

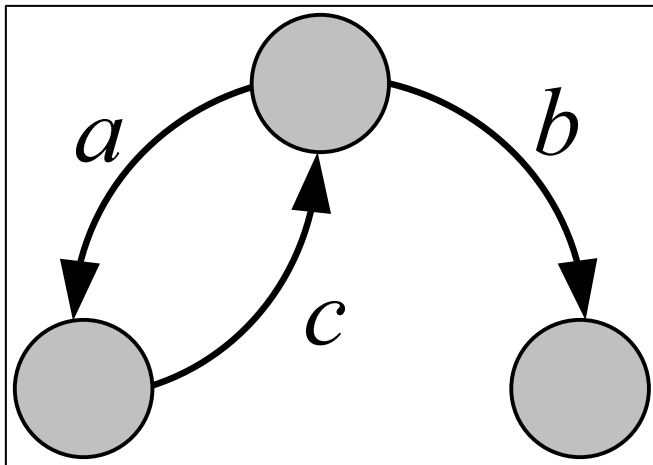
# On Compositionality, Efficiency, and Applicability of Abstraction in Probabilistic Systems

Suzana Andova  
Sonja Georgievska

Eindhoven University of Technology, The Netherlands

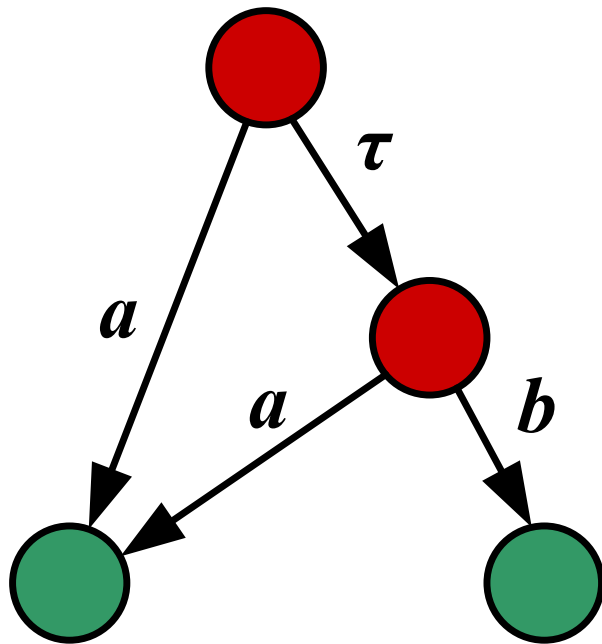
# Labeled Transition Systems

- Formalism for modeling qualitative (functional) behavior
- Directed graphs:
  - nodes = states of the system
  - labels on edges = actions that the system can perform



# Branching Bisimulation Equivalence

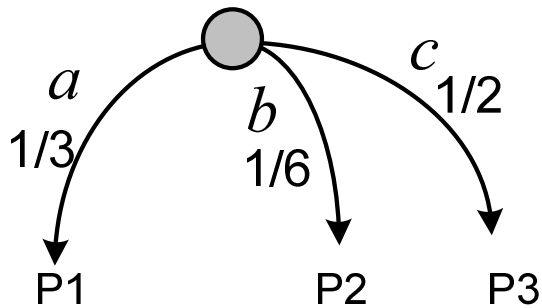
- Equates states with the same action potential
- Preserves branching structure
- Abstracts from internal ( $\tau$ -labeled) transitions



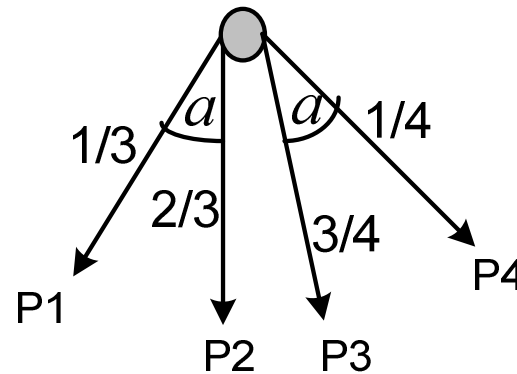
Same colour –  
branching bisimilar states

