How the Timed Automaton Lost its Tail (and Clocks)

Oded Maler

Joint work with Jean-Francois Kempf and Marius Bozga

CNRS - VERIMAG
Grenoble, France

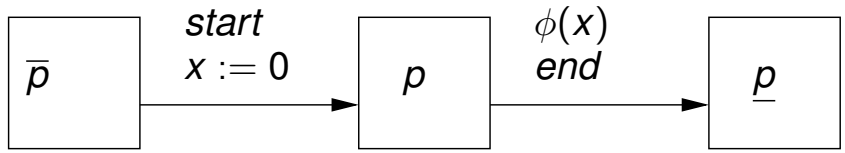
FORMATS
Aalborg 2011

Returning to the Scene of the Crime

- I am happy to present this work in **Aalborg** where it started two years ago by discussions with **Kim Larsen**
- Initial goal was to do timing analysis by **statistical** methods on **duration probabilistic automata**
- But then we had some ideas to **compute** probabilities using **density transformers**, extensions of the **zone transformers** used in the verification of timed automata:
- OM, **Kim Larsen** and **Bruce Krogh**: *On Zone-Based Analysis of Duration Probabilistic Automata*, Infinity 2010
- Similar to **Vicario et al.** and **Alur and Bernadsky**
- The present **clock-free** work is a byproduct of trying to **implement** the ideas
- Let us start with an intuitive introduction to the context

Processes that Take Time

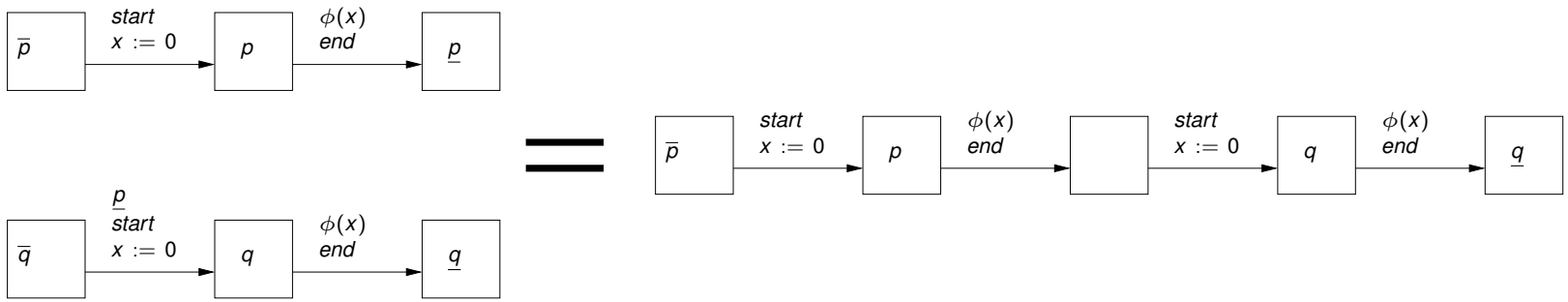
- Processes that take some time to conclude after having started, for example:
  - Propagation delay between *send* and *receive*
  - Execution time of a program
  - Duration of a step in a manufacturing process
- Mathematically they are simple timed automata:



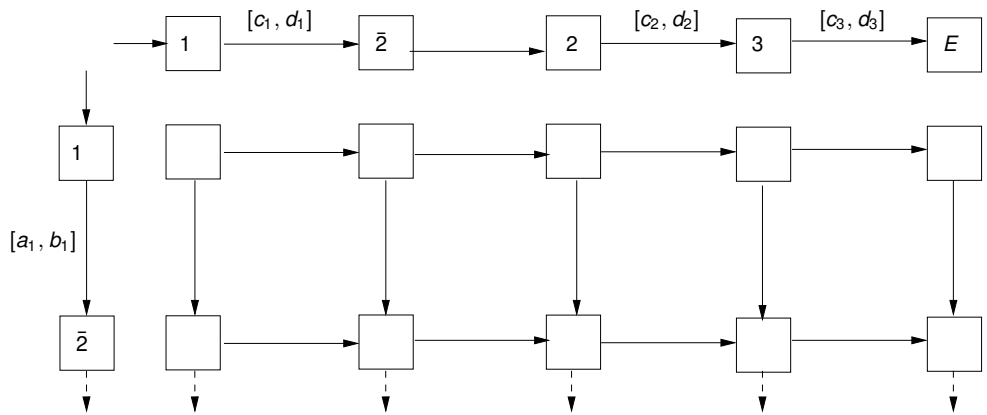
- A waiting state $\overline{p}$; a *start* transition which resets a clock x to measure time elapsed in active state p
- An *end* transition guarded by a temporal condition $\phi(x)$
- Condition ϕ can be **true** (no constraint), $x = d$ (deterministic), $x \in [a, b]$ (non-deterministic) or probabilistic

Composition

- Such processes can be combined:
- Sequentially to represent precedence relations between tasks, for example p precedes q:

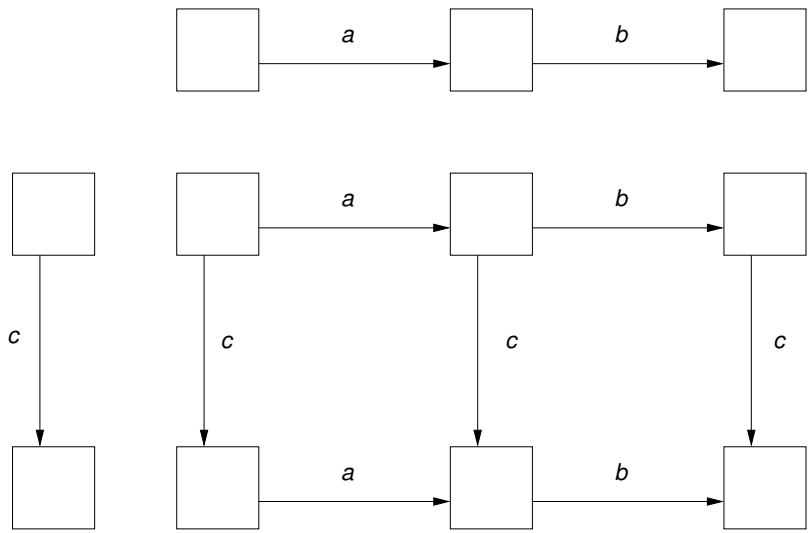


- In parallel to express partially-independent processes, sometimes competing with each other



