Synthesis problem

Given a timed system TS and constraint K, we want to compute a non-empty control invariant K' of TS which implies K, that is K'$\Rightarrow$ K and TS/K'$\models$ inv(K').

Timed system TS represents the processes to be scheduled and their timing constraints.

Constraint K represents scheduling requirements. In general, $K = K_{algo} \wedge K_{sched}$, where K_{algo} is a constraint characterising a particular scheduling algorithm, and K_{sched} expresses the timing requirements.

The resulting system TS/K' describes the scheduled system.

Scheduling constraints

Correctness = "deadlock-freedom" + "live-lock freedom".

TS$= \{P_1, \ldots P_n\}$ is a timed system of *structurally time-lock free* processes.

We want each P_i to be deadlock-free, i.e. it will always eventually execute some action (fairness of the scheduling algorithm).

$\mathrm{K}_{\mathrm{sched}} = \bigwedge_{P_i} \mathrm{K}_{\mathrm{sched_i}}$, where

$\mathrm{K}_{\mathrm{sched_i}} = \bigvee_{s \in S_i} s \wedge (\bigvee_{(s,a,s') \in T_i} \Diamond g_a)$

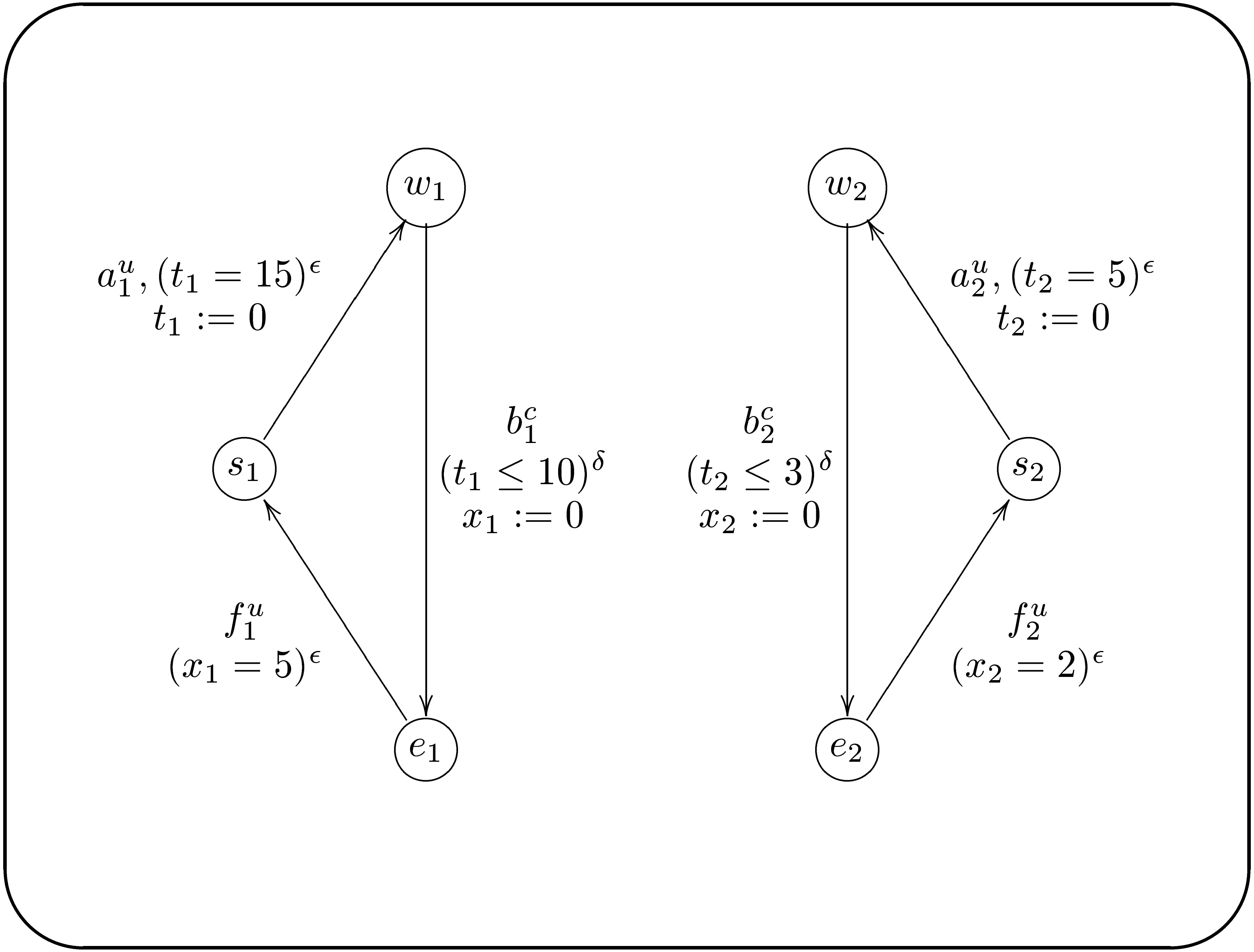
Notation: $(\Diamond C)(x)$ stands for $\exists t \geq 0.\ C(x+t)$

