Markovian Behavioral Equivalences: Their Spectrum, Some Known Results, and One Open Problem

Marco Bernardo
University of Urbino – Italy

Part I:
Introduction

Performance-Oriented Notations

- Building performance-aware models of computing systems:
  - Predicting the satisfiability of QoS requirements.
  - Choosing among alternative designs based on their expected QoS.
- *Theory*:
  - Queueing networks (1950's).
  - Stochastic Petri nets (1980's).
  - Stochastic process calculi (1990's).
- *Practice*:
  - System/software performance engineering approaches.
  - Object-oriented modeling languages (UML profiles).
  - Architectural description languages (ÆMILIA).
  - Formal modeling languages (MODEST).
  - Coordination languages (STOKLAIM).

- Performance-oriented notations usually produce behavioral models.
- These models can be uniformly expressed as state transition graphs.
- Representation of the current state:
  - Current number of customers in each service center.
  - Current Petri net marking.
  - Current process term.
- Cause of the state change associated with a transition:
  - Execution of a certain activity.
  - Occurrence of a certain event.
- Adoption of an interleaving view of concurrency in which independent activities can be executed in any order but not simultaneously.

Behavioral Equivalences

- Behavioral models are equivalent whenever they represent systems that behave the same.
- Need for the introduction of behavioral equivalences.
- Useful for theoretical and applicative purposes:
  - Comparing models that are syntactically different on the basis of the behavior they exhibit.
  - Relating models of the same system at different abstraction levels (top-down modeling).
  - Manipulating models in a way that preserves certain properties (state space reduction before analysis).

- Most studied approaches developed in a purely functional framework:
  - Bisimilarity: two models are equivalent if they are able to *mimic* each other's behavior *stepwise*.
  - Testing: two models are equivalent if an *external observer* cannot distinguish between them by interacting with them by means of *tests* and comparing their reactions.
  - Trace: two models are equivalent if they are able to perform the same *sequences* of activities.
- How to extend behavioral equivalences to performance-aware models?
- It is necessary to take into account *quantitative aspects* related to system evolution over time (event probabilities, activity durations, costs/gains, . . .).

Markovian Framework

- A Markov chain is a discrete-state stochastic process $\{RV(t) \mid t \in \mathbb{R}_{\geq 0}\}$ such that for all $n \in \mathbb{N}$, time instants $t_0 < t_1 < \ldots < t_n < t_{n+1}$, and states $s_0, s_1, \ldots, s_n, s_{n+1} \in S$:

$$\boxed{\begin{aligned} &\Pr\{RV(t_{n+1}) = s_{n+1} \mid RV(t_0) = s_0 \wedge RV(t_1) = s_1 \wedge \ldots \wedge RV(t_n) = s_n\} \\ &\quad = \Pr\{RV(t_{n+1}) = s_{n+1} \mid RV(t_n) = s_n\} \end{aligned}}$$

- The past history is completely summarized by the current state.
- Equivalently, the stochastic process has no memory of the past.
- Time homogeneity: probabilities independent of state change times.
- The solution of a Markov chain is its state probability distribution $\boldsymbol{\pi}()$ at an arbitrary time instant (CTMC vs. DTMC).

- Representation and solution of a continuous-time Markov chain (CTMC):
  - State transitions are described by a rate matrix $\boldsymbol{Q}$.
  - The sojourn time in any state is exponentially distributed.
  - Given $\boldsymbol{\pi}(0)$, the transient solution $\boldsymbol{\pi}(t)$ is obtained by solving:

$$\boxed{\boldsymbol{\pi}(t) \cdot \boldsymbol{Q} = \frac{\mathrm{d}\boldsymbol{\pi}(t)}{\mathrm{d}t}}$$

  - The stationary solution $\boldsymbol{\pi} = \lim_{t \to \infty} \boldsymbol{\pi}(t)$ is obtained (if any) by solving:

$$\boxed{\begin{array}{rcl} \boldsymbol{\pi} \cdot \boldsymbol{Q} & = & \mathbf{0} \\ \sum_{s \in S} \pi[s] & = & 1 \end{array}}$$

- Exponentially distributed random variables are the only continuous random variables satisfying the memoryless property:

$$\boxed{\Pr\{RV \leq v + v' \mid RV > v'\} = \Pr\{RV \leq v\}}$$

- Every CTMC (time-aware model) has an embedded DTMC (time-abstract model):
  - State transitions are described by a probability matrix $\boldsymbol{P}$.
  - $\boldsymbol{P}$ is obtained from $\boldsymbol{Q}$ by dividing the rate of each transition by the sum of the rates of the transitions that depart from the source state.
  - The sojourn time in any state is geometrically distributed.
  - Given $\boldsymbol{\pi}(0)$, the transient solution $\boldsymbol{\pi}(n)$ is computed as follows:

$$\boxed{\boldsymbol{\pi}(n) \;=\; \boldsymbol{\pi}(0) \cdot \boldsymbol{P}^n}$$

  - The stationary solution $\boldsymbol{\pi} = \lim_{n \to \infty} \boldsymbol{\pi}(n)$ is obtained (if any) by solving:

$$\boxed{\begin{aligned} \boldsymbol{\pi} &= \boldsymbol{\pi} \cdot \boldsymbol{P} \\ \sum_{s \in S} \pi[s] &= 1 \end{aligned}}$$

- A CTMC is a state transition graph in which every transition is labeled with a positive real number expressing the rate at which the state change takes place.
- Rates subsume both time information and probability information:
  - The sojourn time in a state is exponentially distributed with rate given by the sum of the rates of the outgoing transitions.
  - The probability of executing a transition is proportional to its rate.
- A CTMC can thus be viewed as a state transition graph in which:
  - Every state has an exponentially distributed random variable associated with it that expresses the sojourn time.
  - Every transition has a positive real number not greater than 1 associated with it that expresses the execution probability.

Markovian Behavioral Equivalences

- Focus on exponential distributions for activity durations.
- Their memoryless property results in a simpler mathematical treatment:
  - $\triangleleft$ Compliance with the *interleaving view* of concurrency.
  - $\triangleleft$ Easy calculation of *state sojourn times* and *transition probabilities*.

  without sacrificing expressiveness:
  - $\triangleright$ Adequate for modeling the timing of many *real-life phenomena* like arrival processes, failure events, and chemical reactions.
  - $\triangleright$ Most appropriate stochastic approximation in the case in which only the *average duration* of an activity is known.
  - $\triangleright$ Proper combinations (phase-type distributions) approximate most of *general distributions* arbitrarily closely.
- How to define Markovian behavioral equivalences?

Process Algebraic Markovian Modeling

- Behavioral equivalences abstract from the specific kind of model but ...
- ... are better investigated and understood in a process algebraic setting.
- Action-based modeling relying on a set of behavioral operators.
- Performance-oriented process calculi with CTMC semantics:
  - ⊙ TIPP [Götz, Herzog, Rettelbach].
  - ⊙ PEPA [Hillston].
  - ⊙ MPA [Buchholz].
  - ⊙ $\mathrm{EMPA_{gr}}$ [Bernardo, Bravetti, Gorrieri].
  - ⊙ Sπ [Priami].
  - ⊙ IMC [Hermanns].
  - ⊙ PIOA [Stark, Cleaveland, Smolka].

- Markovian process calculi differ for the *action representation*.
- Durational actions (integrated time):
  - An action is executed while time passes.
  - Single action prefix operator comprising the name a of the action and the rate $\lambda \in \mathbb{R}_{>0}$ of the exponentially distributed random variable quantifying the duration of the action: $<a, \lambda>.\_$
  - The choice among several actions is probabilistic.
  - TIPP, PEPA, MPA, $\mathrm{EMPA_{gr}}$, Sπ, PIOA.
- Action names separated from time (orthogonal time):
  - An action is instantaneously executed after some time has elapsed.
  - Two action prefix operators: $(\lambda).\_$ and $a.\_$
  - The choice among several actions is nondeterministic.
  - IMC.

- Markovian process calculi also differ for the discipline adopted for *action synchronization*.
- In the orthogonal time case, action synchronization is governed as in nondeterministic process calculi.
- In the integrated time case, action synchronization can be handled in different ways.
- The more natural choice for deciding the duration of the synchronization of two exponentially timed actions would be to take the maximum of their durations.
- *The maximum of two exponentially distributed random variables is not exponentially distributed* (phase-type: IMC).

- Symmetric synchronizations:
  - The synchronization of two exponentially timed actions is assumed to be exponentially timed.
  - Its rate is defined through an associative and commutative operator applied to the two original rates (multiplication, min, max).
  - TIPP, PEPA, MPA, Sπ.
- Asymmetric synchronizations:
  - Passive actions of the form $<a, *_w>$ whose duration is unspecified.
  - An exponentially timed action can synchronize only with a passive action, thus determining the duration of the synchronization.
  - PEPA, EMPA$_{\mathrm{gr}}$, PIOA.
- *Bounded capacity assumption*: the rate of an action should not increase when synchronizing that action with other actions.

Markovian Process Calculus: Syntax

- Basic design choices: durational actions (more natural modeling style) and asymmetric synchronizations (exp. timed action can synch. only with passive actions).
- $Name_{\mathrm{v}}$: set of visible action names.
- $Name = Name_{\mathrm{v}} \cup \{\tau\}$: set of all action names.
- $Rate = \mathbb{R}_{>0} \cup \{*_w \mid w \in \mathbb{R}_{>0}\}$: set of action rates.
- $Act_{\mathrm{M}} = Name \times Rate$: set of exponentially timed and passive actions.
- $Relab = \{\varphi : Name \rightarrow Name \mid \varphi^{-1}(\tau) = \{\tau\}\}$: set of visibility-preserv. relabeling functions.
- Var: set of process variables ($Const$: set of process constants).

- Process term syntax for process language $\mathcal{PL}_{\mathrm{M}}$:

P	$::=$	$\underline{0}$	inactive process	
	$\mid$	$<a, \lambda>.P$	exp. timed action prefix	$(a \in \mathit{Name}, \lambda \in \mathbb{R}_{>0})$
	$\mid$	$<a, *_w>.P$	passive action prefix	$(a \in \mathit{Name}, w \in \mathbb{R}_{>0})$
	$\mid$	$P + P$	alternative composition	
	$\mid$	$P \parallel_S P$	parallel composition	$(S \subseteq \mathit{Name}_{\mathrm{v}})$
	$\mid$	$P \, / \, H$	hiding	$(H \subseteq \mathit{Name}_{\mathrm{v}})$
	$\mid$	$P \setminus L$	restriction	$(L \subseteq \mathit{Name}_{\mathrm{v}})$
	$\mid$	$P[\varphi]$	relabeling	$(\varphi \in \mathit{Relab})$
	$\mid$	X	process variable	$(X \in \mathit{Var})$
	$\mid$	$\mathrm{rec}\, X : P$	recursion	$(X \in \mathit{Var})$

(process constants are defined by means of equations of the form $B \triangleq P$).

- The duration of $<a, \lambda>$ is the exponentially distributed random variable Exp_λ, where $\Pr\{Exp_\lambda \leq t\} \; = \; 1 - e^{-\lambda \cdot t}$ and $E\{Exp_\lambda\} \; = \; 1 \, / \, \lambda$.
- The choice among exp. timed actions is generative (prob. over arbitrary names) and is solved by applying the race policy (exec. prob. proportional to action rates).
- The duration of $<a, *_w>$ is unspecified (synch. with exponentially timed action).
- The choice among passive actions is reactive (prob. restricted to same name):
  - Probabilistic for passive actions with the same name and solved by applying the preselection policy (exec. prob. proportional to action weights).
  - Nondeterministic for passive actions with different names.
- The choice between an exponentially timed action and a passive action is nondeterministic.

- Applying the race policy to the exponentially timed actions $(\lambda_1, \ldots, \lambda_h)$ enabled by a process term means executing the *fastest* of those actions.
- The sojourn time associated with that term is thus the *minimum* of the random variables quantifying the durations of those actions.
- The sojourn time is exponentially distributed because:

$$\boxed{\min(Exp_{\lambda_1}, \ldots, Exp_{\lambda_h}) \; = \; Exp_{\lambda_1 + \ldots + \lambda_h}}$$

- The average sojourn time is therefore given by $1 \, / \, (\lambda_1 + \ldots + \lambda_h)$.
- The execution probability of exponentially timed action with rate λ_i is $\lambda_i \, / \, (\lambda_1 + \ldots + \lambda_h)$.

- $P_1 + P_2$ behaves as P_1 or P_2 depending on which executes first.
- The choice among several enabled actions is solved by applying either the race policy or the preselection policy.
- The choice is internal if the enabled actions are all invisible, otherwise the choice can be influenced by the external environment.
- $P_1 \parallel_S P_2$ behaves as P_1 in parallel with P_2 under synchronization set S.
- Actions whose name does not belong to S are executed autonomously by P_1 and by P_2 (order determined by race/preselection policy).
- Synchronization is forced between any action enabled by P_1 and any action enabled by P_2 that have the same name belonging to S, in which case the resulting action has the same name as the two original actions ($S = \emptyset$ implies P_1 and P_2 fully independent, $S = Name_{\mathrm{v}}$ implies P_1 and P_2 fully synchronized).

- $\underline{0}$ is a terminated process and hence cannot execute any action.
- $<a, \tilde{\lambda}>.P$ can perform action a at rate $\tilde{\lambda}$ and then behaves as P (action-based sequential composition).
- $P \,/\, H$ behaves as P but every action belonging to H is turned into τ (abstraction mechanism; can be used for preventing a process from communicating).
- $P \setminus L$ behaves as P but every action belonging to L is forbidden (same effect as $P \parallel_L \underline{0}$).
- $P[\varphi]$ behaves as P but every action is renamed according to function φ (redundance avoidance; encoding of the previous two operators if φ is non-visib.-pres./partial).
- Operator precedence: unary operators $> + > \parallel$.
- Operator associativity: $+$ and $\parallel$ are left associative.

- $\mathrm{rec}\, X : P$ behaves as P with every free occurrence of process variable X being replaced by $\mathrm{rec}\, X : P$.

- A process variable is said to occur *free* in a process term if it is not in the scope of a rec binder for that variable, otherwise it is said to be *bound* in that process term.

- A process term is said to be *closed* if all of its occurrences of process variables are bound, otherwise it is said to be *open*.

- A process term is said to be *guarded* iff all of its occurrences of process variables are in the scope of action prefix operators.

- $\mathbb{P}_{\mathrm{M}}$: set of closed and guarded process terms (fully defined, finitely branching).

- **Running example** (MPC syntax):
  - Producer-consumer system: producer, finite buffer, consumer.
  - The producer deposits items into the buffer at rate $\lambda \in \mathbb{R}_{>0}$ as long as the buffer capacity is not exceeded.
  - Stored items are then withdrawn by the consumer at rate $\mu \in \mathbb{R}_{>0}$ according to some predefined discipline (like FIFO or LIFO).
  - Assumption 1: the buffer has only two positions.
  - Assumption 2: identical items, hence the discipline is not important.

- The only observable activities are deposits and withdrawals.
- Names of visible actions: *deposit* and *withdraw*.
- Structure-independent process algebraic description:

$$
\begin{aligned}
ProdCons^{\mathrm{M}}_{0/2} &\triangleq <deposit, \lambda>.ProdCons^{\mathrm{M}}_{1/2} \\
ProdCons^{\mathrm{M}}_{1/2} &\triangleq <deposit, \lambda>.ProdCons^{\mathrm{M}}_{2/2} + \\
& <withdraw, \mu>.ProdCons^{\mathrm{M}}_{0/2} \\
ProdCons^{\mathrm{M}}_{2/2} &\triangleq <withdraw, \mu>.ProdCons^{\mathrm{M}}_{1/2}
\end{aligned}
$$

- Specification to which every correct implementation should conform.

Markovian Process Calculus: Semantics

- State transition graph expressing all computations and branching points and accounting for transition multiplicity ($<a, \lambda>.\underline{0} + <a, \lambda>.\underline{0}$ vs. $<a, \lambda>.\underline{0}$).

- Every $P \in \mathbb{P}_{\mathrm{M}}$ is mapped to a labeled multitransition system $[\![P]\!]_{\mathrm{M}}$:
  - Each state corresponds to a process term into which P can evolve.
  - The initial state corresponds to P.
  - Each transition from a source state to a target state is labeled with the action that determines the corresponding state change.

- Every $P \in \mathbb{P}_{\mathrm{M,pc}}$ is mapped to a CTMC (performance closure if no passive trans.):
  - Dropping action names from all transitions of $[\![P]\!]_{\mathrm{M}}$.
  - Collapsing all the transitions between any two states of $[\![P]\!]_{\mathrm{M}}$ into a single transition by summing up the rates of the original transitions.

- Derivation of one single transition at a time by applying suitable operational semantic rules to the source state of the transition.

- Rules defined by induction on the syntactical structure of process terms.

- Basic rules for action prefix, inductive rules for all the other operators.

- Different formats: dynamic operators (. +), static operators ($\|$ / \ [ ]).

- The multitransition relation $\longrightarrow_{\mathrm{M},P}$ of $[\![P]\!]_{\mathrm{M}}$ is contained in the smallest multiset of elements of $\mathbb{P}_{\mathrm{M}} \times Act_{\mathrm{M}} \times \mathbb{P}_{\mathrm{M}}$ that:
  - Satisfy the operational semantic rules.
  - Keep track of all the possible ways of deriving each transition.

- No rule for $\underline{0}$: $[\![\underline{0}]\!]_{\mathrm{M}}$ has a single state and no transitions.

- Operational semantic rules for action prefix:

$$<a,\lambda>.P \xrightarrow{a,\lambda}_{\mathrm{M}} P$$

$$<a,*_w>.P \xrightarrow{a,*_w}_{\mathrm{M}} P$$

- Operational semantic rules for alternative composition:

$$\frac{P_1 \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P'}{P_1 + P_2 \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P'} \qquad \frac{P_2 \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P'}{P_1 + P_2 \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P'}$$

- Operational semantic rule for recursion:

$$\frac{P\{\mathrm{rec}\, X : P \hookrightarrow X\} \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P'}{\mathrm{rec}\, X : P \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P'} \qquad \left(\frac{B \triangleq P \qquad P \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P'}{B \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P'} \right)$$

- Classical interleaving semantics for parallel composition:
  - *Due to the memoryless property of the exponential distribution, the execution of an exponentially timed action can be thought of as being started in the last state in which the action is enabled.*
  - *Due to the infinite support of the exponential distribution, the probability of simultaneous termination of two concurrent exponentially timed actions is zero.*
- Operational semantic rules for parallel execution:

$$\frac{P_1 \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P_1' \qquad a \notin S}{P_1 \parallel_S P_2 \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P_1' \parallel_S P_2} \qquad \frac{P_2 \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P_2' \qquad a \notin S}{P_1 \parallel_S P_2 \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P_1 \parallel_S P_2'}$$

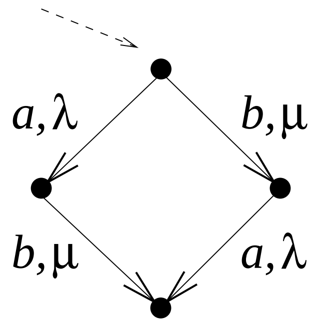
- The following process terms represent structurally different systems:

$$<a, \lambda>.\underline{0} \parallel_{\emptyset} <b, \mu>.\underline{0}$$

$$<a, \lambda>.<b, \mu>.\underline{0} + <b, \mu>.<a, \lambda>.\underline{0}$$

  but they are indistinguishable by an external observer.

- Black-box semantics given by the same labeled multitransition system:



- Interleave concurrent exponentially timed actions without the need of adjusting their rates inside transition labels.

- Synchronization admitted among several actions with the same name, provided that at most one of them is exponentially timed.
- Generative-reactive or reactive-reactive synchronizations.
- The rate of the synchronization of an exponentially timed action with a passive action is given by the rate of the former multiplied by the execution probability of the latter (complies with the bounded capacity assumption).
- Weight of a process term P with respect to passive actions of name a:

$$weight(P, a) \;=\; \sum \{\!| \, w \in \mathbb{R}_{>0} \mid \exists P' \in \mathbb{P}_{\mathrm{M}} .\, P \xrightarrow{a, *_w}_{\mathrm{M}} P' \, |\!\}$$

- Normalizing function for reactive-reactive synchronizations:

$$norm(w_1, w_2, a, P_1, P_2) \;=\; \frac{w_1}{weight(P_1, a)} \cdot \frac{w_2}{weight(P_2, a)} \cdot (weight(P_1, a) + weight(P_2, a))$$

- Operational semantic rules for generative-reactive synchronization:

$$\frac{P_1 \xrightarrow{a,\lambda}_{\mathrm{M}} P'_1 \qquad P_2 \xrightarrow{a,*_w}_{\mathrm{M}} P'_2 \qquad a \in S}{P_1 \parallel_S P_2 \xrightarrow{a,\lambda \cdot \frac{w}{weight(P_2,a)}}_{\mathrm{M}} P'_1 \parallel_S P'_2}$$

$$\frac{P_1 \xrightarrow{a,*_w}_{\mathrm{M}} P'_1 \qquad P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P'_2 \qquad a \in S}{P_1 \parallel_S P_2 \xrightarrow{a,\lambda \cdot \frac{w}{weight(P_1,a)}}_{\mathrm{M}} P'_1 \parallel_S P'_2}$$

- Operational semantic rule for reactive-reactive synchronization:

$$\frac{P_1 \xrightarrow{a,*_{w_1}}_{\mathrm{M}} P'_1 \qquad P_2 \xrightarrow{a,*_{w_2}}_{\mathrm{M}} P'_2 \qquad a \in S}{P_1 \parallel_S P_2 \xrightarrow{a,*_{norm(w_1,w_2,a,P_1,P_2)}}_{\mathrm{M}} P'_1 \parallel_S P'_2}$$

- Operational semantic rules for hiding, restriction, relabeling:

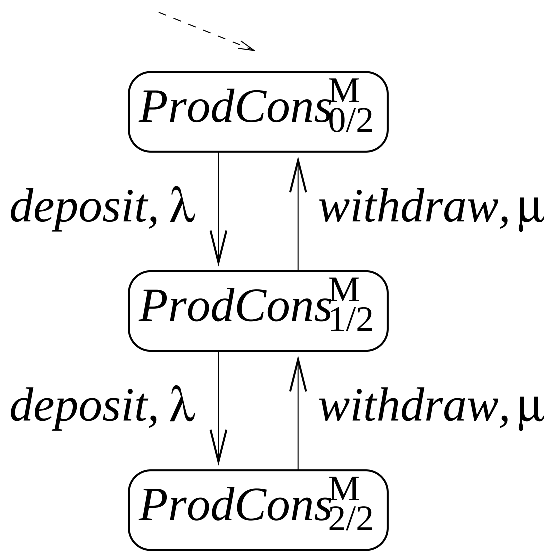
$$\frac{P \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P' \qquad a \in H}{P \,/\, H \xrightarrow{\tau,\tilde{\lambda}}_{\mathrm{M}} P' \,/\, H} \qquad \frac{P \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P' \qquad a \notin H}{P \,/\, H \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P' \,/\, H}$$

$$\frac{P \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P' \qquad a \notin L}{P \setminus L \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P' \setminus L}$$

$$\frac{P \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} P'}{P[\varphi] \xrightarrow{\varphi(a),\tilde{\lambda}}_{\mathrm{M}} P'[\varphi]}$$

- $[\![P]\!]_{\mathrm{M}}$ is finite state if no recursive definition in P contains static ops.