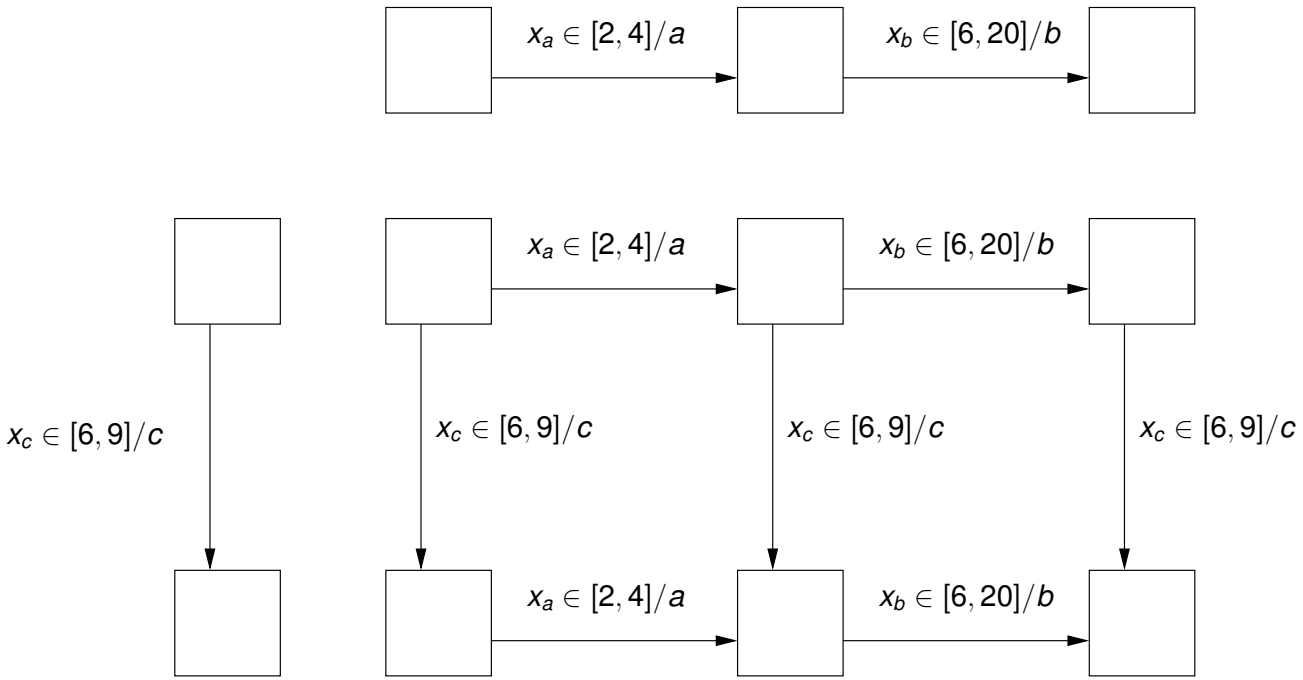
Levels of Abstraction: Untimed

- Untimed (asynchronous) approach:
- Each process may take between zero and infinity time
- Consequently **any** interleaving in $(a \cdot b)||c$ is possible



Levels of Abstraction: Timed

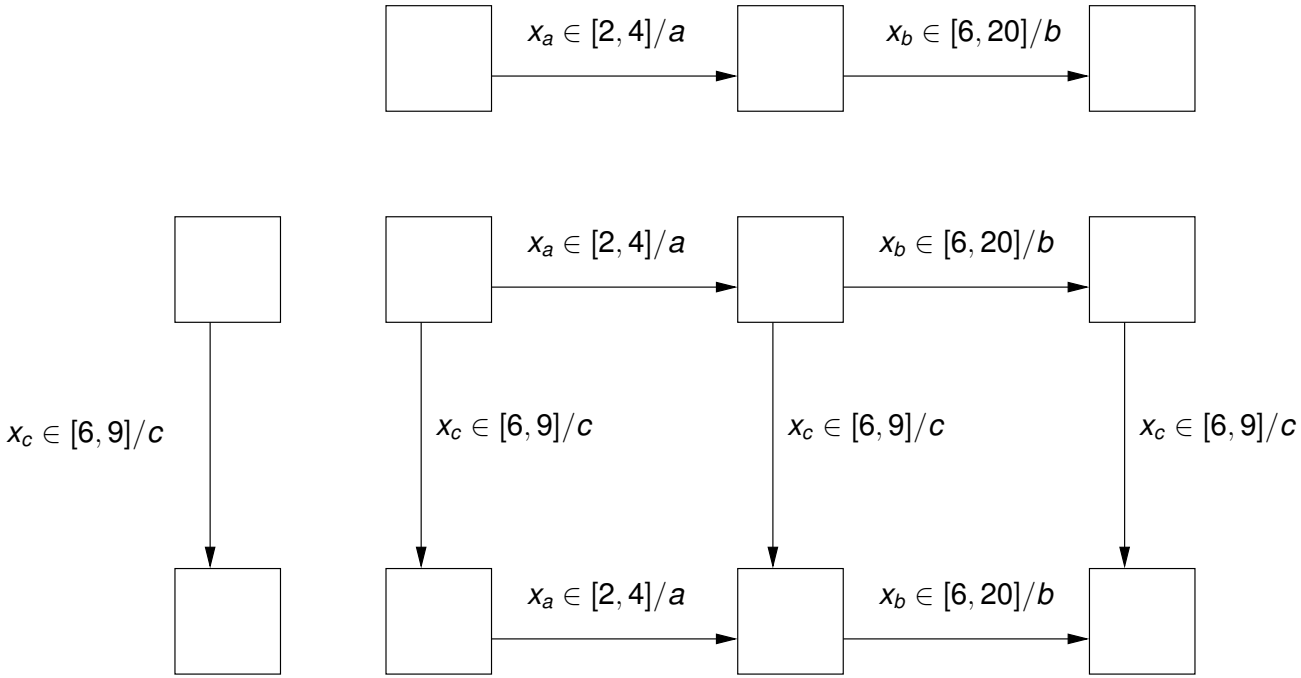
- Timed automata and similar formalisms assume a lower and (finite) upper bound for the duration of each step



- The arithmetics of time eliminates some paths:
- Since $4 < 6$, a must precede c and the set of possible paths is reduced to $a \cdot (b||c) = abc + acb$
- But how likely is abc to occur?

Levels of Abstraction: Timed

- But how likely is *abc* to occur?



- The durations of the steps is a vector $(y_a, y_b, y_c) \in Y = [2,4] \times [6,20] \times [6,9]$
- Event b precedes c only when $y_a + y_b < y_c$
- Since $y_a + y_b$ ranges in $[8,24]$ and $y_c \in [6,9]$, it is less likely than c preceding b

Probabilistic Interpretation of Timing Uncertainty

- Interpreting temporal guards probabilistically as **uniform distribution** over $[a, b]$ gives precise quantitative meaning to this intuition
- Using this model we can compute probabilities of paths as **volumes** in the **duration space**
- We can discard low-probability paths, compute expected performance of schedulers, etc.
- This talk explains how to do it gradually
  1. A single sequential process
  2. Multiple independent processes
  3. Processes executing under scheduler coordination

Sequential Stochastic Processes I

- $S = P^1 || \cdots || P^n$ of *n sequential stochastic processes*
- A process is a sequence of steps with probabilistic duration
- A step cannot start before its predecessor terminates
- Two scenarios:
  - Independent executions
  - Coordinated execution: resource conflicts on some steps, resolved by a scheduler that guarantees mutual exclusion
- We want to compare the (expected) performance of scheduling policies for the second scenario
- We start with the first for didactic reasons

Bounded Uniform Distributions

- A *uniform* distribution inside an interval $I = [a, b]$ is characterized by a density ψ defined as

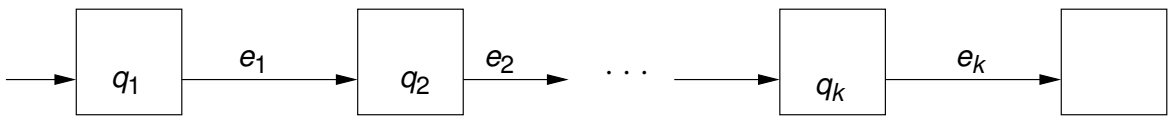
$$\psi(y) = \begin{cases} 1/(b-a) & \text{if } a \leq y < b \\ 0 & \text{otherwise} \end{cases}$$

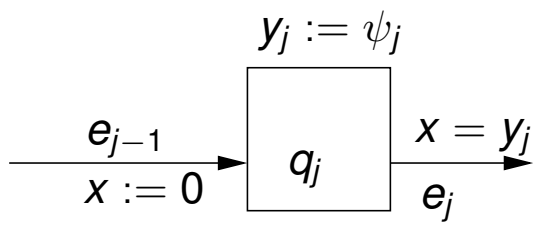
- Or in terms of distribution:

$$F(y) = \int_0^y \psi(\tau) d\tau = \begin{cases} 0 & \text{if } y < a \\ (y-a)/(b-a) & \text{if } a \leq y \leq b \\ 1 & \text{if } b \leq y \end{cases}$$

Sequential Stochastic Processes II

- A sequential stochastic process: $P = (\mathcal{I}, \Psi)$:
- $\mathcal{I} = \{I_j\}_{j \in K}$ where $I_j = [a_j, b_j]$ is the interval of possible durations of step P_j
- $\Psi = \{\psi_j\}_{j \in K}$ is a sequence of densities with each ψ_j uniform over I_j
- We consider finite acyclic processes with $K = \{1, \ldots, k\}$
- Automaton view:





Duration Space

- A finite sequence of *independent* uniform random variables $\{y_j\}_{j \in K}$ ranging over a *duration space* D, consisting of vectors

$$y = (y_1, \ldots, y_k) \in D = I_1 \times \cdots \times I_k \subseteq \mathbb{R}^k$$

with density

$$\psi(y_1, \ldots, y_k) = \psi_1(y_1) \cdots \psi_k(y_k)$$

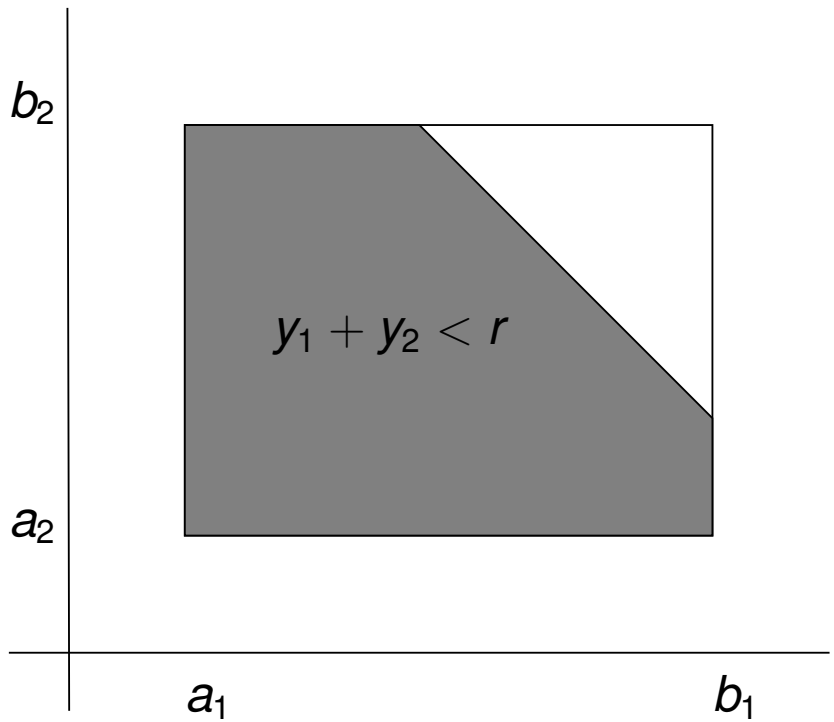
- A point $y \in D$ induces a *unique* behavior of the system

$$\xi_y = y_1 \ e_1 \ y_2 \ e_2 \ \cdots \ y_k \ e_k$$

where $y_j \in I_j$ is the *duration* of step P_j and e_j is the *termination event*

Volume and Probability

- The timed language of the process $L = \{\xi_y : y \in D\}$
- The untimed (qualitative) language $\underline{L} = \{e_1\ e_2 \cdots\ e_k\}$
- The probability of any subset of L is the relative volume of the subset of D that generates it
- For example, the probability to terminate before deadline r:
- The volume of $D \wedge (y_1 + \cdots + y_k < r)$ divided by the volume of D



From Durations to Time Stamps

- A timed word $\xi_y = y_1\ e_1\ y_2\ e_2\ \cdots\ y_k\ e_k$ can be written as a sequence of time-stamped events
$$\xi_t = (e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k)$$
- where $t_j = y_1 + \cdots + y_j$ is the absolute time of e_j
$y_j = t_j - t_{j-1}$
- A coordinate transformations $t = Ty$ and $y = T't$ between the duration space D and the time-stamp space C

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \qquad T' = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

- These transformations preserve volume. We do our calculations on the time-stamp space C which is a *zone* defined by
$$\varphi_C : \quad \bigwedge_{j \in K} a_j \leq t_j - t_{j-1} \leq b_j$$

Processes in Parallel

- Consider n processes $S = P^{\mathbf{1}} || \cdots || P^{\mathbf{n}} = \{(\mathcal{I}^{\mathbf{i}}, \Psi^{\mathbf{i}})\}_{i=1}^{n}$
- Notations: $P_j^{\mathbf{i}}$ (step j of process i), $I_j^{\mathbf{i}} = [a_j^{\mathbf{i}}, b_j^{\mathbf{i}}]$ and $\psi_j^{\mathbf{i}}$
- All processes have the same number k of steps
- Event alphabet $\Sigma = \{e_1^{\mathbf{1}}, e_2^{\mathbf{1}}, \ldots, e_{k-1}^{\mathbf{n}}, e_k^{\mathbf{n}}\}$
- A global behavior corresponds to a point in the global duration space

$$y = (y_1^{\mathbf{1}}, y_2^{\mathbf{1}}, \ldots, y_{k-1}^{\mathbf{n}}, y_k^{\mathbf{n}}) \in \mathcal{D} = \prod_{i=1}^{n} \prod_{j=1}^{k} I_j^{\mathbf{i}} \subset \mathbb{R}^{nk}$$

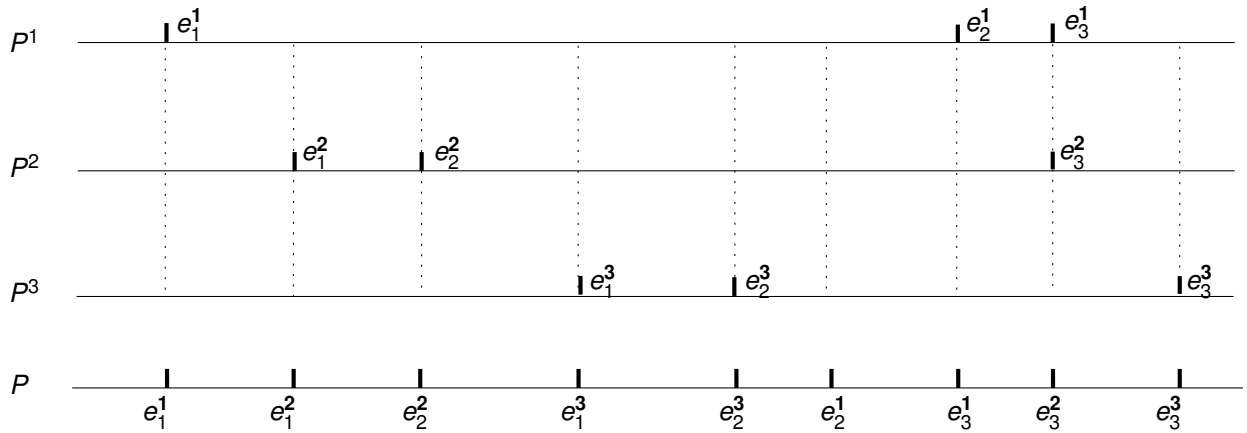
or equivalently to a point t in the time-stamp space

$$t = (t_1^{\mathbf{1}}, t_2^{\mathbf{1}}, \ldots, t_{k-1}^{\mathbf{n}}, t_k^{\mathbf{n}}) \in \mathcal{C} = T\mathcal{D}$$

where T is a block diagonal matrix.