# Adding Probabilities

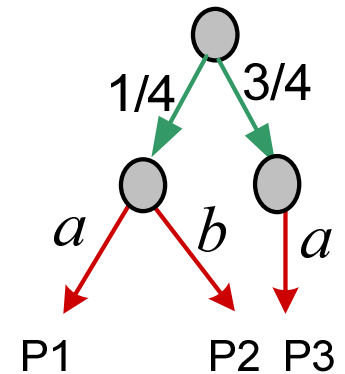
- To model quantitative aspects of systems
- Several existing models



Fully probabilistic



Segala



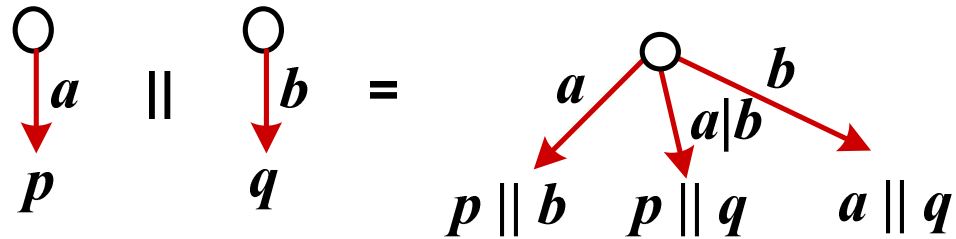
Alternating

- Further refinements:
  - reactive, generative
  - strictly alternating, non-strictly alternating
  - stratified models

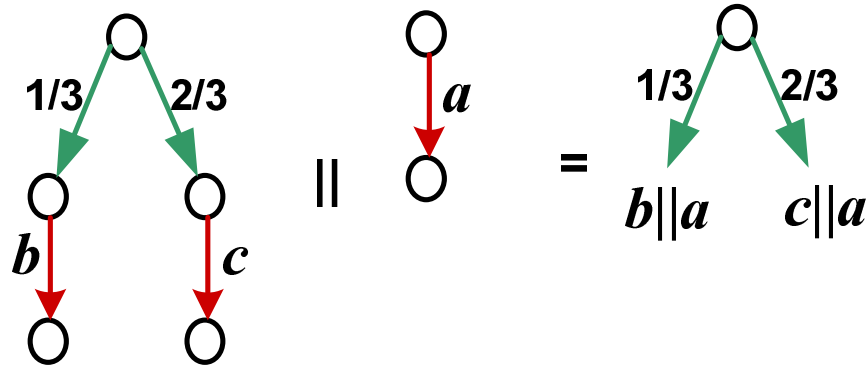
# Alternating model: weak equivalences

- Weak bisimulation [Philippou/Lee/Sokolsky '00]
- Branching bisimulation [Andova/Willemse '06]
- Branching bisimulation congruence [Trcka/Georgievska '08]
  - The coarsest congruence for parallel composition included in [Andova/Willemse '06]