Example:

$$s \xrightarrow{a,\ x\leq 5} s' \xrightarrow{b,\ x\leq 4} s''$$

In s, we may wait until $x = 5$ and then proceed via a to s'. However, that means deadlock at s' as the guard $x \leq 4$ will never become true. If we replace additional constraint on s, $K = s' \wedge x \leq 4$, then we get a deadlock-free system

$$s \xrightarrow{a,\ x\leq 5 \wedge x\leq 4} s' \xrightarrow{b,\ x\leq 4} s''$$


w_1
$a_1^u, (t_1 = 15)^\epsilon$
$t_1 := 0$
s_1
b_1^c
$(t_1 \leq 10)^\delta$
$x_1 := 0$
f_1^u
$(x_1 = 5)^\epsilon$
e_1
w_2
$a_2^u, (t_2 = 5)^\epsilon$
$t_2 := 0$
b_2^c
$(t_2 \leq 3)^\delta$
$x_2 := 0$
s_2
f_2^u
$(x_2 = 2)^\epsilon$
e_2

Schedulability constraint for the composition of n periodic processes

$\mathrm{K}_{\mathrm{sched}} = \bigwedge_{P_i}(s_i \wedge \Diamond g_{a_i} \vee e_i \wedge \Diamond g_{f_i} \vee w_i \wedge \Diamond g_{b_i})$.

The previous example will yield

$$\mathrm{K}_{\mathrm{sched}} = \begin{array}{r} s_1 \wedge t_1 \leq 15 \\ \vee\ e_1 \wedge x_1 \leq 15 \\ \vee\ w_1 \wedge t_1 \leq 10 \end{array} \bigwedge \begin{array}{r} s_2 \wedge t+2 \leq 5 \\ \vee\ e_2 \wedge x_1 \leq 2 \\ \vee\ w_2 \wedge t_2 \leq 3 \end{array}$$

A maximal control invariant implying $\mathrm{K}_{\mathrm{sched}}$ can be computed by tool KRONOS.

Composability

Incremental methodology for computing a system restricted by a sequence of constraints.

Example: we have a timed system TS and constraints K_1, K_2. Let's assume that TS/K_2 satisfies K_2, and now we want to restrict it further to satisfy K_1. However, $TS/(K_1 \wedge K_2)$ does not necessarily need to imply K_1.

Def: A constraint K_1 is a *composable invariant* of TS if for all constraints K_2, $TS/((K_1 \wedge K_2) \models inv(K_1)$.

This notion allows to apply restriction sequentially, to build more precise approximations of the correct scheduler.

Timed systems with priorities

- a natural way of specifying scheduling algorithms

Def: *Priority order* $\prec$ is a binary relation on actions that is a partial order on actions.

Def: *Timed system with priorities* (TS,pr) is a timed system with a finite set of priority rules pr$= \{(C^i, \prec^i)\}$, where C^i are clock constraints that determine which priority order $\prec^i$ applies.

Transformation of (TS,pr) into TS' - guards g_j of TS are replaced by g'_j:

$$g'_j = g_j \wedge \bigwedge_{(C,\prec)\in pr} (\neg C \vee \bigwedge_{a_j \prec a_i} \neg \Diamond g_i)$$

Example: action a_1 - deadline 5, execution time t_1; action a_2 - deadline 10, execution time t_2;

$$\begin{tikzcd} & s_1 \\ s \arrow[ur, "{a_1, tt}"] \arrow[r, "{a_2, tt}"'] & s_2 \end{tikzcd}$$

$$\mathrm{pr} = \{(5 - t_1 < 10 - t_2, \{a_2 \prec a_1\}), (10 - t_1 < 5 - t_2, \{a_1 \prec a_2\})\}$$

$$(s, 4, 4) \xrightarrow{\quad a_1 \quad} (s_1, 4, 4) \qquad\qquad a_2 \prec a_1$$

$$(s, 3, 9) \xrightarrow{\quad a_2 \quad} (s_1, 3, 9) \qquad\qquad a_1 \prec a_2$$

Example: *Earliest deadline first (EDF)* policy on a system of processes $\{P_1, \ldots, P_n\}$.

Process P_i - deadline D_i, execution time t_i; action b_i stands for begin execution on CPU;

EDF algorithm grants CPU to the waiting process P_i that is closest to its relative deadline, i.e. has minimum difference between D_i and t_i.

EDF can be specified (partially) as follows:

$$pr = \{D_i - t_i < D_j - t_j, \{b_j \prec b_i\}\}_{i \neq j}$$

Unified approach using constraints - a timed system with priorities transferred into an equivalent timed system with constraints:

given (TS,pr), where pr$= \{(C^i, \prec^i)\}$, we construct a constraint K$_{pr}$:

$$\bigwedge_{s \in S} \bigwedge_{(C,\prec)\in pr} \bigwedge_{a_j \prec a_i} (s_j \wedge (\neg C \vee \neg \Diamond^s g_i)$$

where for a given s, a_i is a transition with guard g_i leading from s to s_i.

This is a composable constraint, and, (TS,pr) is strongly equivalent to TS/K.