Global Behaviors

- Merging local behaviors $L = L^1 || \cdots || L^n$

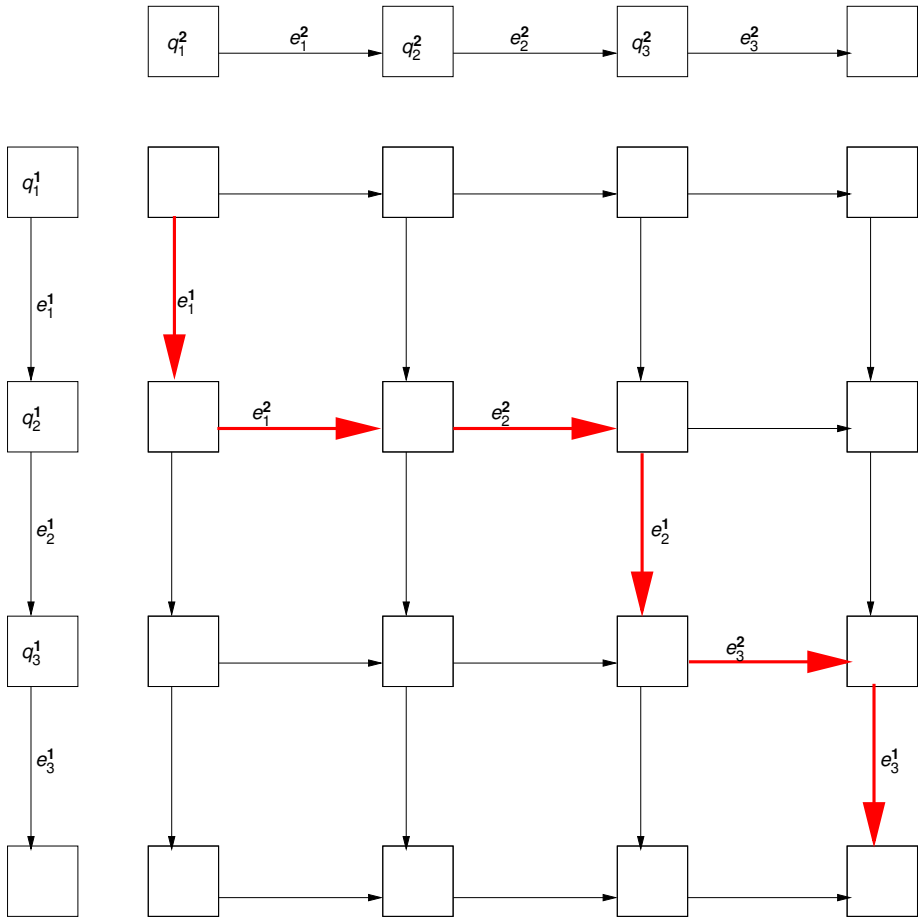


$$w = e_1^1\ e_1^2\ e_2^2\ e_1^3\ e_3^2\ e_2^1\ e_3^1\ e_2^3\ e_3^3$$

- Qualitative behavior: equivalence class of all timed behaviors with the same *order* of events
- All potentially possible behaviors are part of the *shuffle* (interleavings) of the local languages $\underline{L} = \underline{L}^1 || \cdots || \underline{L}^n$

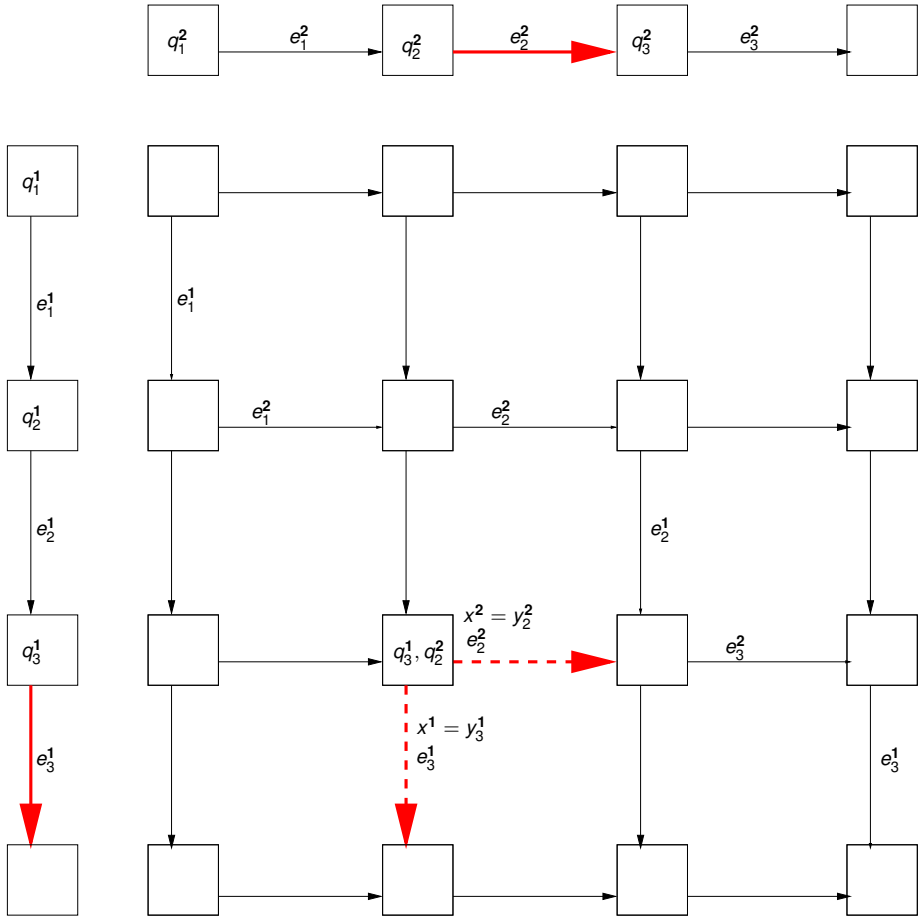
Automaton View

- A qualitative behavior is the set of all runs that go through the same path in the global (product) automaton



$$w = e_1^1\ e_1^2\ e_2^2\ e_1^3\ e_3^2\ e_2^1\ e_3^1\ e_2^3\ e_3^3$$

Races



- In state (q_3^1, q_2^2) there is a *race* between e_3^1 and e_2^2
- The winner depends on which termination condition (transition guard) is satisfied first
- Which reduces to the relation between t_3^1 and t_2^2

Probability of Qualitative Behavior

- We formulate the following question:
- Compute the probability of a qualitative behavior w, ie the probability that events occur in a particular order
- Two-stage solution: characterize the subset Z_w of the time-stamp space $\mathcal{C}$ that yields w
- Compute the volume of this subset divided by the volume of $\mathcal{C}$
- This will be expressed by a constraint $\varphi_{\mathcal{C}} \wedge \varphi_w$ with

$$\varphi_{\mathcal{C}} : \quad \bigwedge_{i \in N} \bigwedge_{j \in K} a_j^i \leq t_j^i - t_{j-1}^i \leq b_j^i$$

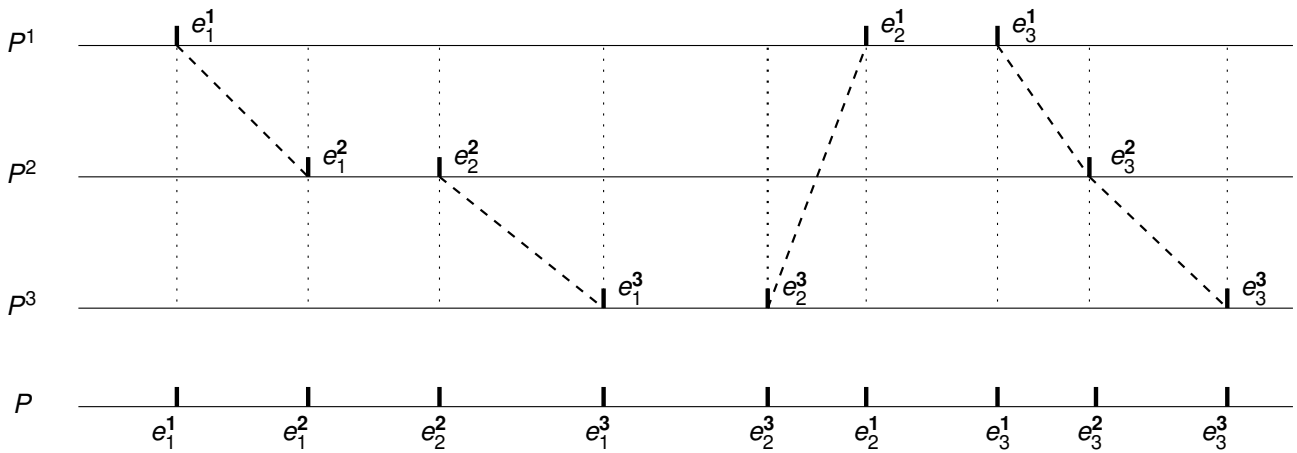
Zone of a Qualitative Behavior

- Example: $w = e_1^1\ e_1^2\ e_2^2\ e_1^3\ e_3^2\ e_2^1\ e_3^1\ e_2^3\ e_3^3$

$$\varphi_w :\ \varphi_{\mathcal{C}} \wedge\ t_1^1 < t_1^2 < t_2^2 < t_1^3 < t_3^2 < t_2^1 < t_3^1 < t_2^3 < t_3^3$$

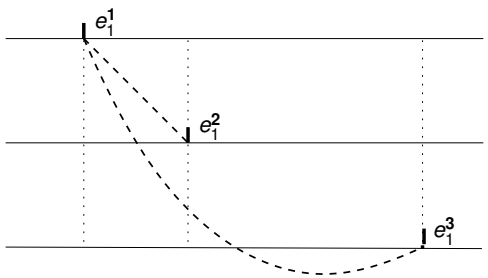
- Some constraints are implied by $\varphi_{\mathcal{C}}$ and transitivity
- The minimal set of inter-process constraints that characterize w:

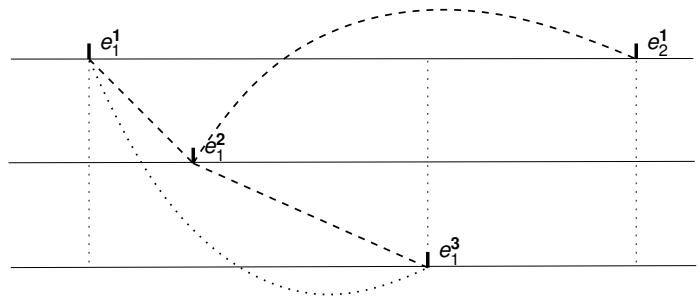
$$\varphi_w : \varphi_{\mathcal{C}} \wedge (t_1^1 < t_1^2) \wedge (t_2^2 < t_1^3) \wedge (t_2^3 < t_2^1) \wedge (t_3^1 < t_3^2) \wedge (t_3^2 < t_3^3)$$



Incremental Construction

- Constraints can be computed *incrementally* as we move along the *prefix* of a qualitative behavior
- For every w the probability of all behaviors having w as a prefix is $p(w) = |Z_w|/|\mathcal{C}|$
- $\varphi_\epsilon : \varphi_\mathcal{C}$
- $\varphi_{e_1^1} : \varphi_\mathcal{C} \wedge (t_1^1 < t_1^2) \wedge (t_1^1 < t_1^3)$
- $\varphi_{e_1^1 e_1^2} : \varphi_\mathcal{C} \wedge (t_1^1 < t_1^2) \wedge (t_1^2 < t_1^3) \wedge (t_1^2 < t_2^1)$





- When a new event occurs Z_w is split among its successors satisfying

$$\sum_e |Z_{w\ e}| = |Z_w|$$

Integration: Back to School

- The volume of Z_w is computed by integration
- A concrete example: 3 one-step processes

$$\mathcal{D} = \mathcal{C} = [2,5] \times [3,4] \times [4,7]$$

- To compute the probability that P^1 makes the first step

$$\varphi_{e_1^1} : \quad (2 \leq t_1^1 \leq 5) \wedge (3 \leq t_1^2 \leq 4) \wedge (4 \leq t_1^3 \leq 7) \wedge (t_1^1 < t_1^2) \wedge (t_1^1 < t_1^3)$$

- We choose integration order (order of variable elimination) $t_1^3 \prec t_1^2 \prec t_1^1$:

$$|Z_{e_1^1}| = \int_2^3 \int_{\max(3,t_1^1)}^4 \int_{\max(4,t_1^1)}^7 dt_1^3 dt_1^2 dt_1^1$$

Integration: Back to School

- To compute

$$\int_2^3 \int_{\max(3,t_1^1)}^4 \int_{\max(4,t_1^1)}^7 dt_1^3\, dt_1^2\, dt_1^1$$

we split I_1^1 as $[2,5] = [2,3] \cup [3,4] \cup [4,5]$

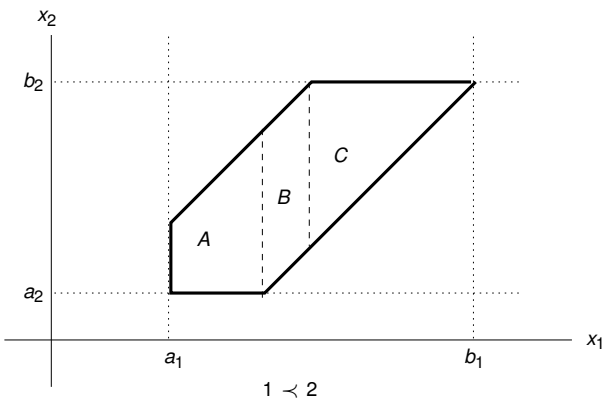
$$\left[\int_2^3 \int_3^4 \int_4^7 + \int_3^4 \int_{t_1^1}^4 \int_4^7 + \int_4^5 \int_{t_1^1}^4 \int_{t_1^1}^7 \right] dt_1^3\, dt_1^2\, dt_1^1$$

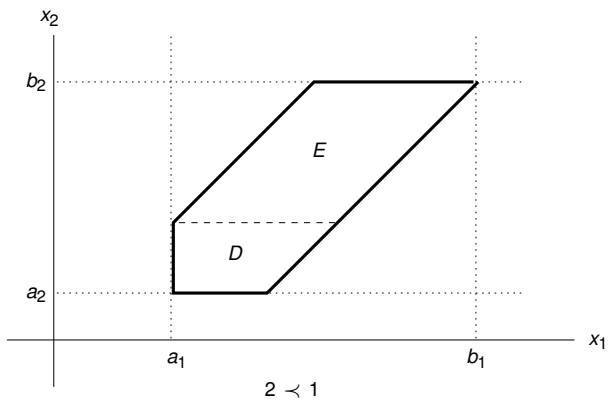
$$= 3 + \tfrac{3}{2} + 0 = \tfrac{9}{2}$$

- Dividing by $|\mathcal{C}| = 9$ gives a probability of $1/2$ for e_1^1 winning the first race

Integration over Zones

- First, we use DBM to check if a zone is empty
- Then in n dimensions there are $n!$ possible orders of integration
- Each order yields different splits and different forms of intermediate objects





- Orders of magnitude differences in complexity
- Our heuristic so far is to eliminate “later” variables first

Theorem 1

- The probability of a qualitative behavior in a system of acyclic stochastic sequential processes with uniform probabilistic durations is computable
- From this we can also compute the *expected makespan* (total termination time)
- In any behavior of the form $w = w' e^{\mathbf{i}}_k$ process $P^{\mathbf{i}}$ is the last to terminate and the total termination time is $t^{\mathbf{i}}_k$
- The expected termination time is

$$\mathbb{E}(\Theta) = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{n} \sum_{w = w' e^{\mathbf{i}}_k} \int_{Z_w} t^{\mathbf{i}}_k .$$

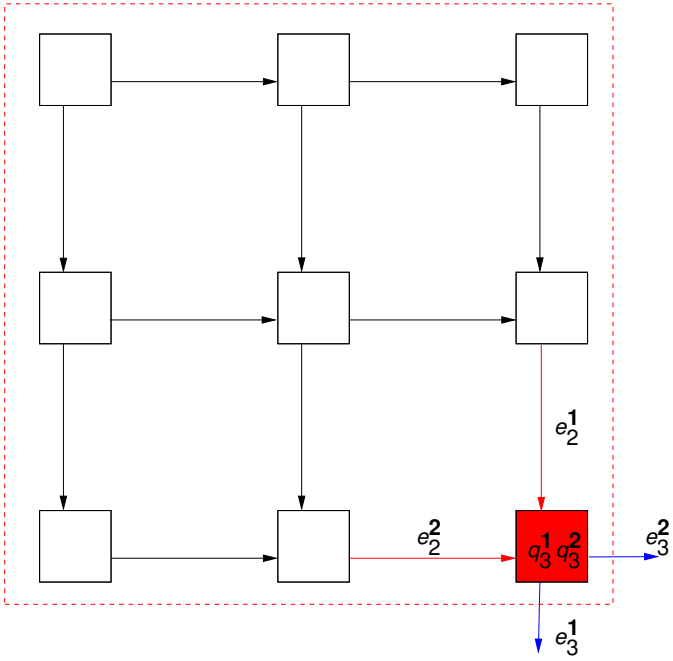
- Corollary: expected makespan is computable

Confluent Paths

- This can be, of course, computed much more efficiently
- All qualitative behaviors that pass through a global state $q = (q^{\mathbf{1}}_{j_1}, \ldots, q^{\mathbf{n}}_{j_n})$ are characterized by

$$\varphi_q : \varphi_{\mathcal{C}} \wedge \bigwedge_{i=1}^{n} \bigwedge_{i' \neq i} t^{\mathbf{i}}_{j_i - 1} < t^{\mathbf{i}'}_{j_{i'}}$$

- We can forget the order among past events (paths to q)