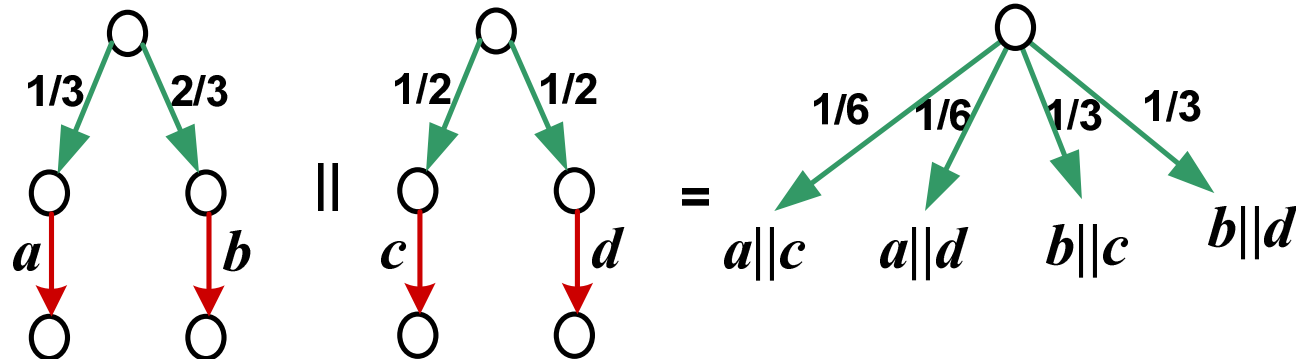
# Parallel composition



Actions: standard

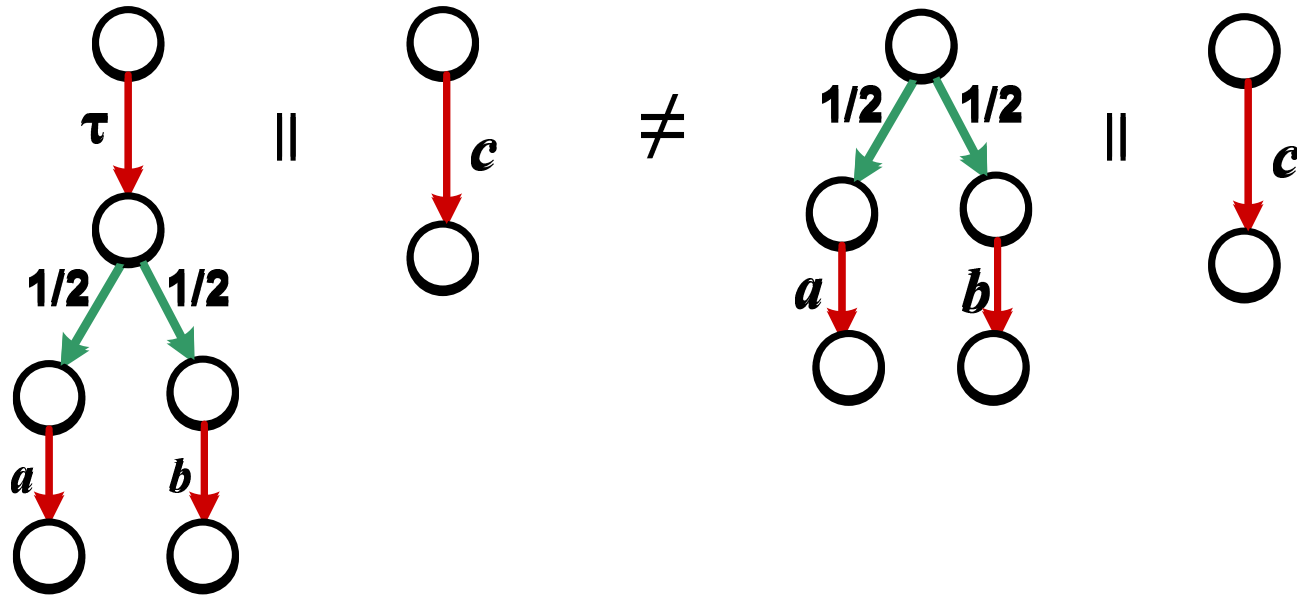


Probabilistic choice resolved first



Probabilistic choices synchronized

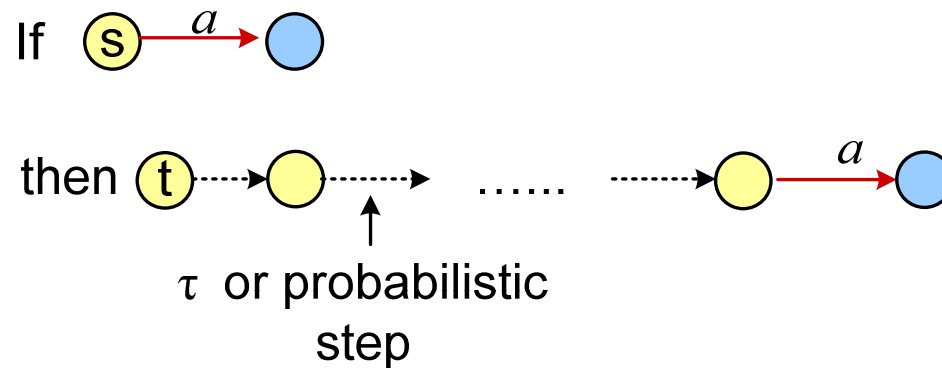
# What was the congruence problem



# Branching bisimulation congruence is...

...any equivalence  $R$  on the (finite) set of states such that  $s R t$  iff

