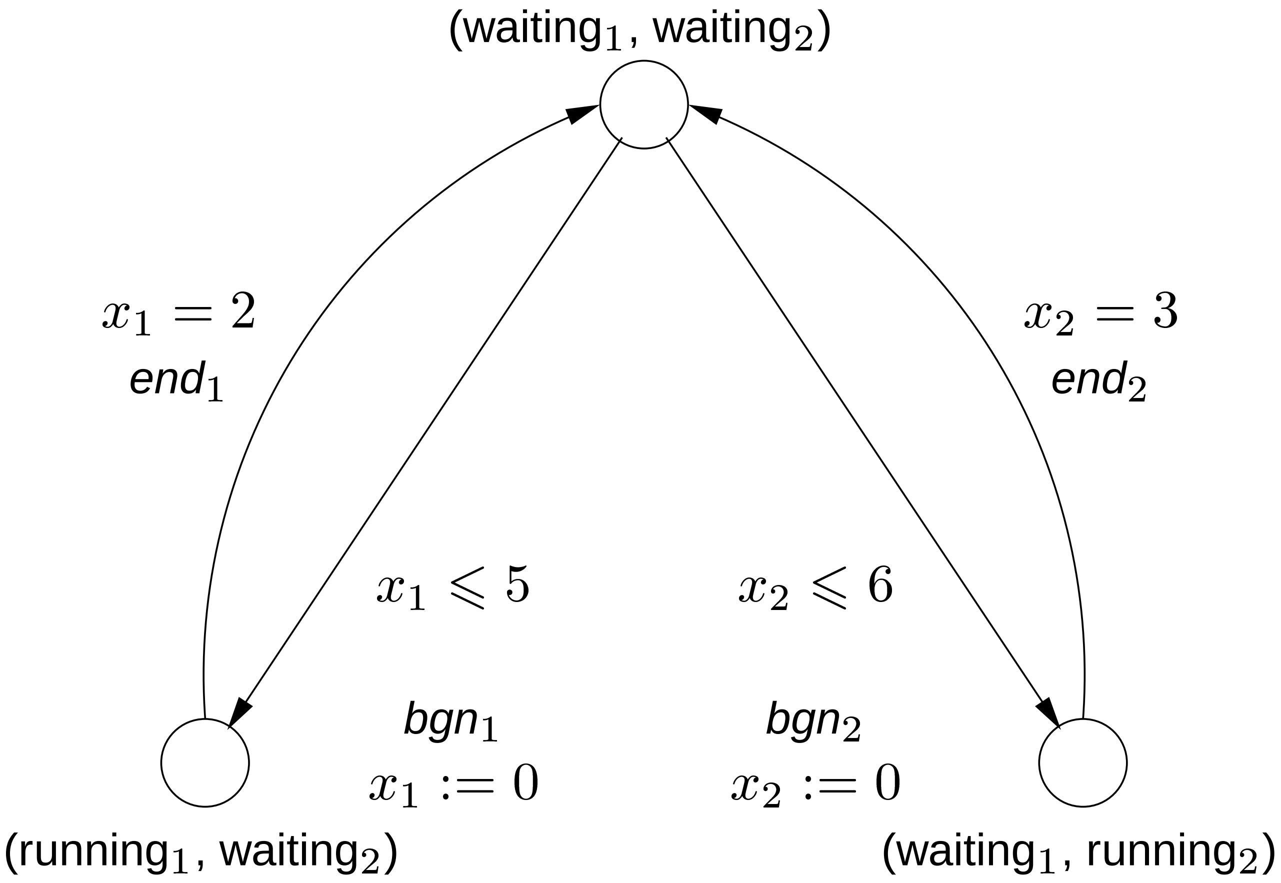
$5 - t_1 < 6 - t_2 \;\Rightarrow\;$ *bgn*$_2 \prec$ *bgn*$_1$