- **Running example** (MPC semantics):
  - Labeled multitransition system $[\![\mathit{ProdCons}^{\mathrm{M}}_{0/2}]\!]_{\mathrm{M}}$ with explicit states:



  - Obtained by mechanically applying the operational semantic rules for process constant, alternative composition, and action prefix.

Part II: The Markovian Spectrum

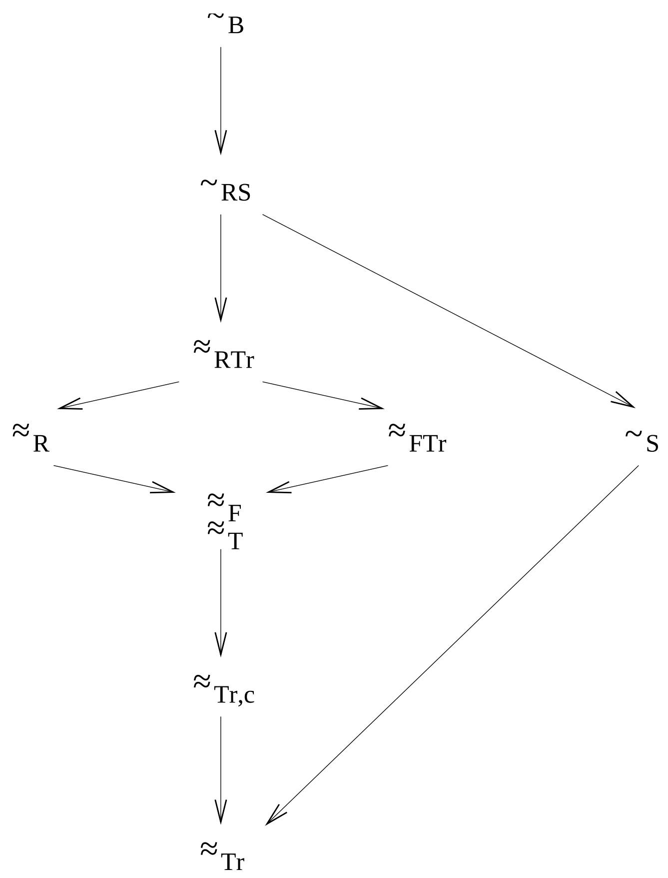
Relating Classical Behavioral Equivalences

- Most studied approaches to the definition of behavioral equivalences:
  - Trace approach: two process terms are equivalent if they are able to execute the same sequences of actions ($\approx_{\mathrm{Tr}}$).
  - Abstraction from branching points leads to *deadlock insensitivity*.
  - Testing approach: two process terms are equivalent if no difference can be discovered when interacting with them by means of tests and comparing their reactions ($\approx_{\mathrm{T}}$).
  - Checking whether *every test* may/must be passed.
  - Bisimulation approach: two process terms are equivalent if they are able to mimic each other's behavior after each action execution ($\sim_{\mathrm{B}}$).
  - Faithful account of branching points leads to *overdiscrimination*.

- Variants of bisimulation equivalence:
  - Simulation equivalence: it is the intersection of two preorders, each of which considers the capability of stepwise behavior mimicking in one single direction ($\sim_{\mathrm{S}}$).
  - Ready-simulation equivalence: same as simulation equivalence, with in addition the fact that each of the two preorders checks for the equality of the sets of actions that are stepwise enabled ($\sim_{\mathrm{RS}}$).
- Less discriminating than bisimulation equivalence.
- The distinctions they make can be traced provided that reasonable operators are included in the process language.

- Deadlock-sensitive variants of trace equivalence:
  - Completed-trace equivalence: it compares process terms also with respect to traces that lead to deadlock ($\approx_{\mathrm{Tr,c}}$).
  - Failure equivalence: it takes into account the set of visible actions that can be refused after executing a trace ($\approx_{\mathrm{F}}$).
  - Ready equivalence: it takes into account the set of visible actions that can be performed after executing a trace ($\approx_{\mathrm{R}}$).
  - Failure-trace equivalence: it takes into account the sets of visible actions that can be refused at each step of a trace ($\approx_{\mathrm{FTr}}$).
  - Ready-trace equivalence: it takes into account the sets of visible actions that can be performed at each step of a trace ($\approx_{\mathrm{RTr}}$).

- Linear-time/branching-time spectrum (for processes without invisible actions):



Relating Markovian Behavioral Equivalences

- Markovian linear-time/branching-time spectrum:

$$\sim_{\rm MB} = \sim_{\rm MRS} = \sim_{\rm MS} \subset \sim_{\rm MRTr} = \sim_{\rm MFTr} \subset \sim_{\rm MR} = \sim_{\rm MF} = \sim_{\rm MT} \subset \sim_{\rm MCTr} = \sim_{\rm MTr}$$

- Similar to the probabilistic spectrum.

- More linear than the nondeterministic spectrum.

- The considered processes do not exhibit nondeterministic behavior.

Part III:
Properties of Markovian Behavioral Equivalences

Comparison Criteria

1. Discriminating power:
   which of them is finer/coarser than the others?

2. Congruence:
   do they support compositional reasoning?

3. Sound and complete axiomatization:
   what are their fundamental equational laws?

4. Modal logic characterization:
   what behavioral properties do they preserve?

5. Complexity (of their verification algorithms):
   can they be checked for efficiently?

6. Exactness (of their induced CTMC-level aggregations):
   do they make sense from a performance viewpoint?

- Congruence enables the equivalence-based compositional reduction of models obtained as the combination of submodels.

- Axioms can be used as rewriting rules that syntactically manipulate models in a way that is consistent with the equivalence.

- The modal logic characterization provides diagnostic information in the form of distinguishing formulas that explain model inequivalence.

- *Exactness*: the probability of being in a macrostate of an aggregated CTMC is the sum of the probabilities of being in one of the constituent microstates of the original CTMC (transient/stationary).

- Exactness guarantees the preservation of performance characteristics when going from the original CTMC to the aggregated one induced by the equivalence.

Markovian Bisimulation Equivalence

- Two process terms are equivalent if they are able to mimic each other's functional and performance behavior stepwise.

- Whenever a process term can perform actions with a certain name that reach a certain set of terms at a certain speed, then any process term equivalent to the given one has to be able to respond with actions with the same name that reach an equivalent set of terms at the same speed.

- Comparison of process term exit rates rather than individual transitions (different from bisimulation equivalence for nondeterministic processes).

- High sensitivity to the branching structure of process terms.

- The exit rate of a process term is the rate at which the process term can execute actions of a given name that lead to a given set of terms (sum of the rates of those actions due to the race policy).

- Exit rate at which $P \in \mathbb{P}_{\mathrm{M}}$ executes actions of name $a \in \mathit{Name}$ and level $l \in \{0, -1\}$ (0 exp. timed, -1 passive) that lead to destination $D \subseteq \mathbb{P}_{\mathrm{M}}$:

$$\boxed{\mathit{rate}_{\mathrm{e}}(P, a, l, D) \; = \; \begin{cases} \sum\{\!| \, \lambda \in \mathbb{R}_{>0} \mid \exists P' \in D. \, P \xrightarrow{a, \lambda}_{\mathrm{M}} P' \, |\!\} & \text{if } l = 0 \\ \sum\{\!| \, w \in \mathbb{R}_{>0} \mid \exists P' \in D. \, P \xrightarrow{a, *_w}_{\mathrm{M}} P' \, |\!\} & \text{if } l = -1 \end{cases}}$$

- Overall exit rate of P w.r.t. a at level l: $\mathit{rate}_{\mathrm{o}}(P, a, l) = \mathit{rate}_{\mathrm{e}}(P, a, l, \mathbb{P}_{\mathrm{M}})$.

- Total exit rate of P at level l: $\mathit{rate}_{\mathrm{t}}(P, l) \; = \sum\limits_{a \in \mathit{Name}} \mathit{rate}_{\mathrm{o}}(P, a, l)$.

- $1 \, / \, \mathit{rate}_{\mathrm{t}}(P, 0)$ is the average sojourn time of P when $P \in \mathbb{P}_{\mathrm{M,pc}}$.

- The exit probability of a process term is the probability with which the process term can execute actions of a given name that lead to a given set of terms.

- Generative probability for exponentially timed actions (arbitrary names).

- Reactive probability for passive actions (restriction to same name).

- Exit probability with which $P \in \mathbb{P}_{\mathrm{M}}$ executes actions of name $a \in \mathit{Name}$ and level $l \in \{0, -1\}$ that lead to destination $D \subseteq \mathbb{P}_{\mathrm{M}}$:

$$\boxed{\mathit{prob}_{\mathrm{e}}(P, a, l, D) \; = \; \begin{cases} \mathit{rate}_{\mathrm{e}}(P, a, l, D) \, / \, \mathit{rate}_{\mathrm{t}}(P, l) & \text{if } l = 0 \\ \mathit{rate}_{\mathrm{e}}(P, a, l, D) \, / \, \mathit{rate}_{\mathrm{o}}(P, a, l) & \text{if } l = -1 \end{cases}}$$

- An equivalence relation $\mathcal{B}$ over $\mathbb{P}_{\mathrm{M}}$ is a Markovian bisimulation iff, whenever $(P_1, P_2) \in \mathcal{B}$, then for all action names $a \in Name$, levels $l \in \{0, -1\}$, and equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\mathcal{B}$:

$$\boxed{rate_{\mathrm{e}}(P_1, a, l, D) \; = \; rate_{\mathrm{e}}(P_2, a, l, D)}$$

- Markovian bisimulation equivalence $\sim_{\mathrm{MB}}$ is the union of all the Markovian bisimulations.

- *A consequence of the coinductive nature of* $\sim_{\mathrm{MB}}$ *is that the derivatives of two equivalent terms are still equivalent.*

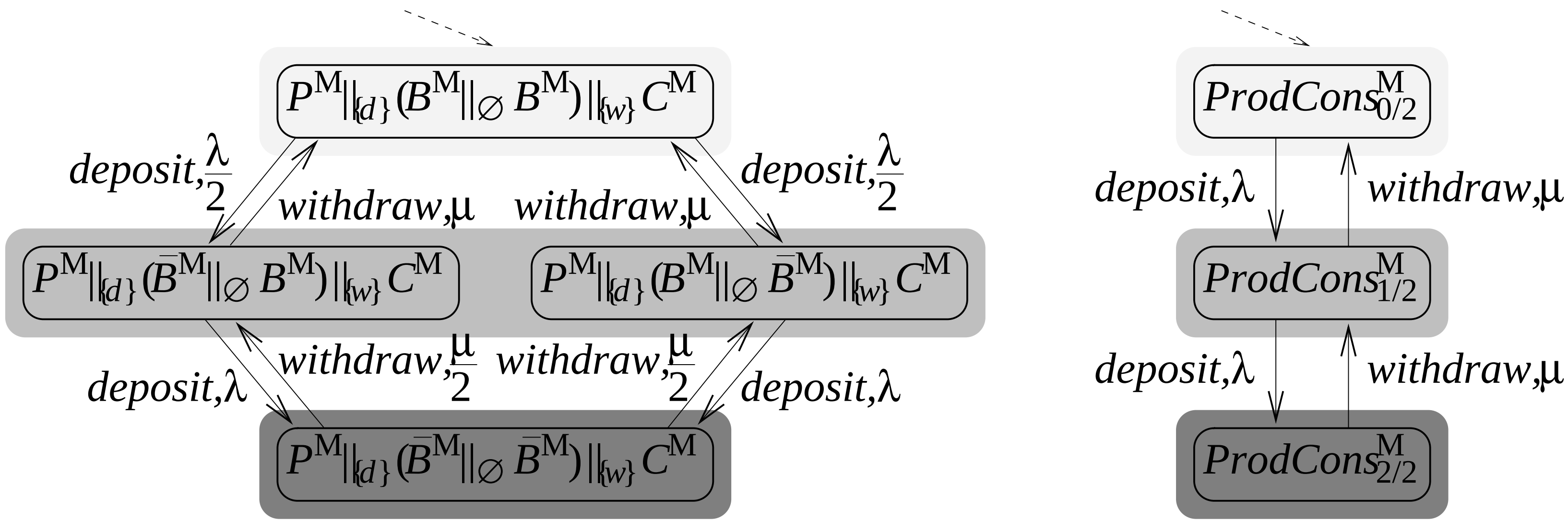
- **Running example** ($\sim_{\rm MB}$):

  - Concurrent implementation with two independent one-pos. buffers:

$$\begin{array}{rcl}
PC^{\rm M}_{\rm conc,2} & \triangleq & \mathit{Prod}^{\rm M} \parallel_{\{\mathit{deposit}\}} (\mathit{Buff}^{\rm M} \parallel_{\emptyset} \mathit{Buff}^{\rm M}) \parallel_{\{\mathit{withdraw}\}} \mathit{Cons}^{\rm M} \\
\mathit{Prod}^{\rm M} & \triangleq & <\mathit{deposit}, \lambda>.\mathit{Prod}^{\rm M} \\
\mathit{Buff}^{\rm M} & \triangleq & <\mathit{deposit}, *_1>.<\mathit{withdraw}, *_1>.\mathit{Buff}^{\rm M} \\
\mathit{Cons}^{\rm M} & \triangleq & <\mathit{withdraw}, \mu>.\mathit{Cons}^{\rm M}
\end{array}$$

  - All the actions occurring in the buffer are passive, consistent with the fact that the buffer is a passive entity.
  - Is $PC^{\rm M}_{\rm conc,2}$ a correct implementation of $\mathit{ProdCons}^{\rm M}_{0/2}$?
  - Yes, because it turns out that $PC^{\rm M}_{\rm conc,2} \sim_{\rm MB} \mathit{ProdCons}^{\rm M}_{0/2}$.
  - Proved by finding a suitable Markovian bisimulation.

- Markovian bisimulation proving $PC^{\mathrm{M}}_{\mathrm{conc},2} \sim_{\mathrm{MB}} ProdCons^{\mathrm{M}}_{0/2}$, with states of the same color belonging to the same equivalence class:



- The initial state on the left-hand side has both outgoing transitions labeled with $\lambda/2$, not λ.
- The bottom state on the left-hand side has both outgoing transitions labeled with $\mu/2$, not μ.

- In order for $P_1 \sim_{\mathrm{MB}} P_2$, it is necessary that for all $a \in \mathit{Name}$ and $l \in \{0, -1\}$:

$$\boxed{\mathit{rate}_{\mathrm{o}}(P_1, a, l) \; = \; \mathit{rate}_{\mathrm{o}}(P_2, a, l)}$$

- A binary relation $\mathcal{B}$ over $\mathbb{P}_{\mathrm{M}}$ is a Markovian bisimulation up to $\sim_{\mathrm{MB}}$ iff, whenever $(P_1, P_2) \in \mathcal{B}$, then for all action names $a \in \mathit{Name}$, levels $l \in \{0, -1\}$, and equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/(\mathcal{B} \cup \mathcal{B}^{-1} \cup \sim_{\mathrm{MB}})^+$:

$$\boxed{\mathit{rate}_{\mathrm{e}}(P_1, a, l, D) \; = \; \mathit{rate}_{\mathrm{e}}(P_2, a, l, D)}$$

- Focus on important pairs of process terms that form a bisimulation.

- In order for $P_1 \sim_{\mathrm{MB}} P_2$, it is sufficient to find a Markovian bisimulation up to $\sim_{\mathrm{MB}}$ that contains (P_1, P_2).

- $\sim_{\mathrm{MB}}$ has an alternative characterization in which time and probability are kept separate (instead of being both subsumed by rates).

- An equivalence relation $\mathcal{B}$ over $\mathbb{P}_{\mathrm{M}}$ is a separate Markovian bisimulation iff, whenever $(P_1, P_2) \in \mathcal{B}$, then for all action names $a \in \mathit{Name}$ and levels $l \in \{0, -1\}$:

$$\boxed{\mathit{rate}_{\mathrm{o}}(P_1, a, l) \; = \; \mathit{rate}_{\mathrm{o}}(P_2, a, l)}$$

  and for all equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\mathcal{B}$:

$$\boxed{\mathit{prob}_{\mathrm{e}}(P_1, a, l, D) \; = \; \mathit{prob}_{\mathrm{e}}(P_2, a, l, D)}$$

- Separate Markovian bisimulation equivalence $\sim_{\mathrm{MB,s}}$ is the union of all the separate Markovian bisimulations.

- For all $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$:

$$\boxed{P_1 \sim_{\mathrm{MB,s}} P_2 \iff P_1 \sim_{\mathrm{MB}} P_2}$$

- $\sim_{\rm MB}$ is a congruence with respect to all the dynamic and static operators as well as recursion.

- Let $P_1, P_2 \in \mathbb{P}_{\rm M}$. Whenever $P_1 \sim_{\rm MB} P_2$, then:

$$\boxed{\begin{array}{c}
<a, \tilde{\lambda}>.P_1 \;\sim_{\rm MB}\; <a, \tilde{\lambda}>.P_2 \\
P_1 + P \;\sim_{\rm MB}\; P_2 + P \qquad\qquad P + P_1 \;\sim_{\rm MB}\; P + P_2 \\
P_1 \,\|_S\, P \;\sim_{\rm MB}\; P_2 \,\|_S\, P \qquad\qquad P \,\|_S\, P_1 \;\sim_{\rm MB}\; P \,\|_S\, P_2 \\
P_1 \,/\, H \;\sim_{\rm MB}\; P_2 \,/\, H \\
P_1 \setminus L \;\sim_{\rm MB}\; P_2 \setminus L \\
P_1[\varphi] \;\sim_{\rm MB}\; P_2[\varphi]
\end{array}}$$

- Recursion: extend $\sim_{\mathrm{MB}}$ to open process terms by replacing all variables freely occurring outside rec binders with every closed process term.

- Let $P_1, P_2 \in \mathcal{PL}_{\mathrm{M}}$ be guarded process terms containing free occurrences of $k \in \mathbb{N}$ process variables $X_1, \ldots, X_k \in \mathit{Var}$ at most.

- We define $P_1 \sim_{\mathrm{MB}} P_2$ iff:

$$\boxed{P_1\{Q_i \hookrightarrow X_i \mid 1 \leq i \leq k\} \ \sim_{\mathrm{MB}} \ P_2\{Q_i \hookrightarrow X_i \mid 1 \leq i \leq k\}}$$

  for all $Q_1, \ldots, Q_k \in \mathbb{P}_{\mathrm{M}}$:

- Whenever $P_1 \sim_{\mathrm{MB}} P_2$, then:

$$\boxed{\mathrm{rec}\, X : P_1 \ \sim_{\mathrm{MB}} \ \mathrm{rec}\, X : P_2}$$

- $\sim_{\rm MB}$ has a sound and complete axiomatization over the set $\mathbb{P}_{\rm M,nrec}$ of nonrecursive process terms of $\mathbb{P}_{\rm M}$.