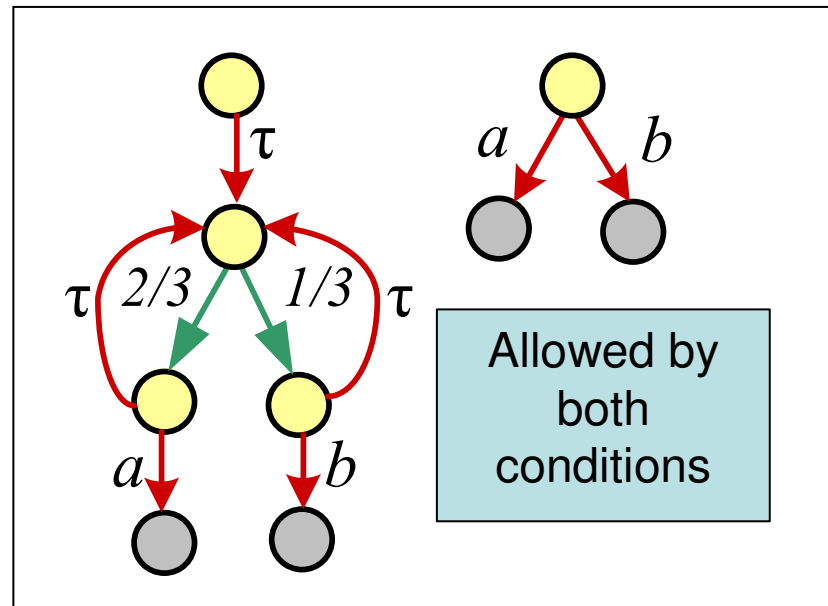
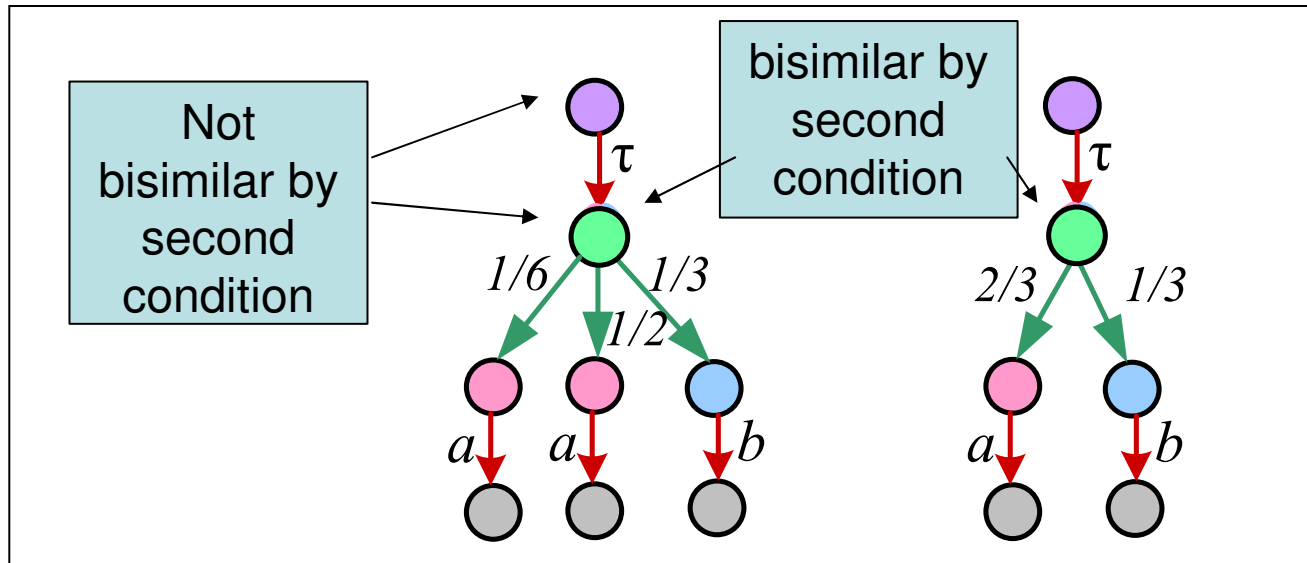
1. First condition:



2. Second condition: for any  $R$ -class  $D$ ,  $P(s,D) = P(t,D)$

- $P(\text{state}, D) =$ 
  - 1 - if **state** is nondeterministic and **state**  $\in D$
  - probability that **state** enters some state in  $D$  in one step - if **state** is probabilistic
  - 0, otherwise