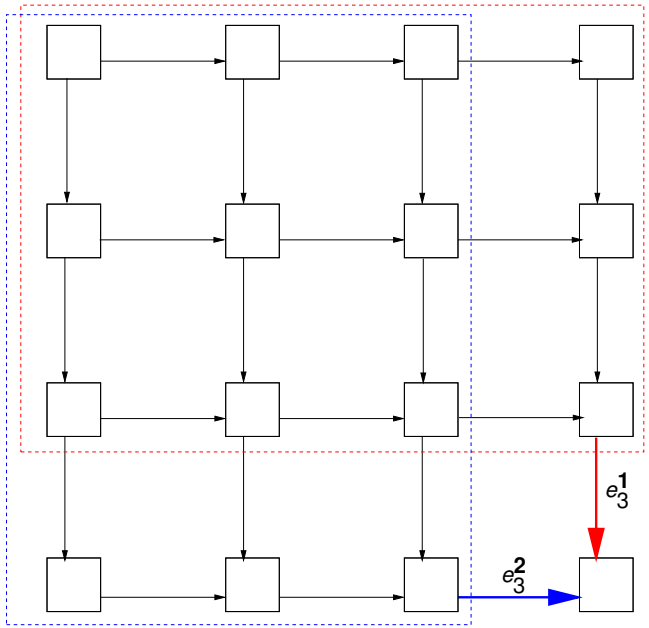


$$t^{\mathbf{1}}_2 < t^{\mathbf{2}}_3 \wedge t^{\mathbf{2}}_2 < t^{\mathbf{1}}_3$$

Confluent Paths

- The qualitative behaviors where $P^{\mathbf{i}}$ makes the last step correspond to the zone $Z^{\mathbf{i}}$ characterized by

$$\varphi^i : \ \varphi c \wedge \bigwedge_{i' \neq i} t_k^{\mathbf{i}'} < t_k^{\mathbf{i}}$$



- The expected termination time is

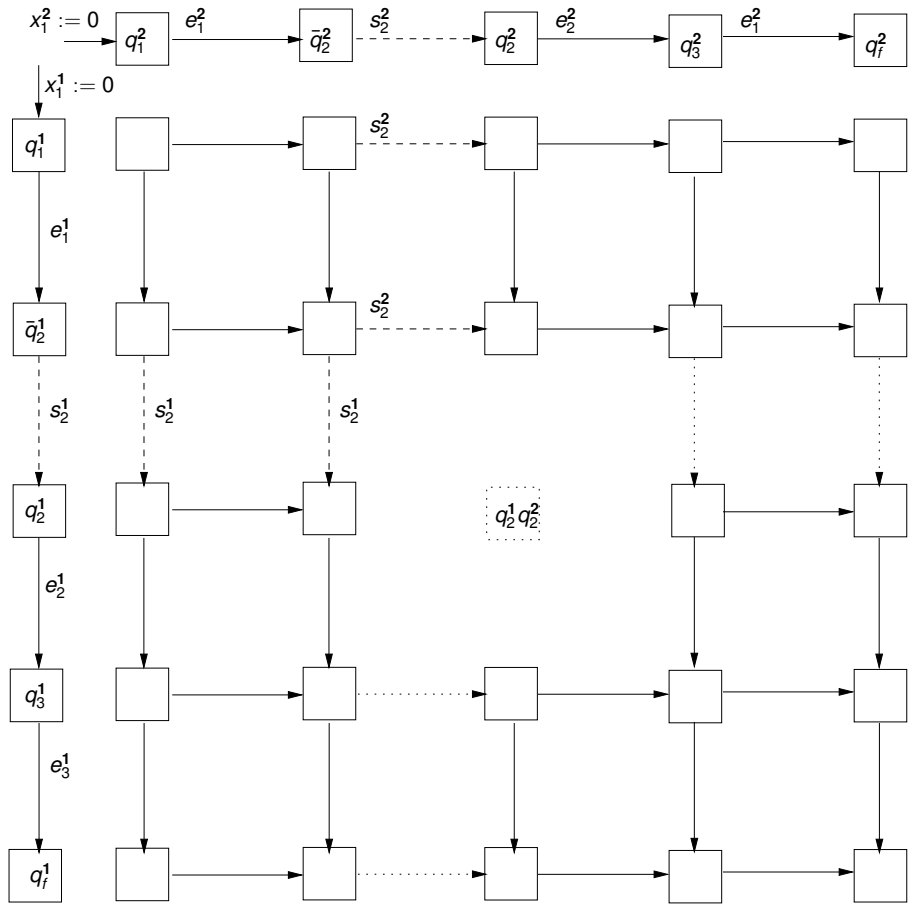
$$\mathbb{E}(\Theta) = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{n} \int_{Z^i} t_k^{\mathbf{i}}$$

Coordinated Execution

- This concludes the warm-up, now we move to serious stuff
- We assume that steps of different processes can be in *conflict* as they require the same bounded resource
- A scheduler should decide to whom to give the resource first based on some policy
- Starting P_j is not automatic upon the termination of P_{j-1}
- We modify the process automaton by inserting a *waiting state* $\bar{q}_j^{\mathbf{i}}$ between $q_{j-1}^{\mathbf{i}}$ and $q_j^{\mathbf{i}}$
- The automaton can leave this state only when it receives a *start* command $s_j^{\mathbf{i}}$ from a scheduler

A Running Example

- Two 3-step processes, a conflict between P_2^1 and P_2^2
- A *forbidden state* (q_2^1, q_2^2) that no scheduler allows in



Non-Determinism Resolved by Schedulers

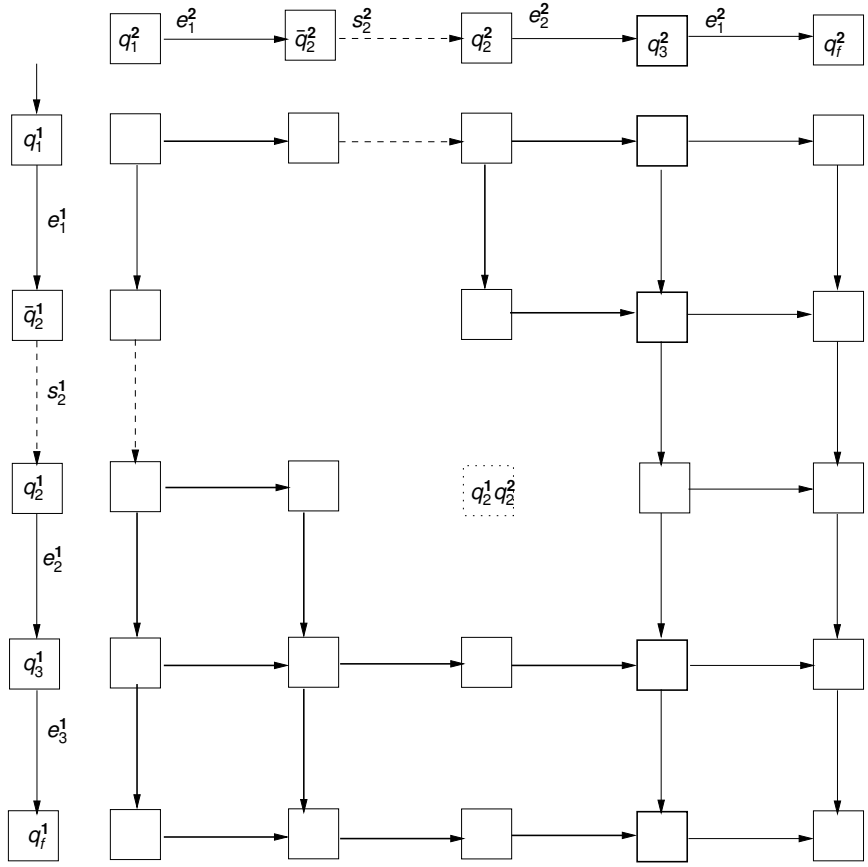
- Before the scheduling policy is defined, the system is not probabilistically correct
- It is “open”, mixing probability with measure-free non-determinism (CS style)
- A scheduling policy eliminates this non-determinism and replaces it by determinism
- A point in the duration space induces a unique behavior
- We will compute probabilities and expected makespan using an extension of the volume-based technique
- We use non-lazy schedulers that do not block a process from using a resource unless another process will benefit from its waiting