- Basic laws (commutativity, associativity, and neutral element of $+$):

$$\boxed{\begin{array}{lrcl}
(\mathcal{X}_{\rm MB,1}) & P_1 + P_2 & = & P_2 + P_1 \\
(\mathcal{X}_{\rm MB,2}) & (P_1 + P_2) + P_3 & = & P_1 + (P_2 + P_3) \\
(\mathcal{X}_{\rm MB,3}) & P + \underline{0} & = & P
\end{array}}$$

- Characterizing laws (race policy and preselection policy, instead of $+$ idempotency):

$$\boxed{\begin{array}{lrcl}
(\mathcal{X}_{\rm MB,4}) & {<}a, \lambda_1{>}.P + {<}a, \lambda_2{>}.P & = & {<}a, \lambda_1 + \lambda_2{>}.P \\
(\mathcal{X}_{\rm MB,5}) & {<}a, *_{w_1}{>}.P + {<}a, *_{w_2}{>}.P & = & {<}a, *_{w_1 + w_2}{>}.P
\end{array}}$$

- Expansion law (interl. view of conc. supported by mem. prop.; I, J nonempty and finite):

$$(\mathcal{X}_{\mathrm{MB},6}) \quad \sum_{i \in I} <a_i, \tilde{\lambda}_i>.P_i \parallel_S \sum_{j \in J} <b_j, \tilde{\mu}_j>.Q_j \;=\;$$

$$\sum_{k \in I, a_k \notin S} <a_k, \tilde{\lambda}_k>.\left(P_k \parallel_S \sum_{j \in J} <b_j, \tilde{\mu}_j>.Q_j\right) \;+$$

$$\sum_{h \in J, b_h \notin S} <b_h, \tilde{\mu}_h>.\left(\sum_{i \in I} <a_i, \tilde{\lambda}_i>.P_i \parallel_S Q_h\right) \;+$$

$$\sum_{k \in I, a_k \in S, \tilde{\lambda}_k \in \mathbb{R}_{>0}} \; \sum_{h \in J, b_h = a_k, \tilde{\mu}_h = *_{w_h}} <a_k, \tilde{\lambda}_k \cdot \frac{w_h}{weight(Q, b_h)}>.(P_k \parallel_S Q_h) \;+$$

$$\sum_{h \in J, b_h \in S, \tilde{\mu}_h \in \mathbb{R}_{>0}} \; \sum_{k \in I, a_k = b_h, \tilde{\lambda}_k = *_{v_k}} <b_h, \tilde{\mu}_h \cdot \frac{v_k}{weight(P, a_k)}>.(P_k \parallel_S Q_h) \;+$$

$$\sum_{k \in I, a_k \in S, \tilde{\lambda}_k = *_{v_k}} \; \sum_{h \in J, b_h = a_k, \tilde{\mu}_h = *_{w_h}} <a_k, *_{norm(v_k, w_h, a_k, P, Q)}>.(P_k \parallel_S Q_h)$$

$$(\mathcal{X}_{\mathrm{MB},7}) \quad \sum_{i \in I} <a_i, \tilde{\lambda}_i>.P_i \parallel_S \underline{0} \;=\; \sum_{k \in I, a_k \notin S} <a_k, \tilde{\lambda}_k>.P_k$$

$$(\mathcal{X}_{\mathrm{MB},8}) \quad \underline{0} \parallel_S \sum_{j \in J} <b_j, \tilde{\mu}_j>.Q_j \;=\; \sum_{h \in J, b_h \notin S} <b_h, \tilde{\mu}_h>.Q_h$$

$$(\mathcal{X}_{\mathrm{MB},9}) \quad \underline{0} \parallel_S \underline{0} \;=\; \underline{0}$$

- Distribution laws (for unary static operators):

$$
\begin{array}{lrcll}
(\mathcal{X}_{\mathrm{MB},10}) & \underline{0} \,/\, H & = & \underline{0} & \\
(\mathcal{X}_{\mathrm{MB},11}) & (<a, \tilde{\lambda}>.P) \,/\, H & = & <\tau, \tilde{\lambda}>.(P \,/\, H) & \text{if } a \in H \\
(\mathcal{X}_{\mathrm{MB},12}) & (<a, \tilde{\lambda}>.P) \,/\, H & = & <a, \tilde{\lambda}>.(P \,/\, H) & \text{if } a \notin H \\
(\mathcal{X}_{\mathrm{MB},13}) & (P_1 + P_2) \,/\, H & = & P_1 \,/\, H + P_2 \,/\, H & \\
 & & & & \\
(\mathcal{X}_{\mathrm{MB},14}) & \underline{0} \setminus L & = & \underline{0} & \\
(\mathcal{X}_{\mathrm{MB},15}) & (<a, \tilde{\lambda}>.P) \setminus L & = & \underline{0} & \text{if } a \in L \\
(\mathcal{X}_{\mathrm{MB},16}) & (<a, \tilde{\lambda}>.P) \setminus L & = & <a, \tilde{\lambda}>.(P \setminus L) & \text{if } a \notin L \\
(\mathcal{X}_{\mathrm{MB},17}) & (P_1 + P_2) \setminus L & = & P_1 \setminus L + P_2 \setminus L & \\
 & & & & \\
(\mathcal{X}_{\mathrm{MB},18}) & \underline{0}[\varphi] & = & \underline{0} & \\
(\mathcal{X}_{\mathrm{MB},19}) & (<a, \tilde{\lambda}>.P)[\varphi] & = & <\varphi(a), \tilde{\lambda}>.(P[\varphi]) & \\
(\mathcal{X}_{\mathrm{MB},20}) & (P_1 + P_2)[\varphi] & = & P_1[\varphi] + P_2[\varphi] &
\end{array}
$$

- $DED(\mathcal{X}_{\text{MB}})$: deduction system based on all the previous axioms plus:
  - ⊙ Reflexivity: $\mathcal{X}_{\text{MB}} \vdash P = P$.
  - ⊙ Symmetry: $\mathcal{X}_{\text{MB}} \vdash P_1 = P_2 \implies \mathcal{X}_{\text{MB}} \vdash P_2 = P_1$.
  - ⊙ Transitivity: $\mathcal{X}_{\text{MB}} \vdash P_1 = P_2 \wedge \mathcal{X}_{\text{MB}} \vdash P_2 = P_3 \implies \mathcal{X}_{\text{MB}} \vdash P_1 = P_3$.
  - ⊙ Substitutivity: $\mathcal{X}_{\text{MB}} \vdash P_1 = P_2 \implies \mathcal{X}_{\text{MB}} \vdash {<}a, \tilde{\lambda}{>}.P_1 = {<}a, \tilde{\lambda}{>}.P_2 \wedge \ldots$

- The deduction system $DED(\mathcal{X}_{\text{MB}})$ is sound and complete for $\sim_{\text{MB}}$ over $\mathbb{P}_{\text{M,nrec}}$; i.e., for all $P_1, P_2 \in \mathbb{P}_{\text{M,nrec}}$:

$$\boxed{\mathcal{X}_{\text{MB}} \vdash P_1 = P_2 \iff P_1 \sim_{\text{MB}} P_2}$$

- $\sim_{\rm MB}$ has a modal logic characterization based on a variant of the Hennessy-Milner logic.
- Basic truth values and propositional connectives, plus modal operators expressing how to behave after executing actions with certain names.
- Diamond operator decorated with a lower bound on the rate/weight with which exponentially timed/passive actions with the given name should be executed (consistent with capturing step-by-step behavior mimicking).
- Syntax of the modal language $\mathcal{ML}_{\rm MB}$ ($a \in Name$, $\lambda, w \in \mathbb{R}_{>0}$):

ϕ	$::=$	true	basic truth value
	$\mid$	$\neg \phi$	negation
	$\mid$	$\phi \wedge \phi$	conjunction
	$\mid$	$\langle a \rangle_{\lambda} \phi$	exponentially timed possibility
	$\mid$	$\langle a \rangle_{*_w} \phi$	passive possibility

- Interpretation of $\mathcal{ML}_{\mathrm{MB}}$ over $\mathbb{P}_{\mathrm{M}}$:

$$
\boxed{
\begin{array}{llll}
P & \models_{\mathrm{MB}} & \mathrm{true} & \\
P & \models_{\mathrm{MB}} & \neg\phi & \text{if } P \not\models_{\mathrm{MB}} \phi \\
P & \models_{\mathrm{MB}} & \phi_1 \wedge \phi_2 & \text{if } P \models_{\mathrm{MB}} \phi_1 \text{ and } P \models_{\mathrm{MB}} \phi_2 \\
P & \models_{\mathrm{MB}} & \langle a \rangle_{\lambda} \phi & \text{if } \mathit{rate}_{\mathrm{e}}(P, a, 0, \mathit{sat}(\phi)) \geq \lambda \\
P & \models_{\mathrm{MB}} & \langle a \rangle_{*_w} \phi & \text{if } \mathit{rate}_{\mathrm{e}}(P, a, -1, \mathit{sat}(\phi)) \geq w
\end{array}
}
$$

where:

$$
\mathit{sat}(\phi) \; = \; \{P' \in \mathbb{P}_{\mathrm{M}} \mid P' \models_{\mathrm{MB}} \phi\}
$$

- For all $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$:

$$
\boxed{P_1 \sim_{\mathrm{MB}} P_2 \iff (\forall \phi \in \mathcal{ML}_{\mathrm{MB}}.\, P_1 \models_{\mathrm{MB}} \phi \iff P_2 \models_{\mathrm{MB}} \phi)}
$$

- $\sim_{\mathrm{MB}}$ is decidable in polynomial time over the set $\mathbb{P}_{\mathrm{M,fin}}$ of finite-state process terms of $\mathbb{P}_{\mathrm{M}}$: Paige-Tarjan partition refinement algorithm.

- Based on the fact that $\sim_{\mathrm{MB}}$ can be characterized as the limit of a sequence of successively finer equivalence relations:

$$\boxed{\sim_{\mathrm{MB}} \; = \; \bigcap_{i \in \mathbb{N}} \sim_{\mathrm{MB},i}}$$

- $\sim_{\mathrm{MB},0} \; = \; \mathbb{P}_{\mathrm{M}} \times \mathbb{P}_{\mathrm{M}}$ hence it induces the trivial partition $\{\mathbb{P}_{\mathrm{M}}\}$.

- Whenever $P_1 \sim_{\mathrm{MB},i} P_2$, $i \in \mathbb{N}_{\geq 1}$, then for all $a \in \mathit{Name}$, $l \in \{0, -1\}$, and $D \in \mathbb{P}_{\mathrm{M}} / \sim_{\mathrm{MB},i-1}$:

$$\mathit{rate}_{\mathrm{e}}(P_1, a, l, D) \; = \; \mathit{rate}_{\mathrm{e}}(P_2, a, l, D)$$

- $\sim_{\mathrm{MB},1}$ refines $\{\mathbb{P}_{\mathrm{M}}\}$ by creating an equivalence class for each set of process terms that satisfy the necessary condition for $\sim_{\mathrm{MB}}$.

- Steps of the algorithm for checking whether $P_1 \sim_{\mathrm{MB}} P_2$:

  1. Build an initial partition with a single class including all the states of $[\![P_1]\!]_{\mathrm{M}}$ and $[\![P_2]\!]_{\mathrm{M}}$.
  2. Initialize a list of splitters with the above class as its only element.
  3. While the list of splitters is not empty, select a splitter and remove it from the list after refining the current partition for each $a \in \mathit{Name}_{P_1,P_2}$ and $l \in \{0,-1\}$:
     a. Split each class of the current partition by comparing the exit rates of its states when performing actions of name a and level l that lead to the selected splitter.
     b. For each class that has been split, insert into the list of splitters all the resulting subclasses except for the largest one.
  4. Return yes/no depending on whether the initial states of $[\![P_1]\!]_{\mathrm{M}}$ and $[\![P_2]\!]_{\mathrm{M}}$ belong to the same class of the final partition or not.

- The time complexity is $O(m \cdot \log n)$ if a *splay tree* is used for representing the subclasses arising from the splitting of a class (they can be more than two).

- $\sim_{\mathrm{MB}}$ induces an exact aggregation known as ordinary lumping.

- A partition $\mathcal{O}$ of the state space of a CTMC is an ordinary lumping iff, whenever $s_1, s_2 \in O$ for some $O \in \mathcal{O}$, then for all $O' \in \mathcal{O}$:

$$\boxed{\begin{array}{r} \sum\{\!| \, \lambda \in \mathbb{R}_{>0} \mid \exists s' \in O' . \, s_1 \xrightarrow{\lambda} s' \, |\!\} \; = \\ \sum\{\!| \, \lambda \in \mathbb{R}_{>0} \mid \exists s' \in O' . \, s_2 \xrightarrow{\lambda} s' \, |\!\} \end{array}}$$

- The probability of being in a macrostate of an ordinarily lumped CTMC is the sum of the probabilities of being in one of its constituent microstates of the original CTMC.

- Two Markovian bisimilar process terms in $\mathbb{P}_{\mathrm{M,pc}}$ are guaranteed to possess the same performance characteristics.

Markovian Testing Equivalence

- Two process terms are equivalent if an external observer cannot distinguish between them, with the only way for the observer to infer information about their functional and performance behavior being to interact with them by means of tests and compare their reactions.

- Was the test passed?
  If so, with which probability?
  And how long did it take to pass the test?

- Tests formalized as process terms.

- Interaction formalized as parallel composition of process term and test with synchronization enforced on any visible action name.

- Comparison of process term probabilities of performing successful test-driven computations within arbitrary time upper bounds.

- A computation of a process term $P \in \mathbb{P}_{\mathrm{M}}$ is a sequence of transitions that can be executed starting from P.
- The *length* of a computation is given by the number of its transitions.
- $\mathcal{C}_{\mathrm{f}}(P)$: multiset of finite-length computations of P.
- Two distinct computations are *independent* of each other iff neither is a proper prefix of the other one.
- Focus on finite multisets of independent, finite-length computations.
- Attributes of a finite-length computation:
  - ⊙ *Trace*.
  - ⊙ *Probability*.
  - ⊙ *Duration*.

- Given a set of sequences, we use:
  - Operator $\_ \circ \_$ for sequence concatenation.
  - Operator $|\_|$ for sequence length.
- The concrete trace associated with the execution of $c \in \mathcal{C}_{\mathrm{f}}(P)$ is the sequence of action names labeling the transitions of c:

$$\boxed{\mathit{trace}_{\mathrm{c}}(c) \; = \begin{cases} \varepsilon & \text{if } |c| = 0 \\ a \circ \mathit{trace}_{\mathrm{c}}(c') & \text{if } c \equiv P \xrightarrow{a,\tilde{\lambda}}_{\mathrm{M}} c' \end{cases}}$$

- We denote by $\mathit{trace}(c)$ the visible part of $\mathit{trace}_{\mathrm{c}}(c)$, i.e., the subsequence of $\mathit{trace}_{\mathrm{c}}(c)$ obtained by removing all the occurrences of τ.

- For the quantitative attributes, we assume $P \in \mathbb{P}_{\mathrm{M,pc}}$.

- The probability of executing $c \in \mathcal{C}_{\mathrm{f}}(P)$ is the product of the execution probabilities of the transitions of c:

$$\boxed{\mathit{prob}(c) \;=\; \begin{cases} 1 & \text{if } |c| = 0 \\ \frac{\lambda}{\mathit{rate}_{\mathrm{t}}(P,0)} \cdot \mathit{prob}(c') & \text{if } c \equiv P \xrightarrow{a,\lambda}_{\mathrm{M}} c' \end{cases}}$$

- Probability of executing a computation in $C \subseteq \mathcal{C}_{\mathrm{f}}(P)$:

$$\boxed{\mathit{prob}(C) \;=\; \sum_{c \in C} \mathit{prob}(c)}$$

assuming that C is finite and all of its computations are independent.

- The stepwise average duration of $c \in \mathcal{C}_{\mathrm{f}}(P)$ is the sequence of average sojourn times in the states traversed by c:

$$\boxed{\mathit{time}_{\mathrm{a}}(c) \;=\; \begin{cases} \varepsilon & \text{if } |c| = 0 \\ \frac{1}{\mathit{rate}_{\mathrm{t}}(P,0)} \circ \mathit{time}_{\mathrm{a}}(c') & \text{if } c \equiv P \xrightarrow{a,\lambda}_{\mathrm{M}} c' \end{cases}}$$

- Multiset of computations in $C \subseteq \mathcal{C}_{\mathrm{f}}(P)$ whose stepwise average duration is not greater than $\theta \in (\mathbb{R}_{>0})^*$:

$$\boxed{C_{\leq \theta} \;=\; \{\!| \, c \in C \mid |c| \leq |\theta| \wedge \forall i = 1, \dots, |c|. \, \mathit{time}_{\mathrm{a}}(c)[i] \leq \theta[i] \, |\!\}}$$

- C^l: multiset of computations in $C \subseteq \mathcal{C}_{\mathrm{f}}(P)$ having length $l \in \mathbb{N}$.

- The stepwise duration of $c \in \mathcal{C}_{\mathrm{f}}(P)$ is the sequence of random variables quantifying the sojourn times in the states traversed by c:

$$\boxed{time_{\mathrm{d}}(c) \;=\; \begin{cases} \varepsilon & \text{if } |c| = 0 \\ Exp_{rate_{\mathrm{t}}(P,0)} \circ time_{\mathrm{d}}(c') & \text{if } c \equiv P \xrightarrow{a,\lambda}_{\mathrm{M}} c' \end{cases}}$$

- Probability distribution of executing a computation in $C \subseteq \mathcal{C}_{\mathrm{f}}(P)$ within a sequence $\theta \in (\mathbb{R}_{>0})^*$ of time units:

$$\boxed{prob_{\mathrm{d}}(C, \theta) \;=\; \sum_{c \in C}^{|c| \le |\theta|} prob(c) \cdot \prod_{i=1}^{|c|} \Pr\{time_{\mathrm{d}}(c)[i] \le \theta[i]\}}$$

  assuming that C is finite and all of its computations are independent.

- Factor $\Pr\{time_{\mathrm{d}}(c)[i] \;\le\; \theta[i]\} \;=\; 1 - \mathrm{e}^{-\theta[i]/time_{\mathrm{a}}(c)[i]}$ stems from the cumulative distribution function of the exponentially distributed random variable $time_{\mathrm{d}}(c)[i]$ (whose expected value is $time_{\mathrm{a}}(c)[i]$).

- Why not summing up sojourn times? (standard duration instead of stepwise one)

- Consider process terms ($\lambda \neq \mu$, $b \neq d$, identical nonmaximal computations):

$$<g, \gamma>.<a, \lambda>.<b, \mu>.\underline{0} + <g, \gamma>.<a, \mu>.<d, \lambda>.\underline{0}$$
$$<g, \gamma>.<a, \lambda>.<d, \mu>.\underline{0} + <g, \gamma>.<a, \mu>.<b, \lambda>.\underline{0}$$

- Maximal computations of the first term:

$$c_{1,1} \equiv \;\; . \xrightarrow{g,\gamma}_{\mathrm{M}} . \xrightarrow{a,\lambda}_{\mathrm{M}} . \xrightarrow{b,\mu}_{\mathrm{M}} .$$
$$c_{1,2} \equiv \;\; . \xrightarrow{g,\gamma}_{\mathrm{M}} . \xrightarrow{a,\mu}_{\mathrm{M}} . \xrightarrow{d,\lambda}_{\mathrm{M}} .$$

- Maximal computations of the second term:

$$c_{2,1} \equiv \;\; . \xrightarrow{g,\gamma}_{\mathrm{M}} . \xrightarrow{a,\lambda}_{\mathrm{M}} . \xrightarrow{d,\mu}_{\mathrm{M}} .$$
$$c_{2,2} \equiv \;\; . \xrightarrow{g,\gamma}_{\mathrm{M}} . \xrightarrow{a,\mu}_{\mathrm{M}} . \xrightarrow{b,\lambda}_{\mathrm{M}} .$$

- Same sum of average sojourn times $\frac{1}{2 \cdot \gamma} + \frac{1}{\lambda} + \frac{1}{\mu}$ and $\frac{1}{2 \cdot \gamma} + \frac{1}{\mu} + \frac{1}{\lambda}$ but ...

- *. . . an external observer would be able to distinguish between the two terms by taking note of the instants at which the actions are performed.*

- Comparing probabilities of passing a test within a time upper bound.
- Syntax of the set $\mathbb{T}_{\mathrm{R}}$ of reactive tests ($a \in Name_{\mathrm{v}}$, $w \in \mathbb{R}_{>0}$):

$$\boxed{\begin{array}{lcl} T & ::= & \mathrm{s} \mid T' \\ T' & ::= & <a, *_w>.T \mid T' + T' \end{array}}$$

- Asymmetric action synchronization: only passive actions within tests.
- Performance closure: passive τ-actions not admitted within tests.
- Presence of a time upper bound: recursion not necessary within tests.
- Denoting test passing: zeroary success operator s (success action may interfere).
- Avoiding ambiguous tests like $\mathrm{s} + T$: two-level syntax for tests.

- Interaction system of $P \in \mathbb{P}_{\mathrm{M,pc}}$ and $T \in \mathbb{T}_{\mathrm{R}}$:

$$\boxed{P \,\|_{Name_{\mathrm{v}}}\, T \;\in\; \mathbb{P}_{\mathrm{M,pc}}}$$

- In any of its states, P generates the proposal of an action to be executed by means of a race among the exponentially timed actions enabled in that state.

- If the name of the proposed action is τ, then P advances by itself.

- Otherwise T:

  - ⊙ either reacts by participating in the interaction with P through a passive action having the same name;

  - ⊙ or blocks the interaction if it has no passive actions with the proposed name.

- Let $P \in \mathbb{P}_{\mathrm{M,pc}}$ and $T \in \mathbb{T}_{\mathrm{R}}$:
  - A *configuration* is a state of $[\![P \parallel_{Name_{\mathrm{v}}} T]\!]_{\mathrm{M}}$.
  - A *test-driven computation* is a computation of $[\![P \parallel_{Name_{\mathrm{v}}} T]\!]_{\mathrm{M}}$.
  - A configuration is formed by *process projection* and *test projection*.
  - A configuration is *successful* iff its test projection is s.
  - A test-driven computation is *successful* iff it traverses a successful configuration.
  - $\mathcal{SC}(P, T)$: multiset of successful computations of $P \parallel_{Name_{\mathrm{v}}} T$.

- If P has no exponentially timed τ-actions:
  - All the computations in $\mathcal{SC}(P, T)$ have a finite length due to the restrictions imposed on the test syntax.
  - All the computations in $\mathcal{SC}(P, T)$ are independent of each other because of their maximality.
  - The multiset $\mathcal{SC}(P, T)$ is finite because both P and T are finitely branching.
- Same considerations for $\mathcal{SC}_{\leq \theta}(P, T)$.
- If there are exponentially timed τ-actions:
  - Are the computations in $\mathcal{SC}_{\leq \theta}(P, T)$ independent of each other?
  - How to distinguish among process terms having only exponentially timed τ-actions, like $<\tau, \lambda>.\underline{0}$ and $<\tau, \mu>.\underline{0}$ with $\lambda > \mu$?

- *Consider subsets of $\mathcal{SC}_{\leq\theta}(P,T)$ including all the successful test-driven computations of the same length.*

- They are $\mathcal{SC}^{l}_{\leq\theta}(P,T)$ for $0 \leq l \leq |\theta|$.

- $\mathcal{SC}^{|\theta|}_{\leq\theta}(P,T)$ is enough as shorter successful test-driven computations can be taken into account when imposing prefixes of θ as time upper bounds.

- Process terms having only exponentially timed τ-actions are compared after giving them the possibility of executing the same number of τ-actions $(\lambda > \mu \Rightarrow \frac{1}{\lambda} < \frac{1}{\mu})$:

$$prob(\mathcal{SC}^{1}_{\leq\frac{1}{\lambda}}(<\tau,\lambda>.\underline{0},\mathrm{s})) \;=\; 1 \;\neq\; 0 \;=\; prob(\mathcal{SC}^{1}_{\leq\frac{1}{\lambda}}(<\tau,\mu>.\underline{0},\mathrm{s}))$$

- $P_1 \in \mathbb{P}_{\mathrm{M,pc}}$ is Markovian testing equivalent to $P_2 \in \mathbb{P}_{\mathrm{M,pc}}$, written $P_1 \sim_{\mathrm{MT}} P_2$, iff for all reactive tests $T \in \mathbb{T}_{\mathrm{R}}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

$$\boxed{\mathit{prob}(\mathcal{SC}^{|\theta|}_{\leq \theta}(P_1, T)) \ = \ \mathit{prob}(\mathcal{SC}^{|\theta|}_{\leq \theta}(P_2, T))}$$

- Not defined as the intersection of may- and must-equivalence as the possibility and the necessity of passing a test are qualitative concepts, hence they are not sufficient (probability > 0, probability $= 1$).

- Not defined as the kernel of a Markovian testing preorder as such a preorder would have boiled down to an equivalence relation.

- The presence of time upper bounds makes it possible to decide whether a test is passed or not even if the process term under test can execute infinitely many exponentially timed τ-actions.

- In order for $P_1 \sim_{\mathrm{MT}} P_2$, it is necessary that for all $c_k \in \mathcal{C}_{\mathrm{f}}(P_k)$, $k \in \{1, 2\}$, there exists $c_h \in \mathcal{C}_{\mathrm{f}}(P_h)$, $h \in \{1, 2\} - \{k\}$, such that:

$$\boxed{\begin{array}{rcl} \mathit{trace}_{\mathrm{c}}(c_k) & = & \mathit{trace}_{\mathrm{c}}(c_h) \\ \mathit{time}_{\mathrm{a}}(c_k) & = & \mathit{time}_{\mathrm{a}}(c_h) \end{array}}$$

  and for all $a \in \mathit{Name}$ and $i \in \{0, \dots, |c_k|\}$:

$$\boxed{\mathit{rate}_{\mathrm{o}}(P_k^i, a, 0) \; = \; \mathit{rate}_{\mathrm{o}}(P_h^i, a, 0)}$$

  with P_k^i (resp. P_h^i) being the i-th state traversed by c_k (resp. c_h).