# Examples of bisimilarity ( $\sim$ )



## In this work we...

- ...investigate further the compositional branching bisimilarity  $\sim$ . We
  - give a “ $\sim$ -classes” computing algorithm of polynomial time complexity w.r.t the number of states
  - present a probabilistic version of CTL-X and prove that  $\sim$  preserves the formulas
  - give axiomatic characterization for finite processes and several sound recursive rules
  - use the axioms and the rules to verify compositionally the Concurrent Alternating Bit Protocol with lossy channels

# Computing the classes of $\sim$

1. Satisfying the **first** condition for  $\sim$ 
  - treat probabilistic steps as  $\tau$ -steps
  - use the **standard algorithm for branching bisimilarity partitioning** [Groote/Vaandrager'90] to compute current partitioning  $\Pi$
2. Satisfying the **second** condition for  $\sim$ 
  - Find Class1 and Class2 such that
    - for some  $s, t \in \text{Class1}$  it holds  $P(s, \text{Class2}) \neq P(t, \text{Class2})$
  - Split Class1 so that Class2 cannot split it further
  - Refine the current partitioning  $\Pi$  accordingly
3. Repeat steps 1 and 2 until no changes of  $\Pi$

# Computing the classes of $\sim$

1. Satisfying the **first** condition for  $\sim$ 
  - treat probabilistic steps as  $\tau$ -steps
  - use the **standard algorithm for branching bisimilarity partitioning** [Groote/Vaandrager'90] to compute current partitioning  $\Pi$

Theorem:

$\sim$  is decidable in polynomial time (of the number of states)

- Refine the current partitioning  $\Pi$  accordingly
3. Repeat steps 1 and 2 until no changes of  $\Pi$

# pCTL-X

- Starting point pCTL [Baier/Kwiatkowska'98, Bianco/Alfaro'95]
  - Add semantics to the setting with  $\tau$ -steps
- First: translate edge-labeled systems into node&edge-labeled (Kripke structures)
- Extend standard non-probabilistic translation to probabilistic systems:
  - Split the **observable** transitions
  - Label the **new** states with the action they represent
  - Label **all** old states with a new symbol (e.g. \$)



# pCTL-X

- Starting point pCTL [Baier/Kwiatkowska'98, Bianco/Alfaro'95]
  - Add semantics to the setting with  $\tau$ -steps
- First: translate edge-labeled systems into node&edge-labeled (Kripke structures)

Theorem:  
Branching bisimilarity is preserved by the translation



## pCTL-X (cont.)

- Grammar of pCTL-X:
- $\Psi := \$ \mid a \mid \neg\Psi \mid \Psi \wedge \Psi \mid \exists \text{Prob}_{\#p}(\Psi \cup \Psi)$ 
  - $a \in \text{Actions} \setminus \{\tau\}$ ,  $p \in [0, 1]$ ,  $\# \in \{\leq, \geq, >, <\}$
  - $\mathbf{s}$  satisfies  $\exists \text{Prob}_{\#p}(\Psi \cup \Psi)$  iff there exists a **scheduler**  $\sigma$  for which the probability measure of the set of all paths in the  $\sigma$  **computation tree** that
    - start in  $\mathbf{s}$
    - satisfy  $\Psi \cup \Psi$is  $\#p$ .

## pCTL-X (cont.)

- Grammar of pCTL-X:
- $\Psi := \$ \mid a \mid \neg\Psi \mid \Psi \wedge \Psi \mid \exists \text{Prob}_{\#p}(\Psi \cup \Psi)$ 
  - $a \in \text{Actions} \setminus \{\tau\}$ ,  $p \in [0, 1]$ ,  $\# \in \{\leq, \geq, >, <\}$
  - $\mathbf{s}$  satisfies  $\exists \text{Prob}_{\#p}(\Psi \cup \Psi)$  iff there exists a

Theorem:  
Branching bisimilar states satisfy same pCTL-X formulas.