Types of Schedulers

- One can consider various types of schedulers varying between two extremes
- *Laissez faire*: a liberal FIFO scheduler that gives a resource which is in conflict to the first task that requires it
- *Control freak*: a priority relation for each resource in conflict. Conflicting tasks are always executed according to this order
- *In between*: the decision of the scheduler to allow a task to take a resource is based on the *global* state of the system

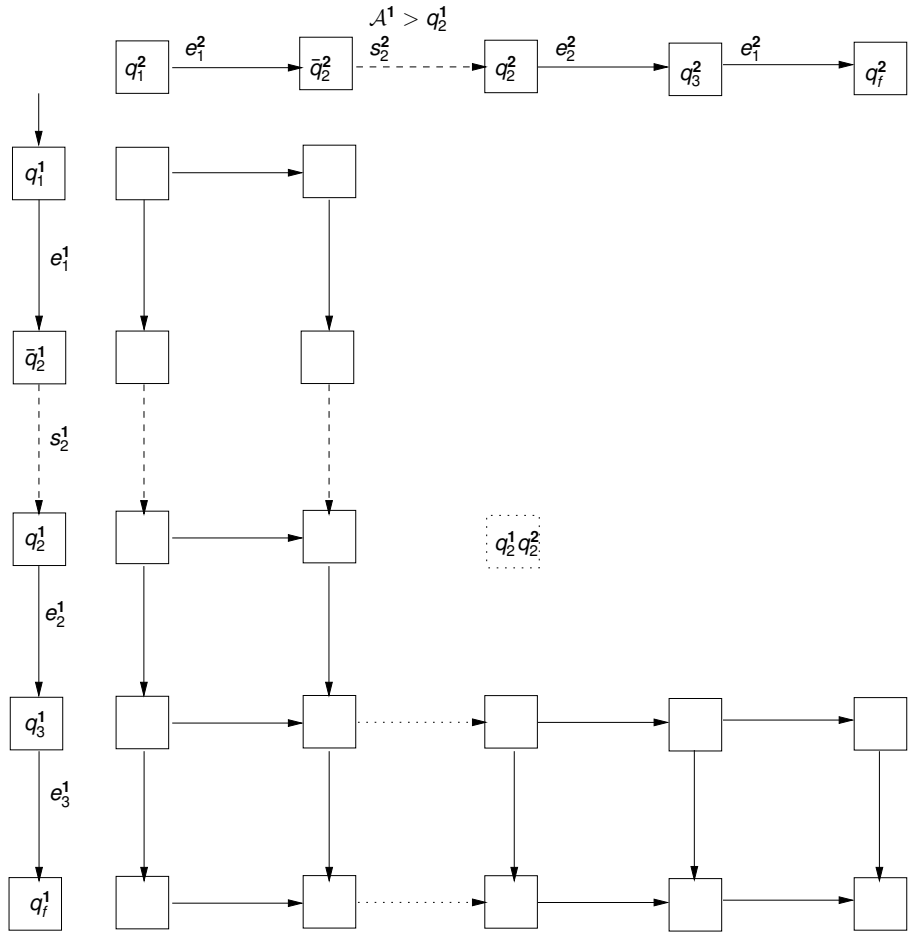
The FIFO Scheduler

- Advantage: natural, no need to think
- Disadvantage: a step of another process which is on the critical path may arrive later and will have to wait



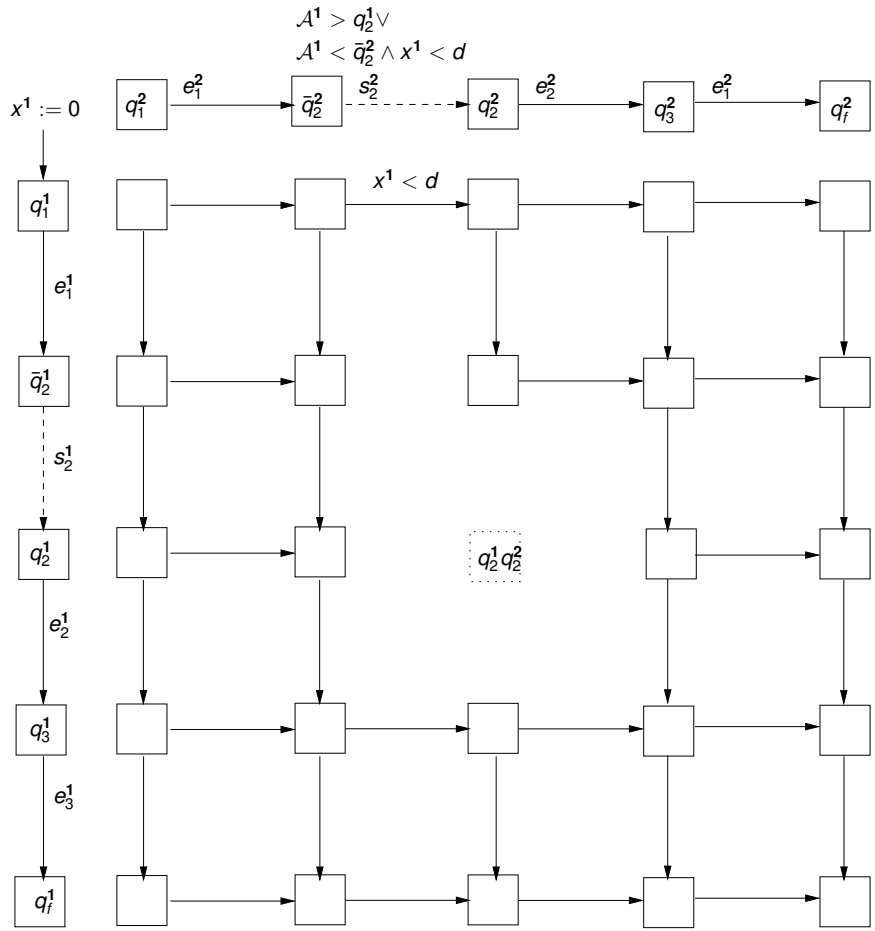
Strict Priority Scheduler

- Advantage: a more global view can keep the resource free for a critical task
- Disadvantage: hard to compute, not adaptive to actual durations, cannot use opportunities



Conditional Priority

- Advantage: the most general and adaptive and hence contains the optimal scheduler;
- Disadvantage: even harder to compute and requires more runtime information to realize