- Process terms satisfying the necessary condition that are not Markovian testing equivalent ($\lambda_1 + \lambda_2 = \lambda_1' + \lambda_2'$ with $\lambda_1 \neq \lambda_1', \lambda_2 \neq \lambda_2'$ and $b \neq c$ or $\mu \neq \gamma$):

$$<a, \lambda_1>.<b, \mu>.\underline{0} + <a, \lambda_2>.<c, \gamma>.\underline{0}$$

$$<a, \lambda_1'>.<b, \mu>.\underline{0} + <a, \lambda_2'>.<c, \gamma>.\underline{0}$$

- $\sim_{\rm MT}$ has three alternative characterizations, each providing further justifications for the way in which the equivalence has been defined.

- The first characterization establishes that the discriminating power does not change if we consider a set $\mathbb{T}_{\rm R,lib}$ of tests with the following more liberal syntax:

$$\boxed{T \ ::= \ {\rm s} \mid <a, *_w>.T \mid T + T}$$

- In this setting, a successful configuration is a configuration whose test projection includes s as top-level summand.

- For all $P_1, P_2 \in \mathbb{P}_{\rm M,pc}$:

$$\boxed{P_1 \sim_{\rm MT,lib} P_2 \iff P_1 \sim_{\rm MT} P_2}$$

- The second characterization establishes that the discriminating power does not change if we consider a set $\mathbb{T}_{\mathrm{R},\tau}$ of tests capable of moving autonomously by executing exponentially timed τ-actions:

$$\boxed{\begin{array}{rcl} T & ::= & \mathrm{s} \mid T' \\ T' & ::= & <a, *_w>.T \mid <\tau, \lambda>.T \mid T' + T' \end{array}}$$

- For all $P_1, P_2 \in \mathbb{P}_{\mathrm{M,pc}}$:

$$\boxed{P_1 \sim_{\mathrm{MT},\tau} P_2 \iff P_1 \sim_{\mathrm{MT}} P_2}$$

- The third characterization establishes that the discriminating power does not change if we consider the probability distribution of passing tests within arbitrary sequences of amounts of time.

- Considering the (more accurate) stepwise durations of test-driven computations leads to the same equivalence as considering the (easier to work with) stepwise average durations.

- $P_1 \in \mathbb{P}_{\mathrm{M,pc}}$ is Markovian distribution-testing equivalent to $P_2 \in \mathbb{P}_{\mathrm{M,pc}}$, written $P_1 \sim_{\mathrm{MT,d}} P_2$, iff for all reactive tests $T \in \mathbb{T}_{\mathrm{R}}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of amounts of time:

$$\boxed{prob_{\mathrm{d}}(\mathcal{SC}^{|\theta|}(P_1, T), \theta) \ = \ prob_{\mathrm{d}}(\mathcal{SC}^{|\theta|}(P_2, T), \theta)}$$

- For all $P_1, P_2 \in \mathbb{P}_{\mathrm{M,pc}}$:

$$\boxed{P_1 \sim_{\mathrm{MT,d}} P_2 \iff P_1 \sim_{\mathrm{MT}} P_2}$$

- $\sim_{\mathrm{MT}}$ has another alternative characterization that *fully abstracts* from comparing process term behavior in response to tests.

- Based on traces that are extended at each step with the set of visible action names permitted by the environment at that step.

- An element ξ of $(Name_{\mathrm{v}} \times 2^{Name_{\mathrm{v}}})^*$ is an extended trace iff either ξ is the empty sequence ε or:

$$\boxed{\xi \;\equiv\; (a_1, \mathcal{E}_1) \circ (a_2, \mathcal{E}_2) \circ \ldots \circ (a_n, \mathcal{E}_n)}$$

  for some $n \in \mathbb{N}_{>0}$ with $a_i \in \mathcal{E}_i$ and $\mathcal{E}_i$ finite for each $i = 1, \ldots, n$.

- $\mathcal{ET}$: set of extended traces.

- Trace associated with $\xi \in \mathcal{ET}$:

$$\boxed{trace_{\mathrm{et}}(\xi) \; = \; \begin{cases} \varepsilon & \text{if } |\xi| = 0 \\ a \circ trace_{\mathrm{et}}(\xi') & \text{if } \xi \equiv (a, \mathcal{E}) \circ \xi' \end{cases}}$$

- $c \in \mathcal{C}_{\mathrm{f}}(P)$ is compatible with $\xi \in \mathcal{ET}$ iff:

$$\boxed{trace(c) \; = \; trace_{\mathrm{et}}(\xi)}$$

- $\mathcal{CC}(P, \xi)$: multiset of computations in $\mathcal{C}_{\mathrm{f}}(P)$ compatible with ξ.

- The probability and the duration of any computation of $\mathcal{CC}(P, \xi)$ have to be calculated by considering only the action names permitted at each step by ξ.

- Probability w.r.t. ξ of executing $c \in \mathcal{CC}(P, \xi)$:

$$\boxed{prob_{\xi}(c) \; = \; \begin{cases} 1 & \text{if } |c| = 0 \\ \frac{\lambda}{rate_{\mathrm{o}}(P, \mathcal{E} \cup \{\tau\}, 0)} \cdot prob_{\xi'}(c') & \text{if } c \equiv P \xrightarrow{a, \lambda}_{\mathrm{M}} c' \\ & \quad \text{with } \xi \equiv (a, \mathcal{E}) \circ \xi' \\ \frac{\lambda}{rate_{\mathrm{o}}(P, \mathcal{E} \cup \{\tau\}, 0)} \cdot prob_{\xi}(c') & \text{if } c \equiv P \xrightarrow{\tau, \lambda}_{\mathrm{M}} c' \\ & \quad \text{with } \xi \equiv (a, \mathcal{E}) \circ \xi' \\ \frac{\lambda}{rate_{\mathrm{o}}(P, \tau, 0)} \cdot prob_{\xi}(c') & \text{if } c \equiv P \xrightarrow{\tau, \lambda}_{\mathrm{M}} c' \wedge \xi \equiv \varepsilon \end{cases}}$$

- Probability w.r.t. ξ of executing a computation in $C \subseteq \mathcal{CC}(P, \xi)$:

$$\boxed{prob_{\xi}(C) \; = \; \sum_{c \in C} prob_{\xi}(c)}$$

assuming that C is finite and all of its computations are independent.

- Stepwise average duration w.r.t. ξ of $c \in \mathcal{CC}(P, \xi)$:

$$\boxed{time_{\mathrm{a},\xi}(c) \;=\; \begin{cases} \varepsilon & \text{if } |c| = 0 \\ \frac{1}{rate_{\mathrm{o}}(P, \mathcal{E} \cup \{\tau\}, 0)} \circ time_{\mathrm{a},\xi'}(c') & \text{if } c \equiv P \xrightarrow{a,\lambda}_{\mathrm{M}} c' \\ & \quad \text{with } \xi \equiv (a, \mathcal{E}) \circ \xi' \\ \frac{1}{rate_{\mathrm{o}}(P, \mathcal{E} \cup \{\tau\}, 0)} \circ time_{\mathrm{a},\xi}(c') & \text{if } c \equiv P \xrightarrow{\tau,\lambda}_{\mathrm{M}} c' \\ & \quad \text{with } \xi \equiv (a, \mathcal{E}) \circ \xi' \\ \frac{1}{rate_{\mathrm{o}}(P, \tau, 0)} \circ time_{\mathrm{a},\xi}(c') & \text{if } c \equiv P \xrightarrow{\tau,\lambda}_{\mathrm{M}} c' \wedge \xi \equiv \varepsilon \end{cases}}$$

- Multiset of computations in $C \subseteq \mathcal{CC}(P, \xi)$ whose stepwise average duration w.r.t. ξ is not greater than $\theta \in (\mathbb{R}_{>0})^*$:

$$\boxed{C_{\leq \theta, \xi} \;=\; \{\!| \, c \in C \mid |c| \leq |\theta| \wedge \forall i = 1, \ldots, |c|. \, time_{\mathrm{a},\xi}(c)[i] \leq \theta[i] \, |\!\}}$$

- C^l: multiset of computations in $C \subseteq \mathcal{CC}(P, \xi)$ having length $l \in \mathbb{N}$.

- Consider $\mathcal{CC}^{|\theta|}_{\leq\theta,\xi}(P,\xi)$ in order to ensure independence.

- $P_1 \in \mathbb{P}_{\mathrm{M,pc}}$ is Markovian extended-trace equivalent to $P_2 \in \mathbb{P}_{\mathrm{M,pc}}$, written $P_1 \sim_{\mathrm{MTr,e}} P_2$, iff for all extended traces $\xi \in \mathcal{ET}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

$$\boxed{prob_\xi(\mathcal{CC}^{|\theta|}_{\leq\theta,\xi}(P_1,\xi)) \;=\; prob_\xi(\mathcal{CC}^{|\theta|}_{\leq\theta,\xi}(P_2,\xi))}$$

- For all $P_1, P_2 \in \mathbb{P}_{\mathrm{M,pc}}$:

$$\boxed{P_1 \sim_{\mathrm{MTr,e}} P_2 \iff P_1 \sim_{\mathrm{MT}} P_2}$$

- Extended traces identify a set of reactive tests necessary and sufficient in order to establish whether two terms are Markovian testing equivalent.

- Each canonical reactive test admits a main computation leading to success, whose intermediate states can have additional computations each leading to failure in one step.

- Failure is represented through a visible action name z that can occur within tests but not within process terms under test.

- Syntax of the set $\mathbb{T}_{\mathrm{R,c}}$ of canonical reactive tests ($a \in \mathcal{E}$, $\mathcal{E} \subseteq Name_{\mathrm{v}}$ finite):

$$\boxed{T \ ::= \ \mathrm{s} \mid <a, *_1>.T + \sum_{b \in \mathcal{E} - \{a\}} <b, *_1>.<\mathrm{z}, *_1>.\mathrm{s}}$$

- $P_1 \sim_{\mathrm{MT}} P_2$ iff for all $T \in \mathbb{T}_{\mathrm{R,c}}$ and $\theta \in (\mathbb{R}_{>0})^*$:

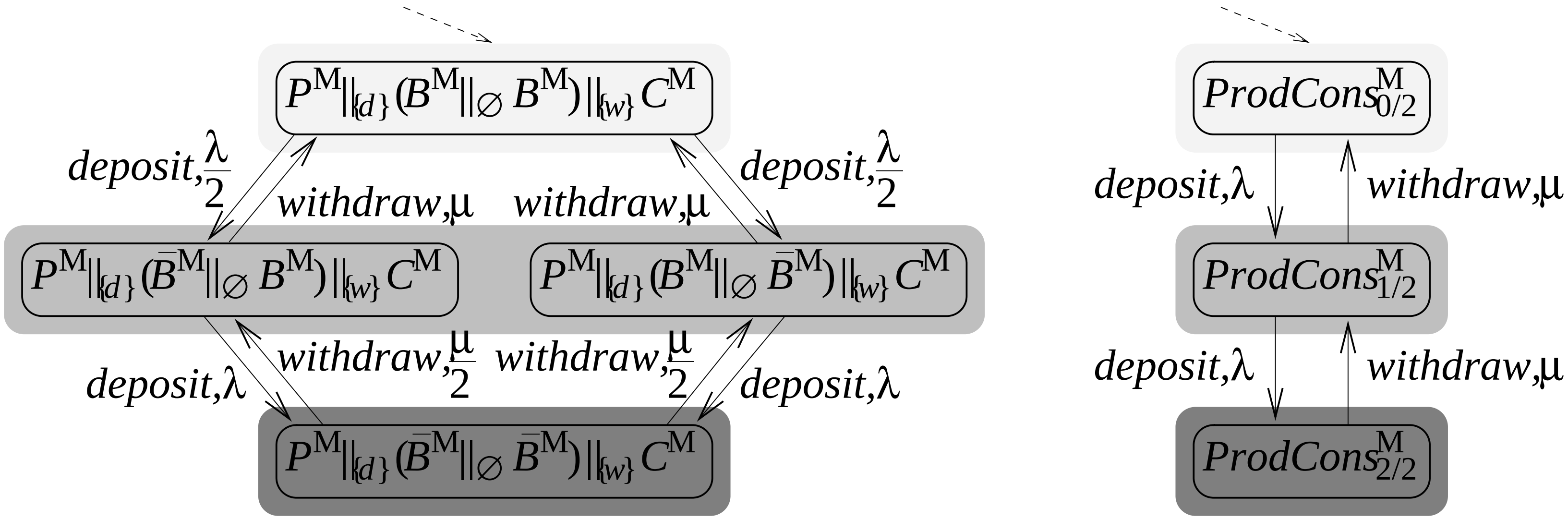
$$\boxed{prob(\mathcal{SC}^{|\theta|}_{\leq \theta}(P_1, T)) \ = \ prob(\mathcal{SC}^{|\theta|}_{\leq \theta}(P_2, T))}$$

- **Running example** ($\sim_{\rm MT}$):
  - Concurrent implementation with two independent one-pos. buffers:

$$\begin{array}{rcl}
PC^{\rm M}_{\rm conc,2} & \triangleq & \mathit{Prod}^{\rm M} \parallel_{\{deposit\}} (\mathit{Buff}^{\rm M} \parallel_{\emptyset} \mathit{Buff}^{\rm M}) \parallel_{\{withdraw\}} \mathit{Cons}^{\rm M} \\
\mathit{Prod}^{\rm M} & \triangleq & <\!deposit, \lambda\!>.\mathit{Prod}^{\rm M} \\
\mathit{Buff}^{\rm M} & \triangleq & <\!deposit, *_1\!>.<\!withdraw, *_1\!>.\mathit{Buff}^{\rm M} \\
\mathit{Cons}^{\rm M} & \triangleq & <\!withdraw, \mu\!>.\mathit{Cons}^{\rm M}
\end{array}$$

  - All the actions occurring in the buffer are passive, consistent with the fact that the buffer is a passive entity.
  - Is $PC^{\rm M}_{\rm conc,2}$ a correct implementation of $\mathit{ProdCons}^{\rm M}_{0/2}$?
  - It turns out that $PC^{\rm M}_{\rm conc,2} \sim_{\rm MT} \mathit{ProdCons}^{\rm M}_{0/2}$.
  - Proved by exploiting the fully abstract alternative characterization.

- Here are the underlying labeled multitransition systems:



- The initial state on the left-hand side has both outgoing transitions labeled with $\lambda/2$, not λ.
- The bottom state on the left-hand side has both outgoing transitions labeled with $\mu/2$, not μ.

- The only sequences of visible actions that the two systems are able to perform are the prefixes of the strings complying with:

$$(\mathit{deposit} \circ (\mathit{deposit} \circ \mathit{withdraw})^* \circ \mathit{withdraw})^*$$

- The only significant extended traces to be considered are those whose associated traces coincide with such prefixes.
- Their nonempty finite sets of visible actions permitted at the various steps necessarily contain at least one between *deposit* and *withdraw*.
- Any two computations of $\mathit{ProdCons}^{\mathrm{M}}_{0/2}$ and $\mathit{PC}^{\mathrm{M}}_{\mathrm{conc},2}$ compatible with such a ξ traverse states that pairwise enable sets of actions with the same names and total rates.
- Therefore the stepwise average durations with respect to ξ of the considered computations are identical.

- ⊙ Four basic cases for the execution probabilities with respect to ξ of $\mathcal{CC}(PC^{\mathrm{M}}_{\mathrm{conc},2}, \xi)$ and $\mathcal{CC}(ProdCons^{\mathrm{M}}_{0/2}, \xi)$:
  - ∗ If $\xi \equiv (deposit, \mathcal{E})$, then for both sets of computations the execution probability is 1.
  - ∗ If $\xi \equiv (deposit, \mathcal{E}_1) \circ (withdraw, \mathcal{E}_2)$, then for both sets of computations the execution probability is 1 if $\mathcal{E}_2$ does not contain *deposit*, $\frac{\mu}{\lambda+\mu}$ otherwise.
  - ∗ If $\xi \equiv (deposit, \mathcal{E}_1) \circ (deposit, \mathcal{E}_2)$, then for both sets of computations the execution probability is 1 if $\mathcal{E}_2$ does not contain *withdraw*, $\frac{\lambda}{\lambda+\mu}$ otherwise.
  - ∗ If $\xi \equiv (deposit, \mathcal{E}_1) \circ (deposit, \mathcal{E}_2) \circ (withdraw, \mathcal{E}_3)$, then for both sets of computations the execution probability is 1 if $\mathcal{E}_2$ does not contain *withdraw*, $\frac{\lambda}{\lambda+\mu}$ otherwise.

- $\sim_{\rm MT}$ is a congruence over $\mathbb{P}_{\rm M,pc}$ with respect to all the dynamic and static operators as well as recursion.

- Let $P_1, P_2 \in \mathbb{P}_{\rm M,pc}$. Whenever $P_1 \sim_{\rm MT} P_2$, then:

$$\boxed{\begin{array}{c} <a, \lambda>.P_1 \;\sim_{\rm MT}\; <a, \lambda>.P_2 \\ P_1 + P \;\sim_{\rm MT}\; P_2 + P \qquad\qquad P + P_1 \;\sim_{\rm MT}\; P + P_2 \\ P_1 \,\|_S\, P \;\sim_{\rm MT}\; P_2 \,\|_S\, P \qquad\qquad P \,\|_S\, P_1 \;\sim_{\rm MT}\; P \,\|_S\, P_2 \\ P_1 \,/\, H \;\sim_{\rm MT}\; P_2 \,/\, H \\ P_1 \setminus L \;\sim_{\rm MT}\; P_2 \setminus L \\ P_1[\varphi] \;\sim_{\rm MT}\; P_2[\varphi] \end{array}}$$

provided that $P \in \mathbb{P}_{\rm M,pc}$ for the alternative composition operator and $P_1 \,\|_S\, P, P_2 \,\|_S\, P \in \mathbb{P}_{\rm M,pc}$ for the parallel composition operator.

- Recursion: extend $\sim_{\mathrm{MT}}$ to open process terms by replacing all variables freely occurring outside rec binders with every closed process term.

- Let $P_1, P_2 \in \mathcal{PL}_{\mathrm{M}}$ be guarded process terms containing free occurrences of $k \in \mathbb{N}$ process variables $X_1, \ldots, X_k \in \mathit{Var}$ at most.

- We define $P_1 \sim_{\mathrm{MT}} P_2$ iff there exist $Q_1, \ldots, Q_k \in \mathbb{P}_{\mathrm{M}}$ such that both $P_1\{Q_i \hookrightarrow X_i \mid 1 \leq i \leq k\}$ and $P_2\{Q_i \hookrightarrow X_i \mid 1 \leq i \leq k\}$ belong to $\mathbb{P}_{\mathrm{M,pc}}$ and for each such group of process terms $Q_1, \ldots, Q_k \in \mathbb{P}_{\mathrm{M}}$:

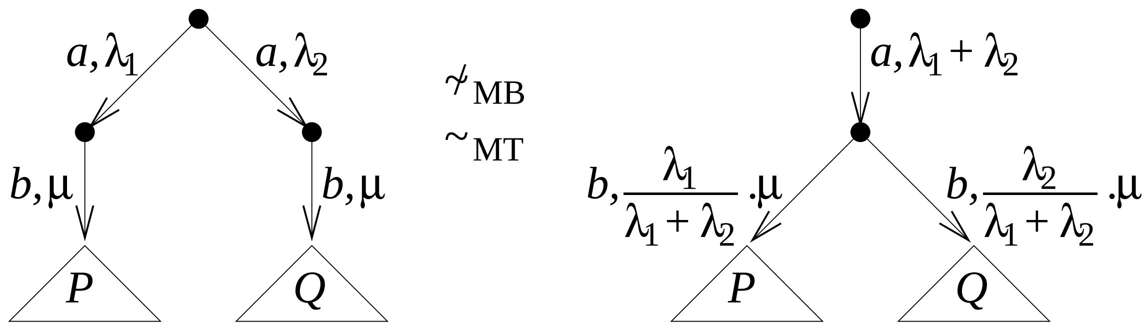
$$\boxed{P_1\{Q_i \hookrightarrow X_i \mid 1 \leq i \leq k\} \ \sim_{\mathrm{MT}} \ P_2\{Q_i \hookrightarrow X_i \mid 1 \leq i \leq k\}}$$

- Whenever $P_1 \sim_{\mathrm{MT}} P_2$, then:

$$\boxed{\mathrm{rec}\, X : P_1 \ \sim_{\mathrm{MT}} \ \mathrm{rec}\, X : P_2}$$

- $\sim_{\rm MT}$ has a sound and complete axiomatization over the set $\mathbb{P}_{\rm M,pc,nrec}$ of nonrecursive process terms of $\mathbb{P}_{\rm M,pc}$.