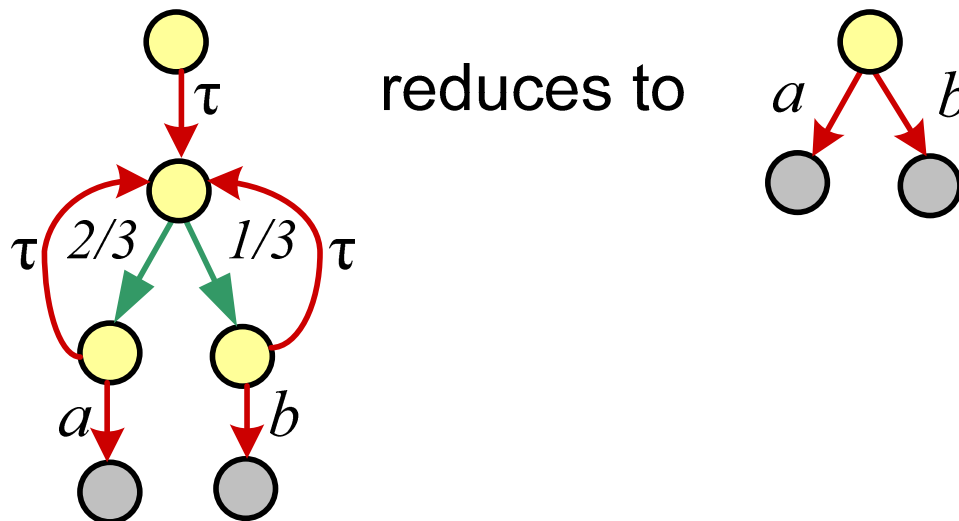
is #p.

# The algebra $\text{pACP}_\tau$

- Extension of ACP (Algebra of Communicating Processes)
  - **Standard operators:** sequential composition ( $\cdot$ ), alternative composition ( $+$ ), parallel composition ( $\parallel$ ), encapsulation, guarded recursion
  - **Added operators:** probabilistic choice ( $\oplus_\pi$ ), action hiding
- **One** axiom characterizes finite processes with  $\tau$ -steps
  - $x.((y + \tau.(y + z)) \oplus_\pi w) = x.((y + z) \oplus_\pi w)$ 
    - if  $y$  and  $z$  are not initially probabilistic
- $\text{ACP}_\tau$  counterpart:  $x.(y + \tau.(y + z)) = x.(y + z)$
- No axiom of type  $x. \tau = x$

# Recursive rules

- Developed initially for the weaker branching bisimulation [Andova/Baeten/Willemse'06]
- Eliminating loops of unobservable transitions from process terms (of which  $\parallel$  has been eliminated)
- A simple example



# Verification of CABP

- CABP: reliable transmission over unreliable channels
- The sender continuously resends same data until acknowledgement from the receiver is received
  - Data can be lost/corrupted with certain probabilities
- The model consists of 6 communicating modules
- Compositional verification
  - Hide the internal behaviour of the components
  - Use  $\sim$  to reduce (verify) the components
  - Put the components in parallel
  - Use  $\sim$  to reduce (verify) the result
  - Result: With a probability 1 CABP acts as a buffer

# Related work & Conclusion

- Andova/Baeten/Willemse 2006
  - Ground-complete axiomatization of the branching bisimulation of [Andova/Willemse2006] and verification of CABP
  - without parallel composition operator
- Non-alternating model (simple Segala automata)
  - weak equivalences decidable in exponential time
    - computing convex closures on sets
  - similarities with our model:  $\tau$  “guards” the probabilistic choice
- To conclude: we have shown that the branching bisimulation congruence considered here...
  - Is decidable in polynomial time
  - Preserves the formulas of a probabilistic extension of CTL-X
  - Can be used in compositional verification of transmission protocols as CABP

Thank you...  
Questions?