Computing Volumes

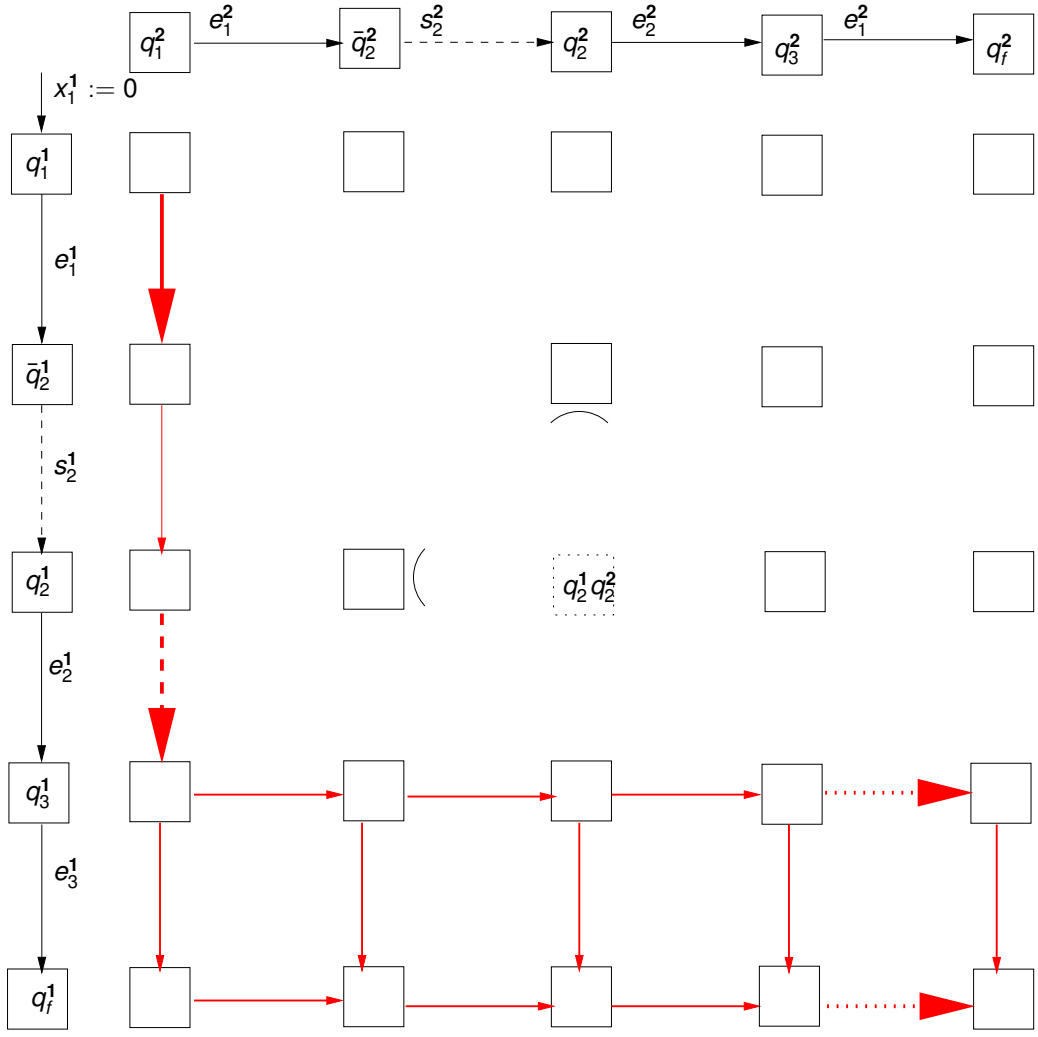
- We adapt the path labeling and volume computation procedures for coordinated execution
- We illustrate on the FIFO schedulers but it extends easily to other schedulers
- In fact, FIFO schedulers may admit more possible scenarios than priority based schedulers and hence the computation is harder
- The crucial point in the coordinated execution scenario:
- The value of $t_j^{\mathbf{i}}$ may sometimes depend on its predecessor $t_{j-1}^{\mathbf{i}}$ and sometimes on $t_{j'}^{\mathbf{i}'}$ where $P_{j'}^{\mathbf{i}'}$ is a process that is in conflict with $P_{j'}^{\mathbf{i}'}$

Conflict Outcome

- In a conflict between two processes P^1 and P^2 there are 4 possible outcomes depending on:
- Who wins and uses the resource first? For FIFO schedulers this depends on who terminates before the step preceding the conflict
- Is the loser delayed? Does it become enabled before or after the winner terminates the conflicting step
- Each scenario can be expressed as a zone in the time stamp space
- Such a zone corresponds to a polytope in the duration space which has the same volume

Case 1: P^1 Wins but P^2 is Not Delayed

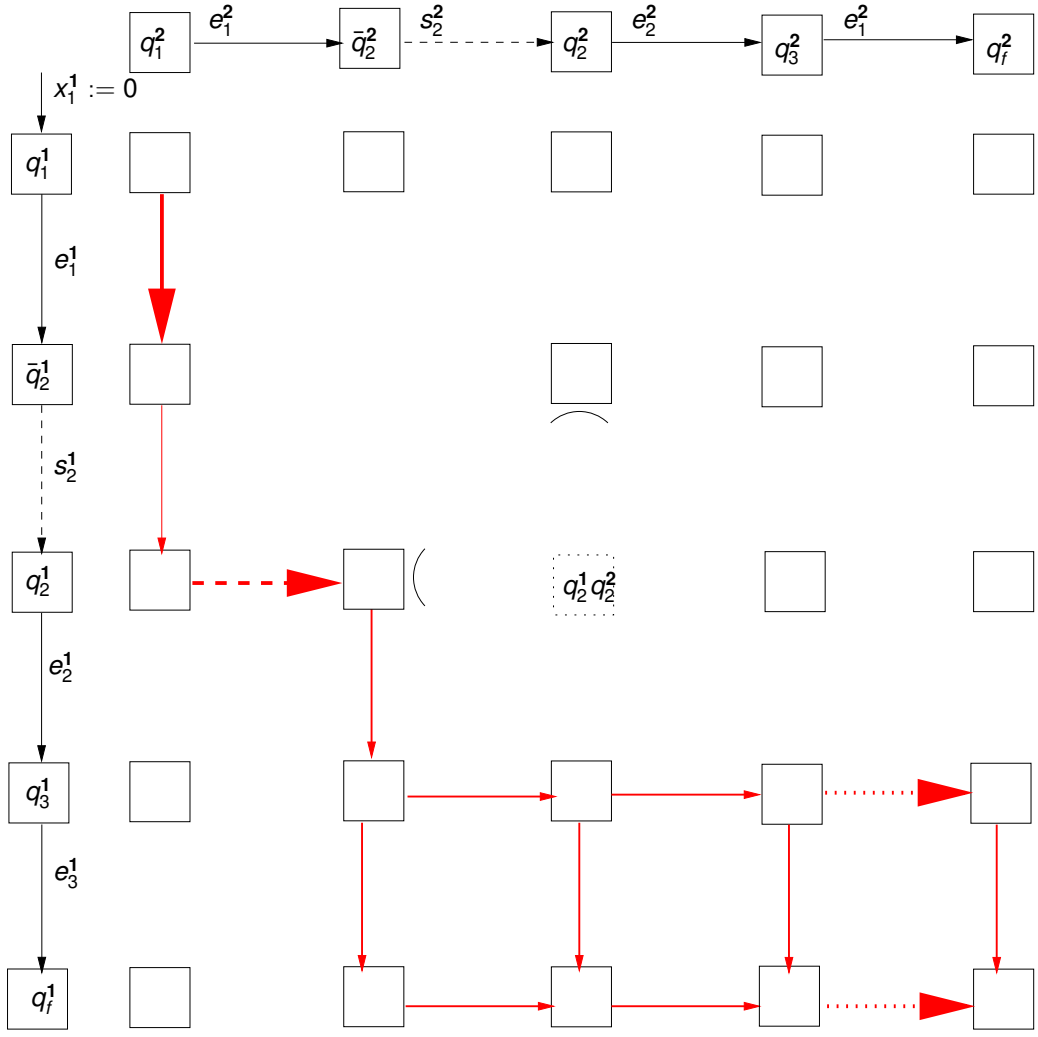
- $t_1^1 < t_1^2 \qquad t_2^1 < t_1^2$



- $t_1^2 + a_2^2 < t_2^2 < t_1^2 + b_2^2$

Case 2: P^1 Wins and P^2 is Delayed

- $t_1^1 < t_1^2 \qquad t_1^2 < t_2^1$



- $t_2^1 + a_2^2 < t_2^2 < t_2^1 + b_2^2$

Computing Probabilities

- The qualitative behaviors are partitioned into equivalence classes
- Each class is characterized by the **utilization scenario** of each of the shared resources:
- At what **order** it is utilized and which steps are **delayed**
- For each class we construct a zone in the time-step space having the same volume as the subset of the duration space that induces it
- The coordinate transformation from $\mathcal{D}$ to $\mathcal{C}$ becomes piecewise-linear
- A priori, a severe combinatorial explosion but in practice many zones are empty because the scenarios violate duration and precedence constraints

Implementation

- A prototype tool:
- Computes the zone for each utilization scenario, using the DBM library of IF to simplify and check emptiness
- Performs integration over the non-empty zones to compute probabilities and expected termination time
- Integration uses high-precision arithmetic (GMP library) to avoid rounding errors
- A heuristic to determine the order of variable elimination integration based on a fast estimation of their ranges
- Preliminary performance observations: can solve (in < 3 minutes) problems with $(n, k) = (1, 63*), (2, 12), (4, 6), (5, 4)$ with two or three conflicts

Future Work

- Improve the algorithm for integration over zones
- Extend to other distributions
- To avoid explosion, develop a **fat-first** exploration procedure that stops when the **accumulated** probability crosses some threshold
- It needs a quick volume estimation procedure
- Extend the approach to **cyclic** systems and infinite behaviors: define suitable performance measures and compute their steady-states
- From analysis to **synthesis**: derive controller which are average-case optimal
- Compare and combine with Monte-Carlo simulation