- The axioms for $\sim_{\rm MB}$ are sound but not complete for $\sim_{\rm MT}$ ($P \not\sim_{\rm MB} Q$):



- *Possibility of deferring choices related to branches starting with the same action name (see the two a-branches on the left-hand side) that are immediately followed by sets of actions having the same names and total rates (see* $\{<b, \mu>\}$ *after each of the two a-branches).*

- Basic laws (identical to those for $\sim_{\rm MB}$):

$$
\begin{array}{|lrcl|}
\hline
(\mathcal{X}_{\rm MT,1}) & P_1 + P_2 & = & P_2 + P_1 \\
(\mathcal{X}_{\rm MT,2}) & (P_1 + P_2) + P_3 & = & P_1 + (P_2 + P_3) \\
(\mathcal{X}_{\rm MT,3}) & P + \underline{0} & = & P \\
\hline
\end{array}
$$

- Characterizing law (subsumes $\sim_{\rm MB}$ axiom for race policy):

$$
\begin{array}{|ll|}
\hline
(\mathcal{X}_{\rm MT,4}) & \sum\limits_{i \in I} {<}a, \lambda_i{>}. \sum\limits_{j \in J_i} {<}b_{i,j}, \mu_{i,j}{>}.P_{i,j} \quad = \\
& \qquad {<}a, \sum\limits_{k \in I} \lambda_k{>}. \sum\limits_{i \in I} \sum\limits_{j \in J_i} {<}b_{i,j}, \frac{\lambda_i}{\Sigma_{k \in I} \lambda_k} \cdot \mu_{i,j}{>}.P_{i,j} \\
& \text{if: } I \text{ is a finite index set with } |I| \geq 2; \\
& \quad \text{for all } i \in I, \text{ index set } J_i \text{ is finite and its summation is } \underline{0} \text{ if } J_i = \emptyset; \\
& \quad \text{for all } i_1, i_2 \in I \text{ and } b \in Name: \\
& \qquad \sum\limits_{j \in J_{i_1}} \{\!| \, \mu_{i_1,j} \mid b_{i_1,j} = b \, |\!\} \; = \; \sum\limits_{j \in J_{i_2}} \{\!| \, \mu_{i_2,j} \mid b_{i_2,j} = b \, |\!\} \\
\hline
\end{array}
$$

- Expansion law (identical to that for $\sim_{\mathrm{MB}}$):

$$
\begin{array}{lrcl}
(\mathcal{X}_{\mathrm{MT},5}) & \sum\limits_{i \in I} <a_i, \tilde{\lambda}_i>.P_i \parallel_S \sum\limits_{j \in J} <b_j, \tilde{\mu}_j>.Q_j & = & \\
& \multicolumn{3}{l}{\sum\limits_{k \in I, a_k \notin S} <a_k, \tilde{\lambda}_k>. \left(P_k \parallel_S \sum\limits_{j \in J} <b_j, \tilde{\mu}_j>.Q_j \right) \; +} \\
& \multicolumn{3}{l}{\sum\limits_{h \in J, b_h \notin S} <b_h, \tilde{\mu}_h>. \left(\sum\limits_{i \in I} <a_i, \tilde{\lambda}_i>.P_i \parallel_S Q_h \right) \; +} \\
& \multicolumn{3}{l}{\sum\limits_{k \in I, a_k \in S, \tilde{\lambda}_k \in \mathbb{R}_{>0}} \; \sum\limits_{h \in J, b_h = a_k, \tilde{\mu}_h = *_{w_h}} <a_k, \tilde{\lambda}_k \cdot \frac{w_h}{weight(Q, b_h)}>.(P_k \parallel_S Q_h) \; +} \\
& \multicolumn{3}{l}{\sum\limits_{h \in J, b_h \in S, \tilde{\mu}_h \in \mathbb{R}_{>0}} \; \sum\limits_{k \in I, a_k = b_h, \tilde{\lambda}_k = *_{v_k}} <b_h, \tilde{\mu}_h \cdot \frac{v_k}{weight(P, a_k)}>.(P_k \parallel_S Q_h) \; +} \\
& \multicolumn{3}{l}{\sum\limits_{k \in I, a_k \in S, \tilde{\lambda}_k = *_{v_k}} \; \sum\limits_{h \in J, b_h = a_k, \tilde{\mu}_h = *_{w_h}} <a_k, *_{norm(v_k, w_h, a_k, P, Q)}>.(P_k \parallel_S Q_h)} \\
(\mathcal{X}_{\mathrm{MT},6}) & \sum\limits_{i \in I} <a_i, \tilde{\lambda}_i>.P_i \parallel_S \underline{0} & = & \sum\limits_{k \in I, a_k \notin S} <a_k, \tilde{\lambda}_k>.P_k \\
(\mathcal{X}_{\mathrm{MT},7}) & \underline{0} \parallel_S \sum\limits_{j \in J} <b_j, \tilde{\mu}_j>.Q_j & = & \sum\limits_{h \in J, b_h \notin S} <b_h, \tilde{\mu}_h>.Q_h \\
(\mathcal{X}_{\mathrm{MT},8}) & \underline{0} \parallel_S \underline{0} & = & \underline{0}
\end{array}
$$

- Distribution laws (identical to those for $\sim_{\rm MB}$):

$$
\begin{array}{llcll}
(\mathcal{X}_{\rm MT,9}) & \underline{0} \,/\, H & = & \underline{0} & \\
(\mathcal{X}_{\rm MT,10}) & (<a, \tilde{\lambda}>.P) \,/\, H & = & <\tau, \tilde{\lambda}>.(P \,/\, H) & \text{if } a \in H \\
(\mathcal{X}_{\rm MT,11}) & (<a, \tilde{\lambda}>.P) \,/\, H & = & <a, \tilde{\lambda}>.(P \,/\, H) & \text{if } a \notin H \\
(\mathcal{X}_{\rm MT,12}) & (P_1 + P_2) \,/\, H & = & P_1 \,/\, H + P_2 \,/\, H & \\
& & & & \\
(\mathcal{X}_{\rm MT,13}) & \underline{0} \setminus L & = & \underline{0} & \\
(\mathcal{X}_{\rm MT,14}) & (<a, \tilde{\lambda}>.P) \setminus L & = & \underline{0} & \text{if } a \in L \\
(\mathcal{X}_{\rm MT,15}) & (<a, \tilde{\lambda}>.P) \setminus L & = & <a, \tilde{\lambda}>.(P \setminus L) & \text{if } a \notin L \\
(\mathcal{X}_{\rm MT,16}) & (P_1 + P_2) \setminus L & = & P_1 \setminus L + P_2 \setminus L & \\
& & & & \\
(\mathcal{X}_{\rm MT,17}) & \underline{0}[\varphi] & = & \underline{0} & \\
(\mathcal{X}_{\rm MT,18}) & (<a, \tilde{\lambda}>.P)[\varphi] & = & <\varphi(a), \tilde{\lambda}>.(P[\varphi]) & \\
(\mathcal{X}_{\rm MT,19}) & (P_1 + P_2)[\varphi] & = & P_1[\varphi] + P_2[\varphi] &
\end{array}
$$

- $DED(\mathcal{X}_{\mathrm{MT}})$: deduction system based on all the previous axioms plus:
  - Reflexivity: $\mathcal{X}_{\mathrm{MT}} \vdash P = P$.
  - Symmetry: $\mathcal{X}_{\mathrm{MT}} \vdash P_1 = P_2 \implies \mathcal{X}_{\mathrm{MT}} \vdash P_2 = P_1$.
  - Transitivity: $\mathcal{X}_{\mathrm{MT}} \vdash P_1 = P_2 \wedge \mathcal{X}_{\mathrm{MT}} \vdash P_2 = P_3 \implies \mathcal{X}_{\mathrm{MT}} \vdash P_1 = P_3$.
  - Substitutivity: $\mathcal{X}_{\mathrm{MT}} \vdash P_1 = P_2 \implies \mathcal{X}_{\mathrm{MT}} \vdash <a, \lambda>.P_1 = <a, \lambda>.P_2 \wedge \dots$

- The deduction system $DED(\mathcal{X}_{\mathrm{MT}})$ is sound and complete for $\sim_{\mathrm{MT}}$ over $\mathbb{P}_{\mathrm{M,pc,nrec}}$; i.e., for all $P_1, P_2 \in \mathbb{P}_{\mathrm{M,pc,nrec}}$:

$$\boxed{\mathcal{X}_{\mathrm{MT}} \vdash P_1 = P_2 \iff P_1 \sim_{\mathrm{MT}} P_2}$$

- $\sim_{\mathrm{MT}}$ has a modal logic characterization over $\mathbb{P}_{\mathrm{M,pc}}$ based on a variant of the Hennessy-Milner logic.

- Negation is not included and conjunction is replaced by disjunction (decreased discriminating power with respect to $\sim_{\mathrm{MB}}$).

- Syntax of the modal language $\mathcal{ML}_{\mathrm{MT}}$ ($a \in Name_{\mathrm{v}}$):

$$\boxed{\begin{array}{rcl} \phi & ::= & \mathrm{true} \mid \phi' \\ \phi' & ::= & \langle a \rangle \phi \mid \phi' \vee \phi' \end{array}}$$

where each formula of the form $\phi_1 \vee \phi_2$ satisfies the following constraint (consistent with the name-deterministic nature of canonical reactive tests):

$$init(\phi_1) \cap init(\phi_2) \; = \; \emptyset$$

with $init(\phi)$ being defined as follows:

$$init(\mathrm{true}) = \emptyset \qquad init(\phi_1 \vee \phi_2) = init(\phi_1) \cup init(\phi_2) \qquad init(\langle a \rangle \phi) = \{a\}$$

- No quantitative decorations in the syntax because the focus is on entire computations rather than on step-by-step behavior mimicking, but ...
- ... replacement of the boolean satisfaction relation with a quantitative interpretation function measuring the probability with which a process term satisfies a formula quickly enough on average.
- Interpretation of $\mathcal{ML}_{\mathrm{MT}}$ over $\mathbb{P}_{\mathrm{M,pc}}$:

$$\boxed{[\![\phi]\!]^{|\theta|}_{\mathrm{MT}}(P,\theta) \;=\; \begin{cases} 0 & \text{if } |\theta| = 0 \land \phi \not\equiv \text{true} \\ & \quad \text{or } |\theta| > 0 \land \mathit{rate}_{\mathrm{o}}(P, \mathit{init}(\phi) \cup \{\tau\}, 0) = 0 \\ 1 & \text{if } |\theta| = 0 \land \phi \equiv \text{true} \end{cases}}$$

otherwise:

$$[\![\mathrm{true}]\!]_{\mathrm{MT}}^{|t\circ\theta|}(P, t\circ\theta) \;=\; \begin{cases} \displaystyle\sum_{P \xrightarrow{\tau,\lambda}_{\mathrm{M}} P'} \frac{\lambda}{rate_{\mathrm{o}}(P,\tau,0)} \cdot [\![\mathrm{true}]\!]_{\mathrm{MT}}^{|\theta|}(P',\theta) & \text{if } \frac{1}{rate_{\mathrm{o}}(P,\tau,0)} \leq t \\ 0 & \text{if } \frac{1}{rate_{\mathrm{o}}(P,\tau,0)} > t \end{cases}$$

$$[\![\langle a\rangle\phi]\!]_{\mathrm{MT}}^{|t\circ\theta|}(P, t\circ\theta) \;=\; \begin{cases} \displaystyle\sum_{P \xrightarrow{a,\lambda}_{\mathrm{M}} P'} \frac{\lambda}{rate_{\mathrm{o}}(P,\{a,\tau\},0)} \cdot [\![\phi]\!]_{\mathrm{MT}}^{|\theta|}(P',\theta) + \\ \qquad \displaystyle\sum_{P \xrightarrow{\tau,\lambda}_{\mathrm{M}} P'} \frac{\lambda}{rate_{\mathrm{o}}(P,\{a,\tau\},0)} \cdot [\![\langle a\rangle\phi]\!]_{\mathrm{MT}}^{|\theta|}(P',\theta) \\ & \text{if } \frac{1}{rate_{\mathrm{o}}(P,\{a,\tau\},0)} \leq t \\ 0 & \text{if } \frac{1}{rate_{\mathrm{o}}(P,\{a,\tau\},0)} > t \end{cases}$$

$$
\begin{aligned}
[\![\phi_1 \vee \phi_2]\!]_{\mathrm{MT}}^{|t \circ \theta|}(P, t \circ \theta) \quad = \quad & p_1 \cdot [\![\phi_1]\!]_{\mathrm{MT}}^{|t_1 \circ \theta|}(P_{\textit{no-init-}\tau}, t_1 \circ \theta) + \\
& p_2 \cdot [\![\phi_2]\!]_{\mathrm{MT}}^{|t_2 \circ \theta|}(P_{\textit{no-init-}\tau}, t_2 \circ \theta) + \\
& \sum_{P \xrightarrow{\tau, \lambda}_{\mathrm{M}} P'} \frac{\lambda}{\mathit{rate}_{\mathrm{o}}(P, \mathit{init}(\phi_1 \vee \phi_2) \cup \{\tau\}, 0)} \cdot [\![\phi_1 \vee \phi_2]\!]_{\mathrm{MT}}^{|\theta|}(P', \theta)
\end{aligned}
$$

where:

- $P_{\textit{no-init-}\tau}$ is P without computations starting with a τ-transition.
- For $j \in \{1, 2\}$:

$$
\begin{aligned}
p_j \quad & = \quad \frac{\mathit{rate}_{\mathrm{o}}(P, \mathit{init}(\phi_j), 0)}{\mathit{rate}_{\mathrm{o}}(P, \mathit{init}(\phi_1 \vee \phi_2) \cup \{\tau\}, 0)} \\
t_j \quad & = \quad t + \left(\frac{1}{\mathit{rate}_{\mathrm{o}}(P, \mathit{init}(\phi_j), 0)} - \frac{1}{\mathit{rate}_{\mathrm{o}}(P, \mathit{init}(\phi_1 \vee \phi_2) \cup \{\tau\}, 0)}\right)
\end{aligned}
$$

with p_j representing the conditional probability with which P performs actions whose name is in $\mathit{init}(\phi_j)$ and t_j representing the extra average time granted to P for satisfying ϕ_j.

- The constraint on disjunctions guarantees that their subformulas exercise independent computations of P (correct probability calculation).
- In the absence of p_1 and p_2, the fact that $\phi_1 \vee \phi_2$ offers a set of initial actions at least as large as the ones offered by ϕ_1 alone and by ϕ_2 alone may lead to an overestimate of the probability of satisfying $\phi_1 \vee \phi_2$.
- Considering t instead of t_j in the satisfaction of ϕ_j in isolation may lead to an underestimate of the probability of satisfying $\phi_1 \vee \phi_2$ within the given time upper bound, as P may satisfy $\phi_1 \vee \phi_2$ within $t \circ \theta$ even if P satisfies neither ϕ_1 nor ϕ_2 taken in isolation within $t \circ \theta$.
- For all $P_1, P_2 \in \mathbb{P}_{\mathrm{M,pc}}$:

$$\boxed{P_1 \sim_{\mathrm{MT}} P_2 \iff \forall \phi \in \mathcal{ML}_{\mathrm{MT}}. \forall \theta \in (\mathbb{R}_{>0})^*. \llbracket \phi \rrbracket_{\mathrm{MT}}^{|\theta|}(P_1, \theta) = \llbracket \phi \rrbracket_{\mathrm{MT}}^{|\theta|}(P_2, \theta)}$$

- $\sim_{\rm MT}$ is decidable in polynomial time over the set $\mathbb{P}_{\rm M,pc,fin}$ of finite-state process terms of $\mathbb{P}_{\rm M,pc}$.

- The reason is that:

  - $\sim_{\rm MT}$ coincides with the Markovian version of ready equivalence.
  - Probabilistic ready equivalence can be decided in polynomial time through a suitable reworking of Tzeng algorithm for probabilistic language equivalence.

- Given two process terms, their name-labeled CTMCs are Markovian ready equivalent iff the corresponding embedded name-labeled DTMCs are probabilistic ready equivalent.

- Markovian ready equivalence and probabilistic ready equivalence coincide on corresponding models if the total exit rate of each state of a name-labeled CTMC is encoded inside the names of all transitions departing from that state in the associated embedded DTMC.

- Steps of the algorithm for checking whether $P_1 \sim_{\rm MT} P_2$:
  1. Transform $[\![P_1]\!]_{\rm M}$ and $[\![P_2]\!]_{\rm M}$ into their corresponding embedded discrete-time versions:
     a. Divide the rate of each transition by the total exit rate of its source state.
     b. Augment the name of each transition with the total exit rate of its source state.
  2. Compute the relation $\mathcal{R}$ that equates any two states of the discrete-time versions of $[\![P_1]\!]_{\rm M}$ and $[\![P_2]\!]_{\rm M}$ whenever the two sets of augmented action names labeling the transitions departing from the two states coincide.
  3. For each equivalence class R induced by $\mathcal{R}$, consider R as the set of accepting states and check whether the discrete-time versions of $[\![P_1]\!]_{\rm M}$ and $[\![P_2]\!]_{\rm M}$ are probabilistic language equivalent.
  4. Return yes/no depending on whether all the checks performed in the previous step have been successful or not.

- Tzeng algorithm for probabilistic language equivalence visits in breadth-first order the tree containing a node for each possible string and studies the linear independence of the state probability vectors associated with a finite subset of the tree nodes.

- Refinement of each iteration of step 3:

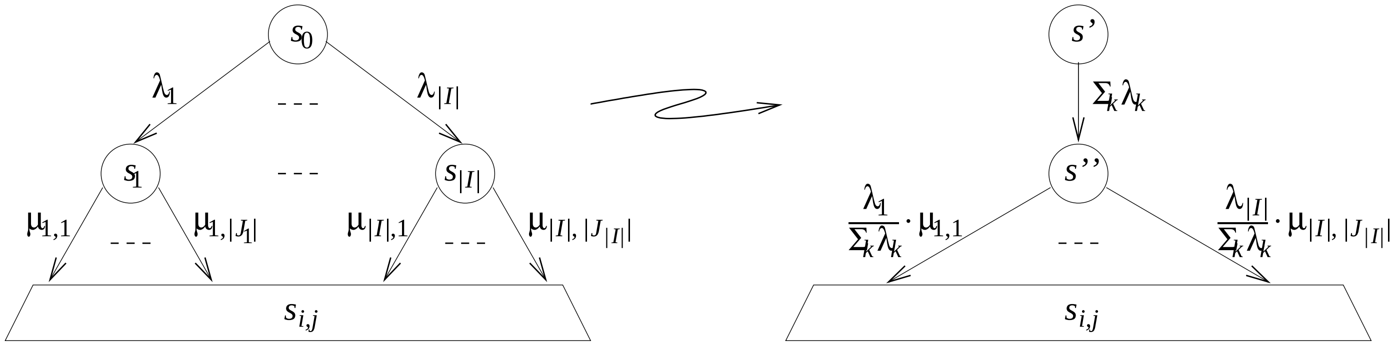
  1. Create an empty set V of state probability vectors.
  2. Create a queue whose only element is the empty string ε.
  3. While the queue is not empty:
     a. Remove the first element from the queue, say string ς.
     b. If the state probability vector of the discrete-time versions of $[\![P_1]\!]_{\mathrm{M}}$ and $[\![P_2]\!]_{\mathrm{M}}$ after reading ς does not belong to the vector space generated by V, then:
        i. For each $a \in \mathit{NameReal}_{P_1,P_2}$, add $\varsigma \circ a$ to the queue.
        ii. Add the state probability vector to V.

4. Build a three-valued state vector $\boldsymbol{u}$ whose generic element is:
    a. 0 if it corresponds to a nonaccepting state.
    b. 1 if it corresponds to an accepting state of $[\![P_1]\!]_{\mathrm{M}}$.
    c. -1 if it corresponds to an accepting state of $[\![P_2]\!]_{\mathrm{M}}$.
5. For each $\boldsymbol{v} \in V$, check whether $\boldsymbol{v} \cdot \boldsymbol{u}^{\mathrm{T}} = 0$.
6. Return yes/no depending on whether all the checks performed in the previous step have been successful or not.

- The time complexity of the overall algorithm is $O(n^5)$.

- $\sim_{\mathrm{MT}}$ induces an exact aggregation called T-lumping.

- T-lumping is strictly coarser than ordinary lumping and graphically definable as follows (name-abstracting axiom schema characterizing $\sim_{\mathrm{MT}}$):



where for all $i_1, i_2 \in I$:

$$\sum_{j \in J_{i_1}} \mu_{i_1,j} = \sum_{j \in J_{i_2}} \mu_{i_2,j}$$

- Exact aggregation not previously known in the CTMC field, but entirely characterizable in a process algebraic framework like ordinary lumping.

- Two Markovian testing equivalent process terms in $\mathbb{P}_{\mathrm{M,pc}}$ are guaranteed to possess the same performance characteristics.

Markovian Trace Equivalence

- Two process terms are equivalent if they can perform computations with the same functional and performance characteristics.
- Test passing replaced by trace acceptance (linear, unconstrained environment).
- Was the trace accepted?
  If so, with which probability?
  And how long did it take to accept the trace?
- Comparison of process term probabilities of performing trace-compatible computations within arbitrary time upper bounds.
- Branching points in process term behavior are all overridden.

- Comparing probabilities of accepting a trace within a time upper bound.
- $c \in \mathcal{C}_{\mathrm{f}}(P)$ is compatible with $\alpha \in (Name_{\mathrm{v}})^*$ iff:

$$\boxed{trace(c) \; = \; \alpha}$$

- $\mathcal{CC}(P, \alpha)$: multiset of computations in $\mathcal{C}_{\mathrm{f}}(P)$ compatible with α.
- If P has no exponentially timed τ-actions:
  - ⊙ All the computations in $\mathcal{CC}(P, \alpha)$ are independent.
  - ⊙ The multiset $\mathcal{CC}(P, \alpha)$ is finite.
- Same properties for $\mathcal{CC}_{\leq \theta}(P, \alpha)$.
- If there are exponentially timed τ-actions:
  - ⊙ Are the computations in $\mathcal{CC}_{\leq \theta}(P, \alpha)$ independent of each other?
  - ⊙ How to distinguish among process terms having only exponentially timed τ-actions, like $<\tau, \lambda>.\underline{0}$ and $<\tau, \mu>.\underline{0}$ with $\lambda > \mu$?

- *Consider subsets of* $\mathcal{CC}_{\leq\theta}(P,\alpha)$ *including all the trace-compatible computations of the same length.*

- They are $\mathcal{CC}^{l}_{\leq\theta}(P,\alpha)$ for $0 \leq l \leq |\theta|$.

- $\mathcal{CC}^{|\theta|}_{\leq\theta}(P,\alpha)$ is enough as shorter trace-compatible computations can be taken into account when imposing prefixes of θ as time upper bounds.