Model-Based Schedulability Analysis

A very simple example



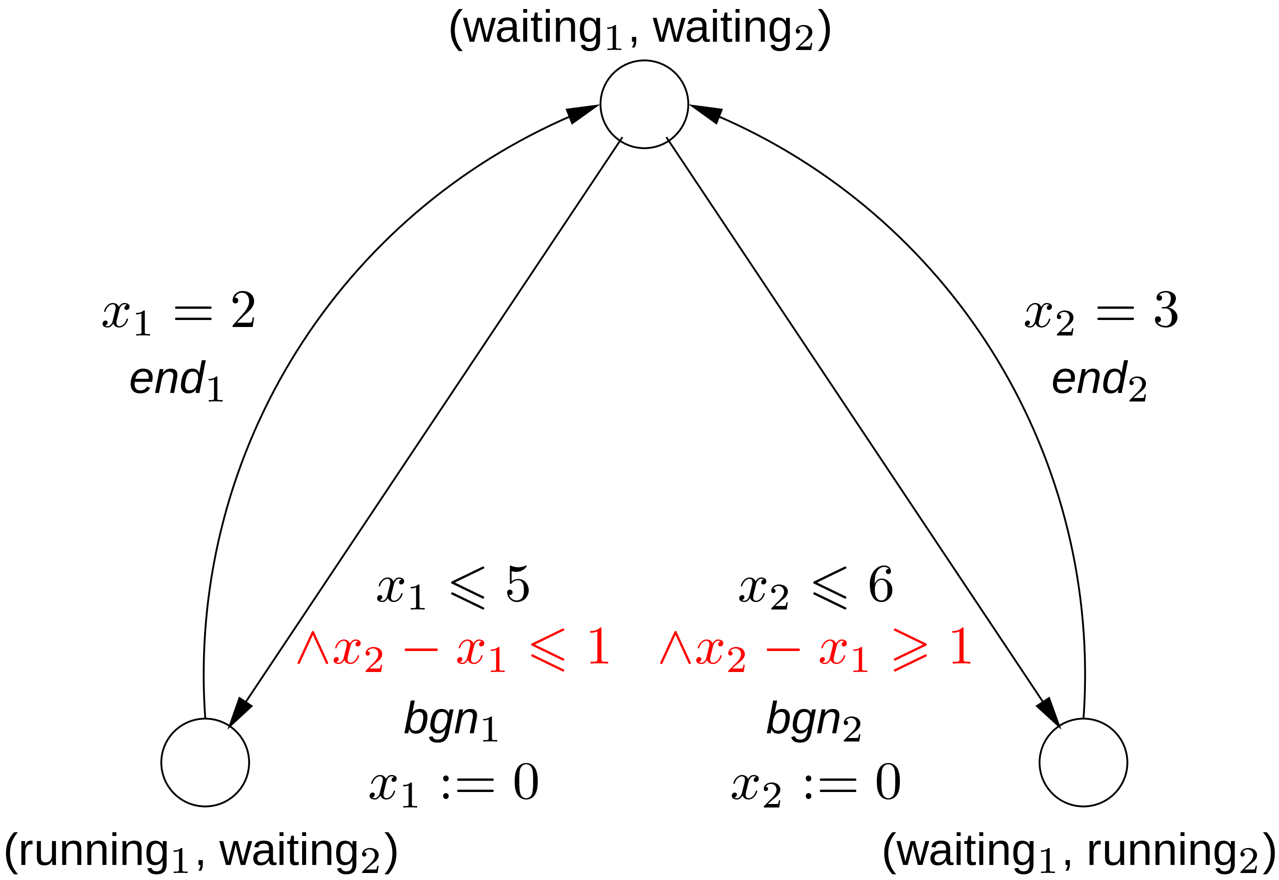
Constructing the product automaton



EDF scheduling: $5 - x_1 > 6 - x_2 \;\Rightarrow\; \textit{bgn}_1 \prec \textit{bgn}_2$