- Process terms having only exponentially timed τ-actions are compared after giving them the possibility of executing the same number of τ-actions ($\lambda > \mu \Rightarrow \frac{1}{\lambda} < \frac{1}{\mu}$):

$$prob(\mathcal{CC}^{1}_{\leq\frac{1}{\lambda}}(<\tau,\lambda>.\underline{0},\varepsilon)) \;=\; 1 \;\neq\; 0 \;=\; prob(\mathcal{CC}^{1}_{\leq\frac{1}{\lambda}}(<\tau,\mu>.\underline{0},\varepsilon))$$

- $P_1 \in \mathbb{P}_{\mathrm{M,pc}}$ is Markovian trace equivalent to $P_2 \in \mathbb{P}_{\mathrm{M,pc}}$, written $P_1 \sim_{\mathrm{MTr}} P_2$, iff for all traces $\alpha \in (Name_{\mathrm{v}})^*$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

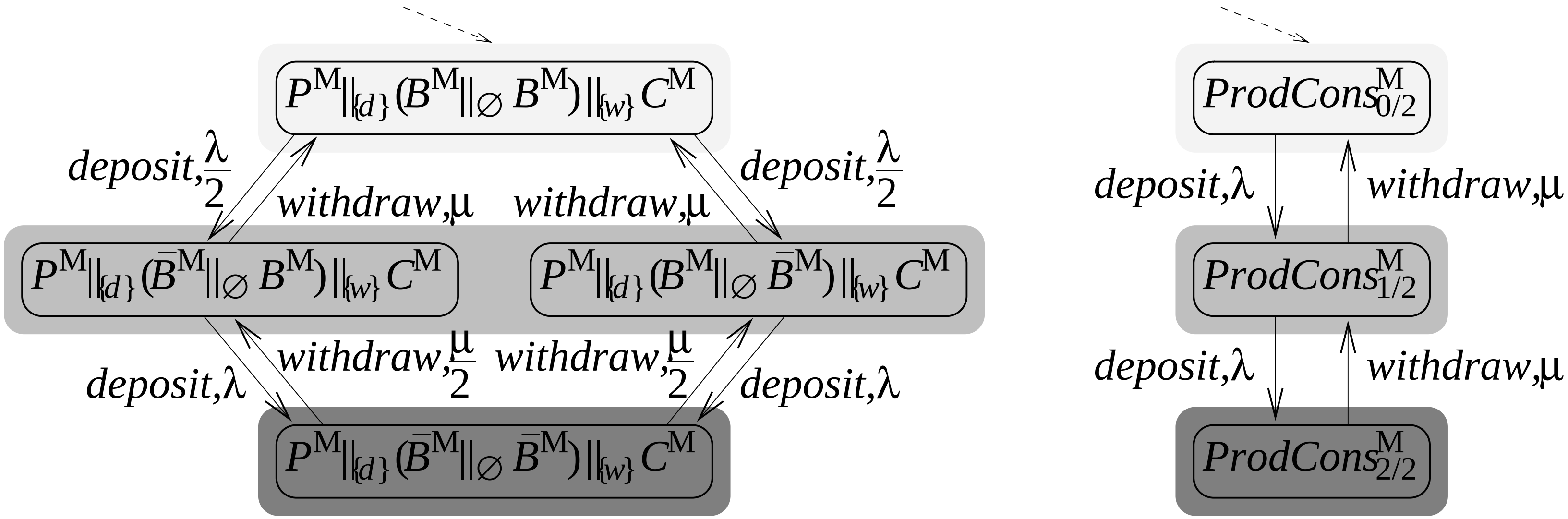
$$\boxed{prob(\mathcal{CC}^{|\theta|}_{\leq\theta}(P_1,\alpha)) \;=\; prob(\mathcal{CC}^{|\theta|}_{\leq\theta}(P_2,\alpha))}$$

- **Running example** ($\sim_{\rm MTr}$):
  - ⊙ Concurrent implementation with two independent one-pos. buffers:

$$
\begin{array}{rcl}
PC^{\rm M}_{\rm conc,2} & \triangleq & Prod^{\rm M} \parallel_{\{deposit\}} (Buff^{\rm M} \parallel_{\emptyset} Buff^{\rm M}) \parallel_{\{withdraw\}} Cons^{\rm M} \\
Prod^{\rm M} & \triangleq & <deposit, \lambda>.Prod^{\rm M} \\
Buff^{\rm M} & \triangleq & <deposit, *_1>.<withdraw, *_1>.Buff^{\rm M} \\
Cons^{\rm M} & \triangleq & <withdraw, \mu>.Cons^{\rm M}
\end{array}
$$

  - ⊙ All the actions occurring in the buffer are passive, consistent with the fact that the buffer is a passive entity.
  - ⊙ Is $PC^{\rm M}_{\rm conc,2}$ a correct implementation of $ProdCons^{\rm M}_{0/2}$?
  - ⊙ It turns out that $PC^{\rm M}_{\rm conc,2} \sim_{\rm MTr} ProdCons^{\rm M}_{0/2}$.

- Here are the underlying labeled multitransition systems:



- The initial state on the left-hand side has both outgoing transitions labeled with $\lambda/2$, not λ.

- The bottom state on the left-hand side has both outgoing transitions labeled with $\mu/2$, not μ.

- The only sequences of visible actions that the two systems are able to perform are the prefixes of the strings complying with:

$$(\mathit{deposit} \circ (\mathit{deposit} \circ \mathit{withdraw})^* \circ \mathit{withdraw})^*$$

- The only significant traces to be considered are those coinciding with such prefixes.
- Any two computations of $\mathit{ProdCons}^{\mathrm{M}}_{0/2}$ and $\mathit{PC}^{\mathrm{M}}_{\mathrm{conc},2}$ compatible with such an α traverse states that pairwise have the same average sojourn time.
- Therefore the stepwise average durations of the considered computations are identical.

⊙ Four basic cases for the execution probabilities of $\mathcal{CC}(PC^{\mathrm{M}}_{\mathrm{conc},2}, \alpha)$ and $\mathcal{CC}(ProdCons^{\mathrm{M}}_{0/2}, \alpha)$:

- If $\alpha \equiv deposit$, then for both sets of computations the execution probability is 1.
- If $\alpha \equiv deposit \circ withdraw$, then for both sets of computations the execution probability is $\frac{\mu}{\lambda+\mu}$.
- If $\alpha \equiv deposit \circ deposit$, then for both sets of computations the execution probability is $\frac{\lambda}{\lambda+\mu}$.
- If $\alpha \equiv deposit \circ deposit \circ withdraw$, then for both sets of computations the execution probability is $\frac{\lambda}{\lambda+\mu}$.

- In order for $P_1 \sim_{\mathrm{MTr}} P_2$, it is necessary that for all $c_k \in \mathcal{C}_{\mathrm{f}}(P_k)$, $k \in \{1, 2\}$, there exists $c_h \in \mathcal{C}_{\mathrm{f}}(P_h)$, $h \in \{1, 2\} - \{k\}$, such that:

$$\boxed{\begin{array}{rcl} \mathit{trace}_{\mathrm{c}}(c_k) & = & \mathit{trace}_{\mathrm{c}}(c_h) \\ \mathit{time}_{\mathrm{a}}(c_k) & = & \mathit{time}_{\mathrm{a}}(c_h) \end{array}}$$

  and for all $i \in \{0, \ldots, |c_k|\}$:

$$\boxed{\mathit{rate}_{\mathrm{t}}(P_k^i, 0) \; = \; \mathit{rate}_{\mathrm{t}}(P_h^i, 0)}$$

  with P_k^i (resp. P_h^i) being the i-th state traversed by c_k (resp. c_h).

- Process terms satisfying the necessary condition that are not Markovian trace equivalent ($\lambda_1 + \lambda_2 = \lambda_1' + \lambda_2'$ with $\lambda_1 \neq \lambda_1', \lambda_2 \neq \lambda_2'$ and $b \neq c$ or $\mu \neq \gamma$):

$$<a, \lambda_1>.<b, \mu>.\underline{0} + <a, \lambda_2>.<c, \gamma>.\underline{0}$$

$$<a, \lambda_1'>.<b, \mu>.\underline{0} + <a, \lambda_2'>.<c, \gamma>.\underline{0}$$

- $\sim_{\mathrm{MTr}}$ has an alternative characterization showing that its discriminating power does not change if we consider the probability distribution of accepting traces within arbitrary sequences of amounts of time.

- Considering the (more accurate) stepwise durations of trace-compatible computations leads to the same equivalence as considering the (easier to work with) stepwise average durations.

- $P_1 \in \mathbb{P}_{\mathrm{M,pc}}$ is Markovian distribution-trace equivalent to $P_2 \in \mathbb{P}_{\mathrm{M,pc}}$, written $P_1 \sim_{\mathrm{MTr,d}} P_2$, iff for all traces $\alpha \in (Name_{\mathrm{v}})^*$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of amounts of time:

$$\boxed{prob_{\mathrm{d}}(\mathcal{CC}^{|\theta|}(P_1, \alpha), \theta) \;=\; prob_{\mathrm{d}}(\mathcal{CC}^{|\theta|}(P_2, \alpha), \theta)}$$

- For all $P_1, P_2 \in \mathbb{P}_{\mathrm{M,pc}}$:

$$\boxed{P_1 \sim_{\mathrm{MTr,d}} P_2 \iff P_1 \sim_{\mathrm{MTr}} P_2}$$

- $\sim_{\mathrm{MTr}}$ is a congruence over $\mathbb{P}_{\mathrm{M,pc}}$ w.r.t. all the dynamic operators.
- Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M,pc}}$. Whenever $P_1 \sim_{\mathrm{MTr}} P_2$, then:

$$\boxed{\begin{array}{c} <a, \lambda>.P_1 \ \sim_{\mathrm{MTr}} \ <a, \lambda>.P_2 \\ P_1 + P \ \sim_{\mathrm{MTr}} \ P_2 + P \qquad\qquad P + P_1 \ \sim_{\mathrm{MTr}} \ P + P_2 \end{array}}$$

- *Not a congruence with respect to parallel composition.*
- For instance, the Markovian trace equivalent process terms $(b \neq c)$:

$$<a, \lambda_1>.<b, \mu>.\underline{0} + <a, \lambda_2>.<c, \mu>.\underline{0}$$

$$<a, \lambda_1 + \lambda_2>.(<b, \tfrac{\lambda_1}{\lambda_1+\lambda_2} \cdot \mu>.\underline{0} + <c, \tfrac{\lambda_2}{\lambda_1+\lambda_2} \cdot \mu>.\underline{0})$$

  are distinguished in the following context:

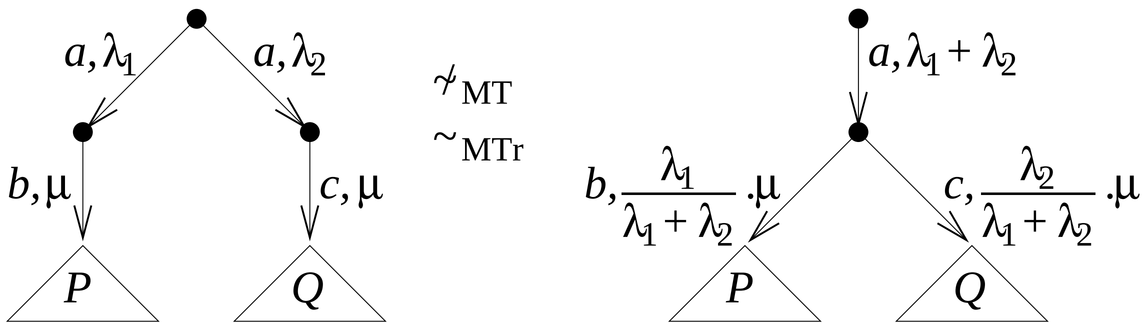
$$\_ \parallel_{\{a,b,c\}} <a, *_1>.<b, *_1>.\underline{0}$$

  by the following trace:

$$\alpha \ \equiv \ a \circ b$$

- $\sim_{\mathrm{MTr}}$ has a sound and complete axiomatization over the set $\mathbb{P}_{\mathrm{M,pc,dyn}}$ of process terms of $\mathbb{P}_{\mathrm{M,pc}}$ comprising only dynamic operators.

- The axioms for $\sim_{\mathrm{MT}}$ are sound but not complete for $\sim_{\mathrm{MTr}}$ $(b \neq c)$:



- *Possibility of deferring choices related to branches starting with actions having the same name (a) that are immediately followed by process terms having the same total exit rate (μ).*

- Names and total rates of the initial actions of such derivative terms can be different in the various branches.

- Basic laws (identical to those for $\sim_{\mathrm{MT}}$):

$$
\begin{array}{lrcl}
(\mathcal{X}_{\mathrm{MTr},1}) & P_1 + P_2 & = & P_2 + P_1 \\
(\mathcal{X}_{\mathrm{MTr},2}) & (P_1 + P_2) + P_3 & = & P_1 + (P_2 + P_3) \\
(\mathcal{X}_{\mathrm{MTr},3}) & P + \underline{0} & = & P
\end{array}
$$

- Characterizing law (subsumes $\sim_{\mathrm{MT}}$ characterizing law):

$$
(\mathcal{X}_{\mathrm{MTr},4}) \quad \sum_{i \in I} <a, \lambda_i>. \sum_{j \in J_i} <b_{i,j}, \mu_{i,j}>.P_{i,j} \; = \; <a, \sum_{k \in I} \lambda_k>. \sum_{i \in I} \sum_{j \in J_i} <b_{i,j}, \frac{\lambda_i}{\Sigma_{k \in I} \lambda_k} \cdot \mu_{i,j}>.P_{i,j}
$$

if: I is a finite index set with $|I| \geq 2$;
for all $i \in I$, index set J_i is finite and its summation is $\underline{0}$ if $J_i = \emptyset$;
for all $i_1, i_2 \in I$:

$$
\sum_{j \in J_{i_1}} \mu_{i_1,j} = \sum_{j \in J_{i_2}} \mu_{i_2,j}
$$

- $DED(\mathcal{X}_{\mathrm{MTr}})$: deduction system based on all the previous axioms plus:
  - ⊙ Reflexivity: $\mathcal{X}_{\mathrm{MTr}} \vdash P = P$.
  - ⊙ Symmetry: $\mathcal{X}_{\mathrm{MTr}} \vdash P_1 = P_2 \implies \mathcal{X}_{\mathrm{MTr}} \vdash P_2 = P_1$.
  - ⊙ Transitivity: $\mathcal{X}_{\mathrm{MTr}} \vdash P_1 = P_2 \wedge \mathcal{X}_{\mathrm{MTr}} \vdash P_2 = P_3 \implies \mathcal{X}_{\mathrm{MTr}} \vdash P_1 = P_3$.
  - ⊙ Substitutivity: $\mathcal{X}_{\mathrm{MTr}} \vdash P_1 = P_2 \implies \mathcal{X}_{\mathrm{MTr}} \vdash {<}a, \lambda{>}.P_1 = {<}a, \lambda{>}.P_2 \wedge \dots$

- The deduction system $DED(\mathcal{X}_{\mathrm{MTr}})$ is sound and complete for $\sim_{\mathrm{MTr}}$ over $\mathbb{P}_{\mathrm{M,pc,dyn}}$; i.e., for all $P_1, P_2 \in \mathbb{P}_{\mathrm{M,pc,dyn}}$:

$$\boxed{\mathcal{X}_{\mathrm{MTr}} \vdash P_1 = P_2 \iff P_1 \sim_{\mathrm{MTr}} P_2}$$

- $\sim_{\mathrm{MTr}}$ has a modal logic characterization over $\mathbb{P}_{\mathrm{M,pc}}$ based on a variant of the Hennessy-Milner logic.

- Neither negation nor any binary connective is included, only diamond (decreased discriminating power with respect to $\sim_{\mathrm{MT}}$).

- Syntax of the modal language $\mathcal{ML}_{\mathrm{MTr}}$ $(a \in \mathit{Name}_{\mathrm{v}})$:

$$\boxed{\phi \ ::= \ \mathrm{true} \mid \langle a \rangle \phi}$$

- No quantitative decorations in the syntax because the focus is on entire computations rather than on step-by-step behavior mimicking, but ...

- ... replacement of the boolean satisfaction relation with a quantitative interpretation function measuring the probability with which a process term satisfies a formula quickly enough on average.

- Interpretation of $\mathcal{ML}_{\mathrm{MTr}}$ over $\mathbb{P}_{\mathrm{M,pc}}$:

$$\boxed{[\![\phi]\!]^{|\theta|}_{\mathrm{MTr}}(P,\theta) \;=\; \begin{cases} 0 & \text{if } |\theta| = 0 \wedge \phi \not\equiv \mathrm{true} \\ & \quad \text{or } |\theta| > 0 \wedge \mathit{rate}_{\mathrm{t}}(P,0) = 0 \\ 1 & \text{if } |\theta| = 0 \wedge \phi \equiv \mathrm{true} \end{cases}}$$

otherwise:

$$[\![\mathrm{true}]\!]^{|t\circ\theta|}_{\mathrm{MTr}}(P,t\circ\theta) \;=\; \begin{cases} \sum\limits_{P \xrightarrow{\tau,\lambda}_{\mathrm{M}} P'} \frac{\lambda}{\mathit{rate}_{\mathrm{t}}(P,0)} \cdot [\![\mathrm{true}]\!]^{|\theta|}_{\mathrm{MTr}}(P',\theta) & \text{if } \frac{1}{\mathit{rate}_{\mathrm{t}}(P,0)} \leq t \\ 0 & \text{if } \frac{1}{\mathit{rate}_{\mathrm{t}}(P,0)} > t \end{cases}$$

$$
[\![\langle a\rangle\phi]\!]_{\mathrm{MTr}}^{|t\circ\theta|}(P, t\circ\theta) \quad = \quad \begin{cases} \displaystyle\sum_{P \xrightarrow{a,\lambda}_{\mathrm{M}} P'} \frac{\lambda}{rate_{\mathrm{t}}(P,0)} \cdot [\![\phi]\!]_{\mathrm{MTr}}^{|\theta|}(P', \theta) + \\ \displaystyle\qquad\sum_{P \xrightarrow{\tau,\lambda}_{\mathrm{M}} P'} \frac{\lambda}{rate_{\mathrm{t}}(P,0)} \cdot [\![\langle a\rangle\phi]\!]_{\mathrm{MTr}}^{|\theta|}(P', \theta) & \text{if } \frac{1}{rate_{\mathrm{t}}(P,0)} \leq t \\ 0 & \text{if } \frac{1}{rate_{\mathrm{t}}(P,0)} > t \end{cases}
$$

- For all $P_1, P_2 \in \mathbb{P}_{\mathrm{M,pc}}$:

$$
\boxed{P_1 \sim_{\mathrm{MTr}} P_2 \iff \forall \phi \in \mathcal{ML}_{\mathrm{MTr}}. \forall \theta \in (\mathbb{R}_{>0})^*. [\![\phi]\!]_{\mathrm{MTr}}^{|\theta|}(P_1, \theta) = [\![\phi]\!]_{\mathrm{MTr}}^{|\theta|}(P_2, \theta)}
$$

- $\sim_{\mathrm{MTr}}$ is decidable in polynomial time over the set $\mathbb{P}_{\mathrm{M,pc,fin}}$ of finite-state process terms of $\mathbb{P}_{\mathrm{M,pc}}$.

- Reworking of Tzeng algorithm for probabilistic language equivalence.

- Given two process terms, their name-labeled CTMCs are Markovian trace equivalent iff the corresponding embedded name-labeled DTMCs are probabilistic trace equivalent.

- Probabilistic trace equivalence is decidable in polynomial time through the algorithm for probabilistic language equivalence.

- Markovian trace equivalence and probabilistic trace equivalence coincide on corresponding models if the total exit rate of each state of a name-labeled CTMC is encoded inside the names of all transitions departing from that state in the associated embedded DTMC.

- Steps of the algorithm for checking whether $P_1 \sim_{\mathrm{MTr}} P_2$:
  1. Transform $[\![P_1]\!]_{\mathrm{M}}$ and $[\![P_2]\!]_{\mathrm{M}}$ into their corresponding embedded discrete-time versions:
     a. Divide the rate of each transition by the total exit rate of its source state.
     b. Augment the name of each transition with the total exit rate of its source state.
  2. Check whether the discrete-time versions of $[\![P_1]\!]_{\mathrm{M}}$ and $[\![P_2]\!]_{\mathrm{M}}$ are probabilistic language equivalent when all of their states are considered as accepting states.
  3. Return yes/no depending on whether the check performed in the previous step has been successful or not.

- Tzeng algorithm for probabilistic language equivalence visits in breadth-first order the tree containing a node for each possible string and studies the linear independence of the state probability vectors associated with a finite subset of the tree nodes.

- Refinement of step 2:

  1. Create an empty set V of state probability vectors.
  2. Create a queue whose only element is the empty string ε.
  3. While the queue is not empty:

     a. Remove the first element from the queue, say string ς.

     b. If the state probability vector of the discrete-time versions of $[\![P_1]\!]_{\mathrm{M}}$ and $[\![P_2]\!]_{\mathrm{M}}$ after reading ς does not belong to the vector space generated by V, then:

        i. For each $a \in \mathit{NameReal}_{P_1,P_2}$, add $\varsigma \circ a$ to the queue.

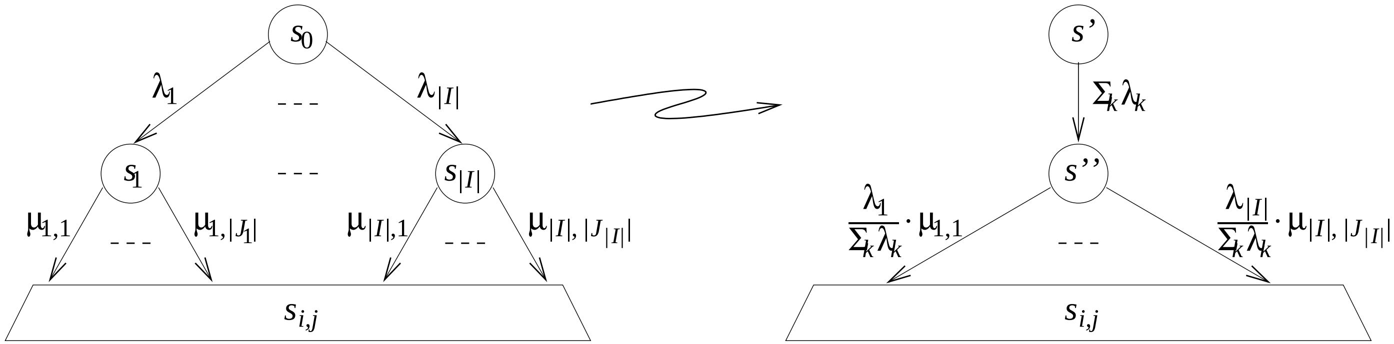
        ii. Add the state probability vector to V.

4. Build a three-valued state vector $\boldsymbol{u}$ whose generic element is:
   a. 0 if it corresponds to a nonaccepting state.
   b. 1 if it corresponds to an accepting state of $[\![P_1]\!]_{\mathrm{M}}$.
   c. -1 if it corresponds to an accepting state of $[\![P_2]\!]_{\mathrm{M}}$.
5. For each $\boldsymbol{v} \in V$, check whether $\boldsymbol{v} \cdot \boldsymbol{u}^{\mathrm{T}} = 0$.
6. Return yes/no depending on whether all the checks performed in the previous step have been successful or not.

- The time complexity of the overall algorithm is $O(n^4)$.

- $\sim_{\mathrm{MTr}}$ induces an exact aggregation called T-lumping.

- T-lumping is strictly coarser than ordinary lumping and graphically definable as follows (name-abstracting axiom schema characterizing $\sim_{\mathrm{MTr}}$):



  where for all $i_1, i_2 \in I$:

$$\sum_{j \in J_{i_1}} \mu_{i_1,j} = \sum_{j \in J_{i_2}} \mu_{i_2,j}$$

- Exact aggregation not previously known in the CTMC field, but entirely characterizable in a process algebraic framework like ordinary lumping.