$5 - x_1 < 6 - x_2 \;\Rightarrow\; \textit{bgn}_2 \prec \textit{bgn}_1$

Applying the scheduling policy


(waiting$_1$, waiting$_2$)
$x_1 = 2$
end$_1$
$x_2 = 3$
end$_2$
$x_1 \leqslant 5$
$\wedge x_2 - x_1 \leqslant 1$
bgn$_1$
$x_1 := 0$
$x_2 \leqslant 6$
$\wedge x_2 - x_1 \geqslant 1$
bgn$_2$
$x_2 := 0$
(running$_1$, waiting$_2$)
(waiting$_1$, running$_2$)

EDF scheduling: $5 - x_1 > 6 - x_2 \;\Rightarrow\; bgn_1 \prec bgn_2$

$5 - x_1 < 6 - x_2 \;\Rightarrow\; bgn_2 \prec bgn_1$

Computing the fixpoint

A state of the automaton: $(s, v) \in S \times V$.

Correctly scheduled states: $X \; \subseteq \; S \times (x_1 \leqslant 5 \; \wedge x_2 \leqslant 6)$ s.t.

$$X \;=\; X \;\cap\; pre(X)$$

where

$$pre(X) = \{(s, v) \mid \exists d \geqslant 0 \; \exists \text{ transition } t \; . \; (s, v + d) \xrightarrow{t} (s', v') \in X\}$$

Computing X:

$$\begin{aligned} X_0 &= S \times (x_1 \leqslant 5 \; \wedge x_2 \leqslant 6) \\ X_1 &= X_0 \;\cap\; pre(X_0) \\ &\;\vdots \\ X_\infty &= \bigcap_{i=0}^{\infty} pre^i(X_0) \end{aligned}$$

Example: Priority Ceiling Protocol

```
SCHEDULER pcp()
    forall Pi: process
    forall Pj: process
        BLOCKING (Pi, Pj) :=
            exists r1: nonpreemptable
            exists r2: nonpreemptable
                USING (Pi, r1) and WAITING (Pj, r2)
                and CEILING (r1) >= PRIO (Pj)

    PRIORITIES
        forall Pi: process
        forall Pj: process
            Pj <0 Pi  if
                forall Pk: process
                    PRIO (Pi) > PRIO (Pj) and
                    (PRIO(Pk) < PRIO(Pi) or not BLOCKING (Pj, Pk))

        forall Pi: process
        forall Pj: process
            Pj <0 Pi  if
                exists Pk: process
                    PRIO (Pi) < PRIO (Pj) and
                    PRIO (Pk) > PRIO (Pj) and BLOCKING (Pi, Pk)
END pcp
```

Conclusion

Overview:

	analytical	model-based
1.	assumptions	only restriction: expressivity of formalism
2.	schedulability test	algorithm to check deadlock-freedom
3.	scheduling algorithm	

- trade-off generality/complexity
- integration of model-based and analytical schedulability analysis?

Related Work

C. L. Liu and J. W. Layland, *Scheduling algorithms for multiprogramming in a hard-real-time environment*, JACM 1973.

T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine, *Symbolic model checking for real-time systems*, LICS’92.

C. Daws, A. Olivero, S. Tripakis, and S. Yovine, *The tool* Kronos, HS’96.

K. G. Larsen, P. Pettersson, and W. Yi, Uppaal *in a nutshell*, Software Tools for Technology Transfer, 1997.

T.A. Henzinger, P.H. Ho, and H. Wong-Toi, HyTech*: A model checker for hybrid systems*, Software Tools for Technology Transfer, 1997.

H. Ben-Abdallah *et al.*, *A process algebraic approach to the schedulability analysis of real-time systems*, JRTS, 1998.

K. Altisen, G. Gössler, and J. Sifakis, *Scheduler modeling based on the controller synthesis paradigm*, JRTS, 2002.