- Two Markovian trace equivalent process terms in $\mathbb{P}_{\mathrm{M,pc}}$ are guaranteed to possess the same performance characteristics.

Summary of Known Results

- Comparing Markovian behavioral equivalences based on given criteria:

	congruence property	sound & complete axiomatization	modal logic characteriz.	verification complexity	exact aggreg.
$\sim_{\mathrm{MB}}$	$\surd$	$\surd$	$\surd$	$O(m \cdot \log n)$	$\surd$
$\sim_{\mathrm{MT}}$	$\surd$	$\surd$	$\surd$	$O(n^5)$	$\surd$
$\sim_{\mathrm{MTr}}$	*dynamic*	*dynamic*	$\surd$	$O(n^4)$	$\surd$

- Not only intuitively appropriate from the functional viewpoint, but also meaningful and useful from the performance standpoint.

- Aggregating the state space of a model by exploiting symmetries or reducing the state space of a model before analysis takes place without altering the performance properties to be assessed.

Part IV:
What Remains To Do?

Open Problems

- Markovian behavioral equivalence inducing the coarsest exact nontrivial CTMC-level aggregation?
- Minimization algorithms for $\sim_{\rm MT}$ and $\sim_{\rm MTr}$ (and T-lumping)?
- Uniform definitions for nondeterministic, probabilistic, and Markovian processes? *[Bernardo - De Nicola - Loreti, 2010]*
- Approximated versions of $\sim_{\rm MB}$, $\sim_{\rm MT}$, and $\sim_{\rm MTr}$ that relax the comparison on exit rates or execution probabilities? *[Aldini, 2010]*
- Weaker versions of $\sim_{\rm MB}$, $\sim_{\rm MT}$, and $\sim_{\rm MTr}$ that abstract from internal exponentially timed actions while preserving nontrivial exactness?

Abstracting from Internal Actions

- When comparing nondeterministic processes, internal actions can be abstracted away via *weak behavioral equivalences*: $a \,.\, \tau \,.\, b \,.\, \underline{0} \approx a \,.\, b \,.\, \underline{0}$.

- Abstraction not always possible when comparing Markovian processes.

- Immediate internal actions: invisible and take no time [Her,Ret,MT,AB].

- An exponentially timed internal action is invisible but takes time.

- $<a, \lambda>.<\tau, \gamma>.<b, \mu>.\underline{0}$ is not equivalent to $<a, \lambda>.<b, \mu>.\underline{0}$ because a nonzero delay can be observed between a and b in the first case.

- However $<a, \lambda>.<\tau, \gamma_1>.<\tau, \gamma_2>.<b, \mu>.\underline{0} \approx <a, \lambda>.<\tau, \gamma>.<b, \mu>.\underline{0}$ if the average duration of the sequence of the two τ-actions on the left is equal to the average duration of the τ-action on the right: $\frac{1}{\gamma_1} + \frac{1}{\gamma_2} = \frac{1}{\gamma}$ or equivalently $\gamma = \frac{\gamma_1 \cdot \gamma_2}{\gamma_1 + \gamma_2}$.

- *To what extent can we abstract from exp. timed internal actions? Only from sequences (of at least two) or also from branches?*

- *How to define a Markovian behavioral equivalence abstracting from sequences/branches of exponentially timed internal actions?*

- *Will it be a congruence with respect to typical operators?*

- *Will it have a sound and complete axiomatization?*

- *Will it induce an exact CTMC-level aggregation?*

- *Any tradeoff among these properties?*

- Conduct the study in a process algebraic framework.

Definition of Weak Markovian Bisimilarity

- Basic idea: weaken the distinguishing power of $\sim_{\mathrm{MB}}$ by viewing every sequence of exponentially timed τ-actions as a single exponentially timed τ-action with the same average duration as the sequence.

- $\mathbb{P}_{\mathrm{M,s}}$: set of stable process terms, which can perform no exponentially timed τ-action.

- $\mathbb{P}_{\mathrm{M,u}}$: set of unstable process terms, which can perform at least one exponentially timed τ-action.

- $\mathbb{P}_{\mathrm{M,fu}}$: set of fully unstable process terms, which can perform only exponentially timed τ-actions (most natural candidates for abstraction).

- A computation c having the form $P_1 \xrightarrow{\tau,\lambda_1}_{\mathrm{M}} P_2 \xrightarrow{\tau,\lambda_2}_{\mathrm{M}} \ldots \xrightarrow{\tau,\lambda_n}_{\mathrm{M}} P_{n+1}$ is reducible iff $P_i \in \mathbb{P}_{\mathrm{M,fu}}$ for all $i = 1, \ldots, n$.

- Length-abstracting measure of a reducible computation c:

$$\boxed{probtime(c) \;=\; \left(\prod_{i=1}^{n} \frac{\lambda_i}{rate(P_i, \tau, \mathbb{P}_{\mathrm{M}})}\right) \cdot \left(\sum_{i=1}^{n} \frac{1}{rate(P_i, \tau, \mathbb{P}_{\mathrm{M}}))}\right)}$$

- The first factor is the product of the execution probabilities of the transitions of c.

- The second factor is the sum of the average sojourn times of the states traversed by c.

- Finite-length reducible computations are enough to distinguish between fully unstable process terms that must be told apart ($\lambda_1 \neq \lambda_2$ and $a \in Name_{\mathrm{v}}$):

$$<\tau, \lambda_1>.<a, \lambda>.P \quad \text{vs.} \quad <\tau, \lambda_2>.<a, \lambda>.P$$

$$\mathrm{rec}\, X : <\tau, \lambda_1>.X \quad \text{vs.} \quad \mathrm{rec}\, X : <\tau, \lambda_2>.X$$

- The weak variant of $\sim_{\mathrm{MB}}$ should work like $\sim_{\mathrm{MB}}$ over $\mathbb{P}_{\mathrm{M,nfu}}$ and abstract from the length of reducible computations while preserving their execution probability and average duration over $\mathbb{P}_{\mathrm{M,fu}}$.

- Need to lift measure *probtime* from a single reducible computation to a multiset of reducible computations with the same origin and destination:

$$\boxed{pbtm(P, D) \; = \; \{\!| \sum_{c \in redcomp(P,D,t)} probtime(c) \mid t \in \mathbb{R}_{>0} \, |\!\}}$$

  where $redcomp(P, D, t)$ is the multiset of reducible computations from $P \in \mathbb{P}_{\mathrm{M}}$ to $D \subseteq \mathbb{P}_{\mathrm{M}}$ whose average duration is $t \in \mathbb{R}_{>0}$.

- Measures must be summed up (otherwise $<\tau, \lambda_1>.\underline{0} + <\tau, \lambda_2>.\underline{0}$ not equivalent to $<\tau, \lambda_1 + \lambda_2>.\underline{0}$) only in case of equal average durations.

- An equivalence relation $\mathcal{B}$ over $\mathbb{P}_{\rm M}$ is a weak Markovian bisimulation iff, whenever $(P_1, P_2) \in \mathcal{B}$, then for all equivalence classes $D \in \mathbb{P}_{\rm M}/\mathcal{B}$:
  - If $P_1, P_2 \in \mathbb{P}_{\rm M,nfu}$, for all action names $a \in \mathit{Name}$:
  $$\mathit{rate}(P_1, a, D) \;=\; \mathit{rate}(P_2, a, D)$$
  - If $P_1, P_2 \in \mathbb{P}_{\rm M,fu}$, when $D \subseteq \mathbb{P}_{\rm M,nfu}$:
  $$\mathit{pbtm}(P_1, D) \;=\; \mathit{pbtm}(P_2, D)$$
- Weak Markovian bisimilarity, denoted $\approx_{\rm MB}$, is the union of all the weak Markovian bisimulations.

- **Example 1** – Consider the two process terms:

$$\bar{P}_1 \equiv <\tau,\mu>.<\tau,\gamma>.Q \qquad (\equiv <\tau,\gamma>.<\tau,\mu>.Q)$$

$$\bar{P}_2 \equiv <\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.Q$$

with $Q \in \mathbb{P}_{\mathrm{M,nfu}}$.

- Then $\bar{P}_1 \approx_{\mathrm{MB}} \bar{P}_2$ because:

$$\begin{aligned} pbtm(\bar{P}_1,[Q]_{\approx_{\mathrm{MB}}}) &= \{\!|\,(1\cdot 1)\cdot(\tfrac{1}{\mu}+\tfrac{1}{\gamma})\,|\!\} = \\ &= \{\!|\,1\cdot\tfrac{\mu+\gamma}{\mu\cdot\gamma}\,|\!\} = pbtm(\bar{P}_2,[Q]_{\approx_{\mathrm{MB}}}) \end{aligned}$$

- In general, for $l \in \mathbb{N}_{>0}$ we have:

$$<\tau,\mu>.<\tau,\gamma_1>.\dots.<\tau,\gamma_l>.Q \approx_{\mathrm{MB}} <\tau,\left(\frac{1}{\mu}+\frac{1}{\gamma_1}+\dots+\frac{1}{\gamma_l}\right)^{-1}>.Q$$

- **Example 2** – Consider the two process terms:

$$\bar{P}_3 \;\equiv\; {<}\tau, \mu{>}.({<}\tau, \gamma_1{>}.Q_1 + {<}\tau, \gamma_2{>}.Q_2)$$

$$\bar{P}_4 \;\equiv\; {<}\tau, \frac{\gamma_1}{\gamma_1+\gamma_2} \cdot \left(\frac{1}{\mu} + \frac{1}{\gamma_1+\gamma_2}\right)^{-1}{>}.Q_1 + {<}\tau, \frac{\gamma_2}{\gamma_1+\gamma_2} \cdot \left(\frac{1}{\mu} + \frac{1}{\gamma_1+\gamma_2}\right)^{-1}{>}.Q_2$$

  with $Q_1, Q_2 \in \mathbb{P}_{\mathrm{M,nfu}}$ and $Q_1 \not\approx_{\mathrm{MB}} Q_2$.

- Then $\bar{P}_3 \approx_{\mathrm{MB}} \bar{P}_4$ because:

$$pbtm(\bar{P}_3, [Q_1]_{\approx_{\mathrm{MB}}}) \;=\; \{\!| \frac{\gamma_1}{\gamma_1+\gamma_2} \cdot \left(\frac{1}{\mu} + \frac{1}{\gamma_1+\gamma_2}\right) |\!\} \;=\; pbtm(\bar{P}_4, [Q_1]_{\approx_{\mathrm{MB}}})$$

$$pbtm(\bar{P}_3, [Q_2]_{\approx_{\mathrm{MB}}}) \;=\; \{\!| \frac{\gamma_2}{\gamma_1+\gamma_2} \cdot \left(\frac{1}{\mu} + \frac{1}{\gamma_1+\gamma_2}\right) |\!\} \;=\; pbtm(\bar{P}_4, [Q_2]_{\approx_{\mathrm{MB}}})$$

- In general, for $n \in \mathbb{N}_{>0}$ we have:

$$\begin{aligned} {<}\tau, \mu{>}.({<}\tau, \gamma_1{>}.Q_1 + \ldots + {<}\tau, \gamma_n{>}.Q_n) \;\approx_{\mathrm{MB}}\; & {<}\tau, \frac{\gamma_1}{\gamma_1+\ldots+\gamma_n} \cdot \left(\frac{1}{\mu} + \frac{1}{\gamma_1+\ldots+\gamma_n}\right)^{-1}{>}.Q_1 \; + \\ & \ldots + \\ & {<}\tau, \frac{\gamma_n}{\gamma_1+\ldots+\gamma_n} \cdot \left(\frac{1}{\mu} + \frac{1}{\gamma_1+\ldots+\gamma_n}\right)^{-1}{>}.Q_n \end{aligned}$$

- **Example 3** – Consider the two process terms:

$$\bar{P}_5 \;\equiv\; <\tau, \mu_1>.<\tau, \gamma>.Q_1 + <\tau, \mu_2>.<\tau, \gamma>.Q_2$$

$$\bar{P}_6 \;\equiv\; <\tau, \frac{\mu_1}{\mu_1+\mu_2} \cdot \left(\frac{1}{\mu_1+\mu_2} + \frac{1}{\gamma}\right)^{-1}>.Q_1 + <\tau, \frac{\mu_2}{\mu_1+\mu_2} \cdot \left(\frac{1}{\mu_1+\mu_2} + \frac{1}{\gamma}\right)^{-1}>.Q_2$$

  with $Q_1, Q_2 \in \mathbb{P}_{\mathrm{M,nfu}}$ and $Q_1 \not\approx_{\mathrm{MB}} Q_2$.

- Then $\bar{P}_5 \approx_{\mathrm{MB}} \bar{P}_6$ because:

$$pbtm(\bar{P}_5, [Q_1]_{\approx_{\mathrm{MB}}}) \;=\; \{\!| \frac{\mu_1}{\mu_1+\mu_2} \cdot \left(\frac{1}{\mu_1+\mu_2} + \frac{1}{\gamma}\right) |\!\} \;=\; pbtm(\bar{P}_6, [Q_1]_{\approx_{\mathrm{MB}}})$$

$$pbtm(\bar{P}_5, [Q_2]_{\approx_{\mathrm{MB}}}) \;=\; \{\!| \frac{\mu_2}{\mu_1+\mu_2} \cdot \left(\frac{1}{\mu_1+\mu_2} + \frac{1}{\gamma}\right) |\!\} \;=\; pbtm(\bar{P}_6, [Q_2]_{\approx_{\mathrm{MB}}})$$

- In general, for $n \in \mathbb{N}_{>0}$ we have:

$$\begin{aligned} <\tau, \mu_1>.<\tau, \gamma>.Q_1 + \ldots + <\tau, \mu_n>.<\tau, \gamma>.Q_n \;\approx_{\mathrm{MB}}\; & <\tau, \frac{\mu_1}{\mu_1+\ldots+\mu_n} \cdot \left(\frac{1}{\mu_1+\ldots+\mu_n} + \frac{1}{\gamma}\right)^{-1}>.Q_1 + \\ & \ldots + \\ & <\tau, \frac{\mu_n}{\mu_1+\ldots+\mu_n} \cdot \left(\frac{1}{\mu_1+\ldots+\mu_n} + \frac{1}{\gamma}\right)^{-1}>.Q_n \end{aligned}$$

- **Example 4** – None of the variants of $\bar{P}_5$ related to actions $<\tau, \gamma>$ leads to a reduction.

- If we consider:

$$\begin{aligned}
\bar{P}_7 &\equiv <\tau, \mu_1>.<\tau, \gamma_1>.Q_1 + <\tau, \mu_2>.<\tau, \gamma_2>.Q_2 \\
\bar{P}_8 &\equiv <\tau, \tfrac{\mu_1}{\mu_1+\mu_2} \cdot \left(\tfrac{1}{\mu_1+\mu_2} + \tfrac{1}{\gamma_1}\right)^{-1}>.Q_1 + <\tau, \tfrac{\mu_2}{\mu_1+\mu_2} \cdot \left(\tfrac{1}{\mu_1+\mu_2} + \tfrac{1}{\gamma_2}\right)^{-1}>.Q_2
\end{aligned}$$

  with $\gamma_1 \neq \gamma_2$, then $\bar{P}_7 \not\approx_{\mathrm{MB}} \bar{P}_8$.

- If we consider:

$$\begin{aligned}
\bar{P}_9 &\equiv <\tau, \mu_1>.<\tau, \gamma>.Q_1 + <\tau, \mu_2>.Q_2 \\
\bar{P}_{10} &\equiv <\tau, \frac{\mu_1}{\mu_1+\mu_2} \cdot \left(\frac{1}{\mu_1+\mu_2} + \frac{1}{\gamma}\right)^{-1}>.Q_1 + <\tau, \mu_2>.Q_2
\end{aligned}$$

  then $\bar{P}_9 \not\approx_{\mathrm{MB}} \bar{P}_{10}$.

- Let:
  - $I \neq \emptyset$ be a finite index set.
  - $J_i \neq \emptyset$ be a finite index set for all $i \in I$.
  - $P_{i,j} \in \mathbb{P}_{\mathrm{M}}$ for all $i \in I$ and $j \in J_i$.

  Whenever $\sum_{j \in J_1} \gamma_{i_1,j} = \sum_{j \in J_2} \gamma_{i_2,j}$ for all $i_1, i_2 \in I$, then:

$$\sum_{i \in I} {<}\tau, \mu_i{>}. \sum_{j \in J_i} {<}\tau, \gamma_{i,j}{>}.P_{i,j} \;\approx_{\mathrm{MB}}$$

$$\sum_{i \in I} \sum_{j \in J_i} {<}\tau, \frac{\mu_i}{\sum\limits_{k \in I} \mu_k} \cdot \frac{\gamma_{i,j}}{\sum\limits_{h \in J_i} \gamma_{i,h}} \cdot \left(\frac{1}{\sum\limits_{k \in I} \mu_k} + \frac{1}{\sum\limits_{h \in J_i} \gamma_{i,h}} \right)^{-1} {>}.P_{i,j}$$

Congruence Property

- $\approx_{\rm MB}$ is a congruence with respect to action prefix and hiding.
- Let $P_1, P_2 \in \mathbb{P}_{\rm M}$. Whenever $P_1 \approx_{\rm MB} P_2$, then:
  - $<a, \lambda>.P_1 \approx_{\rm MB} <a, \lambda>.P_2$ for all $<a, \lambda> \in Act_{\rm M}$.
  - $P_1 / H \approx_{\rm MB} P_2 / H$ for all $H \subseteq Name_{\rm v}$.
- Not a congruence with respect to alternative and parallel composition due to fully unstable process terms:

$$<\tau, \mu>.<\tau, \gamma>.\underline{0} \ \approx_{\rm MB} \ <\tau, \tfrac{\mu \cdot \gamma}{\mu + \gamma}>.\underline{0}$$

but:

$$<\tau, \mu>.<\tau, \gamma>.\underline{0} + <a, \lambda>.\underline{0} \ \ \not\approx_{\rm MB} \ \ <\tau, \tfrac{\mu \cdot \gamma}{\mu + \gamma}>.\underline{0} + <a, \lambda>.\underline{0}$$

$$<\tau, \mu>.<\tau, \gamma>.\underline{0} \,\|_{\emptyset}\, <a, \lambda>.\underline{0} \ \ \not\approx_{\rm MB} \ \ <\tau, \tfrac{\mu \cdot \gamma}{\mu + \gamma}>.\underline{0} \,\|_{\emptyset}\, <a, \lambda>.\underline{0}$$

both for $a \neq \tau$ and for $a = \tau$.

- In order to avoid congruence violations for alternative composition, apply the exit rate equality check also to fully unstable process terms (but consider the equivalence classes with respect to $\approx_{\rm MB}$).

- We say that P_1 is weakly Markovian bisimulation congruent to P_2, written $P_1 \simeq_{\rm MB} P_2$, iff for all action names $a \in \mathit{Name}$ and equivalence classes $D \in \mathbb{P}_{\rm M}/\approx_{\rm MB}$:
$$\mathit{rate}(P_1, a, D) \; = \; \mathit{rate}(P_2, a, D)$$

- $\sim_{\rm MB} \subset \simeq_{\rm MB} \subset \approx_{\rm MB}$, with $\simeq_{\rm MB}$ and $\approx_{\rm MB}$ coinciding over $\mathbb{P}_{\rm M,nfu}$.

- $<a, \lambda>.P_1 \simeq_{\rm MB} <a, \lambda>.P_2$ iff $P_1 \approx_{\rm MB} P_2$.

- Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$. Whenever $P_1 \simeq_{\mathrm{MB}} P_2$, then:
  - $<a, \lambda>.P_1 \simeq_{\mathrm{MB}} <a, \lambda>.P_2$ for all $<a, \lambda> \in \mathit{Act}_{\mathrm{M}}$.
  - $P_1 + P \simeq_{\mathrm{MB}} P_2 + P$ and $P + P_1 \simeq_{\mathrm{MB}} P + P_2$ for all $P \in \mathbb{P}_{\mathrm{M}}$.
  - $P_1 / H \simeq_{\mathrm{MB}} P_2 / H$ for all $H \subseteq \mathit{Name}_{\mathrm{v}}$.
- $\simeq_{\mathrm{MB}}$ is the coarsest congruence contained in $\approx_{\mathrm{MB}}$ over the set $\mathbb{P}_{\mathrm{M,seq}}$ of process terms of $\mathbb{P}_{\mathrm{M}}$ that do not contain any occurrence of the parallel composition operator.
- Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M,seq}}$. Then $P_1 \simeq_{\mathrm{MB}} P_2$ iff $P_1 + P \approx_{\mathrm{MB}} P_2 + P$ for all $P \in \mathbb{P}_{\mathrm{M,seq}}$.

Sound and Complete Axiomatization

- $\simeq_{\rm MB}$ has a sound and complete axiomatization over the set $\mathbb{P}_{\rm M,seq,nrec}$ of nonrecursive process terms of $\mathbb{P}_{\rm M,seq}$.

- Set of basic axioms (the first four coincide with those of $\sim_{\rm MB}$):

$$
\begin{array}{lrcl}
(\mathcal{X}_{\rm MB,1}) & P_1 + P_2 & = & P_2 + P_1 \\
(\mathcal{X}_{\rm MB,2}) & (P_1 + P_2) + P_3 & = & P_1 + (P_2 + P_3) \\
(\mathcal{X}_{\rm MB,3}) & P + \underline{0} & = & P \\
(\mathcal{X}_{\rm MB,4}) & <a, \lambda_1>.P + <a, \lambda_2>.P & = & <a, \lambda_1 + \lambda_2>.P \\
(\mathcal{X}_{\rm MB,5}) & <a, \lambda>.\sum\limits_{i \in I} <\tau, \mu_i>.\sum\limits_{j \in J_i} <\tau, \gamma_{i,j}>.P_{i,j} & = & \\
& \multicolumn{3}{r}{<a, \lambda>.\sum\limits_{i \in I} \sum\limits_{j \in J_i} <\tau, \frac{\mu_i}{\mu} \cdot \frac{\gamma_{i,j}}{\gamma} \cdot \left(\frac{1}{\mu} + \frac{1}{\gamma}\right)^{-1}>.P_{i,j}}
\end{array}
$$

if: $I \neq \emptyset$ is a finite index set

$J_i \neq \emptyset$ is a finite index set for all $i \in I$

$\mu = \sum\limits_{i \in I} \mu_i$

$\gamma = \sum\limits_{j \in J_i} \gamma_{i,j}$ for all $i \in I$

- For proving completeness, we cannot resort to normal form saturation as this would alter the quantitative behavior.

- Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M,seq,nrec}}$. If $P_1 \approx_{\mathrm{MB}} P_2$ but $P_1 \not\simeq_{\mathrm{MB}} P_2$, then at least one between P_1 and P_2 (both of which must be fully unstable) is of the form:

$$\sum_{i \in I} {<}\tau, \mu_i{>}. \sum_{j \in J_i} {<}\tau, \gamma_{i,j}{>}.P_{i,j}$$

where $I \neq \emptyset$ is a finite index set, $J_i \neq \emptyset$ is a finite index set for all $i \in I$, and one of the following two properties holds:

  - $\sum_{j \in J_{i_1}} {<}\tau, \gamma_{i_1,j}{>}.P_{i_1,j} \approx_{\mathrm{MB}} \sum_{j \in J_{i_2}} {<}\tau, \gamma_{i_2,j}{>}.P_{i_2,j}$ for all $i_1, i_2 \in I$.
  - $\sum_{j \in J_{i_1}} \gamma_{i_1,j} = \sum_{j \in J_{i_2}} \gamma_{i_2,j}$ for all $i_1, i_2 \in I$.

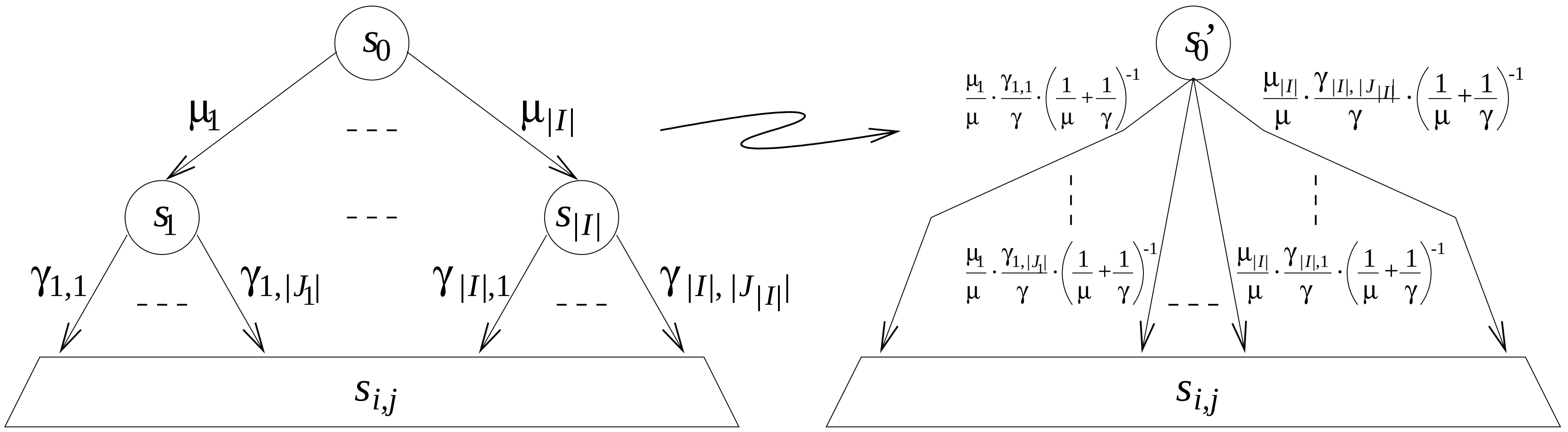
- Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M,seq,nrec}}$. Then $\mathcal{X}_{\mathrm{MB},1..5} \vdash P_1 = P_2 \iff P_1 \simeq_{\mathrm{MB}} P_2$.

Exactness of CTMC-Level Aggregation

- $\approx_{\mathrm{MB}}$ and $\simeq_{\mathrm{MB}}$ allow every sequence of exponentially timed τ-actions to be considered equivalent to a single exponentially timed τ-action having the same average duration.
- This amounts to approximating a hypoexponentially or Erlang distributed random variable with an exponentially distributed random variable having the same expected value.
- This can be exploited to assess more quickly properties expressed in terms of the mean time to certain events.
- *Is there any other performance property that is preserved?*

- Since $\sim_{\mathrm{MB}}$ is consistent with ordinary lumpability and the only new axiom is $\mathcal{X}_{\mathrm{MB},5}$, we can concentrate on this axiom.

- The induced CTMC-level aggregation, called W-lumpability, eliminates $|I|$ states and $|I|$ transitions by merging the first $1 + |I|$ states into a single one:



- W-lumpability is exact at steady state, i.e., the stationary probability of being in a macrostate of a CTMC obtained via W-lumpability is the sum of the stationary probabilities of being in one of the constituent microstates of the CTMC from which the reduced one has been derived.

- Unlike ordinary lumpability and T-lumpability, properties expressed in terms of transient state probabilities may not be preserved.

- Reconsider $\bar{P}_1 \equiv <\tau, \mu>.<\tau, \gamma>.Q$ and $\bar{P}_2 \equiv <\tau, \frac{\mu \cdot \gamma}{\mu + \gamma}>.Q$.

- The probability of being in the first state of $[\![\bar{P}_2]\!]_{\rm M}$ at time $t \in \mathbb{R}_{>0}$ is $1 - (1 - {\rm e}^{-\frac{\mu \cdot \gamma}{\mu + \gamma} \cdot t}) = {\rm e}^{-\frac{\mu \cdot \gamma}{\mu + \gamma} \cdot t}$, which reduces to ${\rm e}^{-\frac{\mu}{2} \cdot t}$ when $\mu = \gamma$.

- The sum of the probabilities of being in one of the first two states of $[\![\bar{P}_1]\!]_{\rm M}$ at the same time instant is $\frac{\gamma}{\gamma - \mu} \cdot {\rm e}^{-\mu \cdot t} - \frac{\mu}{\gamma - \mu} \cdot {\rm e}^{-\gamma \cdot t}$ for $\mu \neq \gamma$ or $(1 + \mu \cdot t) \cdot {\rm e}^{-\mu \cdot t}$ for $\mu = \gamma$.

Generalization to Concurrent Processes

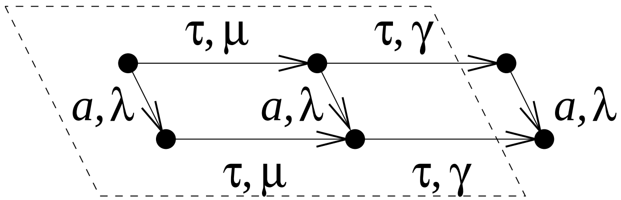
- Not being a congruence with respect to parallel composition significantly reduces the usefulness of $\simeq_{\mathrm{MB}}$ for compositional state space reduction.
- It is possible to modify the definition of the equivalence so that it becomes a congruence with respect to parallel composition ...
- ... but exactness will hold at steady state only for a certain class of processes.
- Revise the notion of reducible computation by admitting the traversal of unstable states (that are not fully unstable) satisfying certain conditions.
- Local computations may traverse fully unstable local states that are part of global states that are not fully unstable.

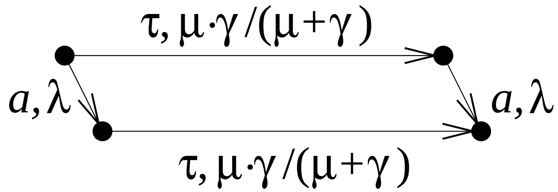
- The following two process terms:

$$<\tau, \mu>.<\tau, \gamma>.\underline{0} \parallel_{\emptyset} <a, \lambda>.\underline{0}$$

$$<\tau, \frac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} \parallel_{\emptyset} <a, \lambda>.\underline{0}$$

should be considered equivalent and should give rise to the following CTMC-level aggregation:





- Trees of sequences of exponentially timed τ-actions may be replicated.

- Take into account at once all the corresponding trees of computations and pinpoint their initial and final states.

- Let $P \in \mathbb{P}_{\mathrm{M}}$, $m, n \in \mathbb{N}_{>0}$, and $P_1, P_2, \ldots, P_m \in \mathbb{P}_{\mathrm{M}}$ reachable from P and different from each other.

- Let $C_1, C_2, \ldots, C_m$ be m sets each containing n distinct finite-length computations all starting from $P_1, P_2, \ldots, P_m$, respectively, with:
  - Each computation traversing different states except at most the last state and one of the preceding states.
  - Computations in different sets being disjoint.

- Assume that those computations can be partitioned into n groups each consisting of m computations from all the m sets, such that all the computations in the same group have the same length and are labeled with the same sequence of exponentially timed τ-actions.

- $C_k = \{c_{k,i} \mid c_{k,i} \equiv P_{k,i,1} \xrightarrow{\tau,\lambda_{i,1}}_{\mathrm{M}} P_{k,i,2} \xrightarrow{\tau,\lambda_{i,2}}_{\mathrm{M}} \ldots \xrightarrow{\tau,\lambda_{i,l_i}}_{\mathrm{M}} P_{k,i,l_i+1}, 1 \leq i \leq n\}$

- The family of computations $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$ is generally reducible iff either $m = 1$ and for all $i = 1, \dots, n$:
  - $P_{1,i,j} \in \mathbb{P}_{\mathrm{M,fu}}$ for all $j = 1, \dots, l_i$.
  - $P_{1,i,l_i+1} \in \mathbb{P}_{\mathrm{M,nfu}}$ or $P_{1,i,l_i+1} \equiv P_{1,i,j}$ for some $j = 1, \dots, l_i$.

  or $m \geq 1$, not all the states $P_{k,i,j}$, $1 \leq j \leq l_i$, are fully unstable when $m = 1$, and for all $i = 1, \dots, n$:
  - For all $k = 1, \dots, m$, $j = 1, \dots, l_i$, and $<a, \lambda> \in Act_{\mathrm{M}}$:
    1. Whenever $P_{k,i,j} \xrightarrow{a,\lambda}_{\mathrm{M}} P'$ with $P' \not\equiv P_{k,i,j+1}$, then:
       a) either $P' \equiv P_{k',i,j}$ for some $k' = 1, \dots, m$;
       b) or $P' \equiv P_{k,i',j'}$ with $a = \tau$ and $\lambda = \lambda_{i',j'-1}$ for some $i' = 1, \dots, n$ such that $i' \neq i$ and some $j' = 2, \dots, l_{i'+1}$.

2. For all $k' = 1, \ldots, m$, it holds that $P_{k,i,j} \xrightarrow{a,\lambda}_{\mathrm{M}} P_{k',i,j}$ iff $P_{k,i,j'} \xrightarrow{a,\lambda}_{\mathrm{M}} P_{k',i,j'}$ for all $j' = 1, \ldots, l_i$.
3. For all $i' = 1, \ldots, n$ such that $i' \neq i$ and $j' = 2, \ldots, l_{i'+1}$, it holds that $P_{k,i,j} \xrightarrow{a,\lambda}_{\mathrm{M}} P_{k,i',j'}$ iff $P_{k',i,j} \xrightarrow{a,\lambda}_{\mathrm{M}} P_{k',i',j'}$ for all $k' = 1, \ldots, m$.

– One of the following holds:

$\widetilde{4}$. There is no $\lambda_{i,l_i+1} \in \mathbb{R}_{>0}$ such that $P_{k,i,l_i+1} \xrightarrow{\tau,\lambda_{i,l_i+1}}_{\mathrm{M}} P_{k,i,l_i+2}$ for all $k = 1, \ldots, m$.

$\overline{4}$. If there exists $\lambda_{i,l_i+1} \in \mathbb{R}_{>0}$ such that $P_{k,i,l_i+1} \xrightarrow{\tau,\lambda_{i,l_i+1}}_{\mathrm{M}} P_{k,i,l_i+2}$ for all $k = 1, \ldots, m$, then at least one of conditions 1, 2, and 3 above is not satisfied by P_{k',i,l_i+1} for some $k' = 1, \ldots, m$.

$\widehat{4}$. $P_{k,i,l_i+1} \equiv P_{k,i,j}$ for all $k = 1, \ldots, m$ and some $j = 1, \ldots, l_i$.

- $source(\mathcal{C}) = \{P_k \mid 1 \le k \le m\}$: set of initial states of $\mathcal{C}$.
- $target(\mathcal{C}) = \{P_{k,i,l_i+1} \mid 1 \le k \le m, 1 \le i \le n\}$: set of final states of $\mathcal{C}$.
- In order to avoid interferences among computations in $C_1, C_2, \dots, C_m$ and computations across $C_1, C_2, \dots, C_m$, we redefine:

$$\boxed{probtime(c_{k,i}) = (\prod_{j=1}^{l_i} \frac{\lambda_{i,j}}{rate(P_{k,i,j},\tau,\mathcal{P}'_k)}) \cdot (\sum_{i=1}^{l_i} \frac{1}{rate(P_{k,i,j},\tau,\mathcal{P}'_k)})}$$

  where $\mathcal{P}'_k = \{P_{k,i',j'} \mid 1 \le i' \le n, 2 \le j' \le l_{i'+1}\}$.

- We redefine $reducomp(P_k, D, t)$ as the multiset of computations identical to those in C_k that go from P_k to D and have average duration t.
- If $\mathcal{C}$ is generally reducible, then $probtime(c_{k,i}) = probtime(c_{k',i})$ and $pbtm(P_k, target(\mathcal{C})) = pbtm(P_{k'}, target(\mathcal{C}))$.

- An equivalence relation $\mathcal{B}$ over $\mathbb{P}_{\mathrm{M}}$ is a g-weak Markovian bisimulation iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
  - For all visible action names $a \in \mathit{Name}_{\mathrm{v}}$ and equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\mathcal{B}$:
  $$\mathit{rate}(P_1, a, D) \;=\; \mathit{rate}(P_2, a, D)$$
  - If P_1 is not an initial state of any g-reducible family of computations, then P_2 is not an initial state of any g-reducible family of computations either, and for all equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\mathcal{B}$:
  $$\mathit{rate}(P_1, \tau, D) \;=\; \mathit{rate}(P_2, \tau, D)$$
  - If P_1 is an initial state of some g-reducible family of computations, then P_2 is an initial state of some g-reducible family of computations too, and for all g-reducible families of computations $\mathcal{C}_1$ with $P_1 \in \mathit{source}(\mathcal{C}_1)$ there exists a g-reducible family of computations $\mathcal{C}_2$ with $P_2 \in \mathit{source}(\mathcal{C}_2)$ such that for all equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\mathcal{B}$:
  $$\mathit{pbtm}(P_1, D \cap \mathit{target}(\mathcal{C}_1)) \;=\; \mathit{pbtm}(P_2, D \cap \mathit{target}(\mathcal{C}_2))$$
- G-weak Markovian bisimilarity, denoted $\approx_{\mathrm{MB,g}}$, is the union of all the g-weak Markovian bisimulations.

- All the examples that we have seen before for $\approx_{\mathrm{MB}}$ are valid for $\approx_{\mathrm{MB,g}}$, because a tree of computations reducible in the sense of the original definition forms a g-reducible family of computations.

- Unlike $\approx_{\mathrm{MB}}$, it turns out that $\approx_{\mathrm{MB,g}}$ is a congruence with respect to parallel composition too.

- Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$. Whenever $P_1 \approx_{\mathrm{MB,g}} P_2$, then:
  - $<a, \lambda>.P_1 \approx_{\mathrm{MB,g}} <a, \lambda>.P_2$ for all $<a, \lambda> \in Act_{\mathrm{M}}$.
  - $P_1 \parallel_S P \approx_{\mathrm{MB,g}} P_2 \parallel_S P$ and $P \parallel_S P_1 \approx_{\mathrm{MB,g}} P \parallel_S P_2$ for all $S \subseteq Name_{\mathrm{v}}$ and $P \in \mathbb{P}_{\mathrm{M}}$.
  - $P_1/H \approx_{\mathrm{MB,g}} P_2/H$ for all $H \subseteq Name_{\mathrm{v}}$.

- Like $\approx_{\mathrm{MB}}$, we have that $\approx_{\mathrm{MB,g}}$ is not a congruence with respect to alternative composition either.

- We say that P_1 is g-weakly Markovian bisimulation congruent to P_2, written $P_1 \simeq_{\mathrm{MB,g}} P_2$, iff for all action names $a \in \mathit{Name}$ and equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\approx_{\mathrm{MB,g}}$:

$$\mathit{rate}(P_1, a, D) = \mathit{rate}(P_2, a, D)$$

- Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$. Whenever $P_1 \simeq_{\mathrm{MB,g}} P_2$, then:
  - $<a, \lambda>.P_1 \simeq_{\mathrm{MB,g}} <a, \lambda>.P_2$ for all $<a, \lambda> \in \mathit{Act}_{\mathrm{M}}$.
  - $P_1 + P \simeq_{\mathrm{MB,g}} P_2 + P$ and $P + P_1 \simeq_{\mathrm{MB,g}} P + P_2$ for all $P \in \mathbb{P}_{\mathrm{M}}$.
  - $P_1 \,\|_S\, P \simeq_{\mathrm{MB,g}} P_2 \,\|_S\, P$ and $P \,\|_S\, P_1 \simeq_{\mathrm{MB,g}} P \,\|_S\, P_2$ for all $S \subseteq \mathit{Name}_{\mathrm{v}}$ and $P \in \mathbb{P}_{\mathrm{M}}$.
  - $P_1/H \simeq_{\mathrm{MB,g}} P_2/H$ for all $H \subseteq \mathit{Name}_{\mathrm{v}}$.

- Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$. Then $P_1 \simeq_{\mathrm{MB,g}} P_2$ iff $P_1 + P \approx_{\mathrm{MB,g}} P_2 + P$ for all $P \in \mathbb{P}_{\mathrm{M}}$.

- The CTMC-level aggregation induced by $\approx_{\mathrm{MB,g}}$ and $\simeq_{\mathrm{MB,g}}$ is exact at steady-state only for those process terms with a restricted use of synchronization.

- This limitation stems from insensitivity conditions for GSMPs (with GSMPs coming into play due to the reduction of sequences of exponentially timed τ-transitions) and emphasizes a tradeoff between achieving compositionality over concurrent processes and preserving exactness at steady state.

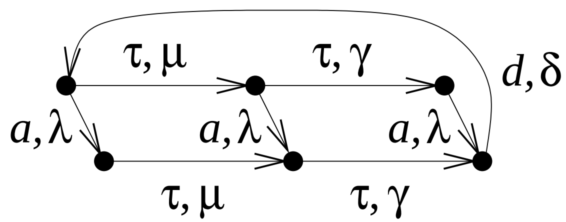
- GW-lumpability is exact at steady state over each process term $P \in \mathbb{P}_{\mathrm{M}}$ such that, for all g-reducible families of computations $\mathcal{C}$ in $[\![P]\!]_{\mathrm{M}}$ with size $m \geq 2$ or size $m = 1$ and not all the traversed states being fully unstable, no transition in $[\![P]\!]_{\mathrm{M}}$ arising from action synchronization has an element of $\mathit{source}(\mathcal{C})$ as its target state.

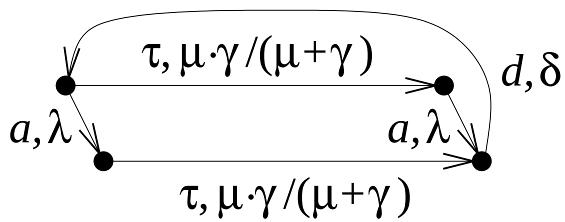
- **Example 5** – Consider the two process terms with synchronization:

$$\text{rec}\, X : {<}\tau, \mu{>}.{<}\tau, \gamma{>}.{<}d, \delta{>}.X \;\|_{\{d\}}\; \text{rec}\, Y : {<}a, \lambda{>}.{<}d, *_w{>}.Y$$

$$\text{rec}\, X : {<}\tau, \tfrac{\mu \cdot \gamma}{\mu + \gamma}{>}.{<}d, \delta{>}.X \;\|_{\{d\}}\; \text{rec}\, Y : {<}a, \lambda{>}.{<}d, *_w{>}.Y$$

- Resulting CTMC-level aggregation:





- Not exact due to the following steady-state probabilities $(\mu = \gamma = \lambda = \delta = 1)$:

2/13 1/13 1/13 2/10 1/10

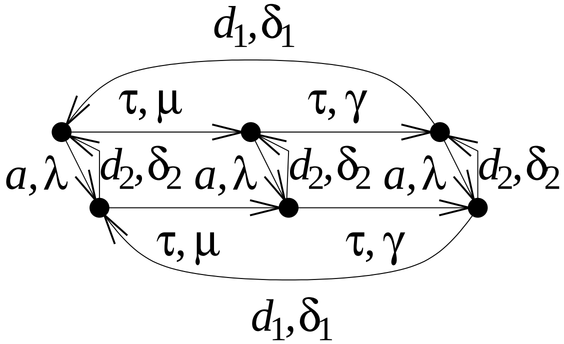
2/13 3/13 4/13 4/10 3/10

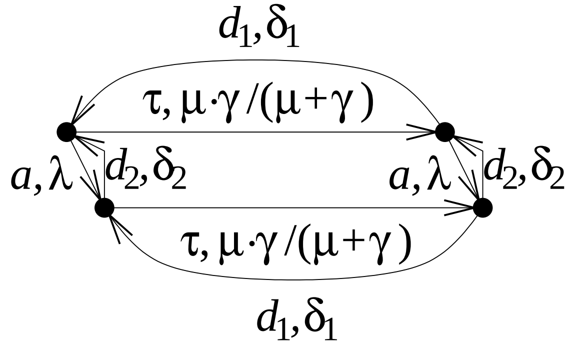
- **Example 6** – Consider the two process terms without synchronization:

$$\mathrm{rec}\, X : <\tau, \mu>.<\tau, \gamma>.<d_1, \delta_1>.X \parallel_{\emptyset} \mathrm{rec}\, Y : <a, \lambda>.<d_2, \delta_2>.Y$$

$$\mathrm{rec}\, X : <\tau, \frac{\mu \cdot \gamma}{\mu + \gamma}>.<d_1, \delta_1>.X \parallel_{\emptyset} \mathrm{rec}\, Y : <a, \lambda>.<d_2, \delta_2>.Y$$

- Resulting CTMC-level aggregation:





- Exact due to the following steady-state probabilities $(\mu = \gamma = \lambda = \delta_1 = \delta_2 = 1)$:

1/6 1/6 1/6 2/6 1/6

1/6 1/6 1/6 2/6 1/6

Related and Future Work

- Problem originally addressed in [Hil1996] through a relation called weak isomorphism, from which we have taken the idea of preserving the average duration of internal action sequences.

- Congruence and steady-state exactness of weak isomorphism have been investigated, but no axiomatization is known (too strong, no branches).

- Different approach proposed in [Bra2002], where a variant of Markovian bisimilarity is defined that checks for exit rate equality with respect to all equivalence classes apart from the one including the process terms under examination.

- Congruence and axiomatization results have been provided, but nothing is said about exactness.

- Axiomatization of $\simeq_{\mathrm{MB,g}}$, modal logic characterization, algorithms.