---

$PVR_1^*$	$x = y \uplus_{\pi} i \cdot x$ <span style="border: 1px solid red; padding: 2px; margin-left: 20px;"><math>y = y + y</math></span> <span style="margin-left: 20px;"><math>i \in I</math></span>	
	$\tau \cdot \tau_I(x) = \tau \cdot \tau_I(y)$	
$VR_1$	$x = y + i \cdot x$ <span style="margin-left: 40px;"><math>y = y + y</math></span> <span style="margin-left: 40px;"><math>i \in I</math></span>	
	$\tau \cdot \tau_I(x) = \tau \cdot \tau_I(y)$	
$PVR_3$	$x = (z \uplus_{\pi} y) + i \cdot x$ <span style="margin-left: 40px;"><math>y = y + y</math></span> <span style="margin-left: 40px;"><math>i \in I</math></span>	
	$\tau \cdot \tau_I(x) = \tau \cdot \tau_I(x')$ for $x' = z + y + i \cdot x'$	
$PKR_{n \geq 1}^b$	$x = y_0 + i_0 \cdot x_1$ <span style="margin-left: 20px;"><math>x_1 = y_1 + i_1 \cdot x_2</math></span> <span style="margin-left: 20px;"><math>\dots</math></span> <span style="margin-left: 20px;"><math>x_n = y_n + i_n \cdot x</math></span> <span style="margin-left: 20px;"><math>i_k \in I, \exists i_j \neq \tau</math></span>	
	$\tau \cdot \tau_I(x) = \tau \cdot \tau_I(x')$ for $x' = y_0 + y_1 \dots y_n + i_0 \cdot x'$	
$VR_2$	$x = z \uplus_{\pi} (u + i \cdot x)$ <span style="margin-left: 40px;"><math>z = z + u</math></span> <span style="margin-left: 40px;"><math>z = z + z</math></span> <span style="margin-left: 40px;"><math>i \in I</math></span>	
	$\tau \cdot \tau_I(x) = \tau \cdot \tau_I(z)$	
$VR_3$	$x = z + i \cdot y$ <span style="margin-left: 40px;"><math>y = z \uplus_{\pi} (u + j \cdot x)</math></span> <span style="margin-left: 40px;"><math>z = z + z</math></span> <span style="margin-left: 40px;"><math>z = z + u</math></span> <span style="margin-left: 40px;"><math>i, j \in I</math></span>	
	$\tau \cdot \tau_I(x) = \tau \cdot \tau_I(y')$ for $y' = z \uplus_{\pi} (u + i \cdot y')$	

restriction of the original rule in [Andova/Baeten/Willemse]

---

**Table 1.** Axioms of  $\text{pACP}_\tau$ .  $a, b \in A \cup \{\delta\}$ ,  $I, H \subseteq A \setminus \{\tau\}$ ,  $w, x, y, z \in V$

---

<p>A1 <math>x + y = y + x</math></p> <p>A2 <math>(x + y) + z = x + (y + z)</math></p> <p>AA3 <math>a + a = a</math></p> <p>A4 <math>(x + y) \cdot z = x \cdot z + y \cdot z</math></p> <p>A5 <math>x \cdot (y \cdot z) = (x \cdot y) \cdot z</math></p> <p>A6 <math>x + \delta = x</math></p> <p>A7 <math>\delta \cdot x = \delta</math></p> <p>TI1 <math>\tau_I(a) = a</math> if <math>a \notin I</math></p> <p>TI1' <math>\tau_I(a) = \tau</math> if <math>a \in I</math></p> <p>TI2 <math>\tau_I(x + y) = \tau_I(x) + \tau_I(y)</math></p> <p>TI3 <math>\tau_I(x \cdot y) = \tau_I(x) \cdot \tau_I(y)</math></p> <p>TI4 <math>\tau_I(x \boxplus_\pi y) = \tau_I(x) \boxplus_\pi \tau_I(y)</math></p>	<p>PA1 <math>x \boxplus_\pi y = y \boxplus_{1-\pi} x</math></p> <p>PA2 <math>x \boxplus_\pi (y \boxplus_\rho z) = (x \boxplus_{\frac{\pi}{\pi+\rho-\pi\rho}} y) \boxplus_{\pi+\rho-\pi\rho} z</math></p> <p>PA3 <math>x \boxplus_\pi x = x</math></p> <p>PA4 <math>(x \boxplus_\pi y) \cdot z = x \cdot z \boxplus_\pi y \cdot z</math></p> <p>PA5 <math>(x \boxplus_\pi y) + z = (x + z) \boxplus_\pi (y + z)</math></p>
<p>M <math>x \parallel y = x \llcorner y + y \llcorner x + x \mid y</math> if</p>	<p>D1 <math>\partial_H(a) = a</math> if <math>a \notin H</math></p> <p>D2 <math>\partial_H(a) = \delta</math> if <math>a \in H</math></p> <p>D3 <math>\partial_H(x + y) = \partial_H(x) + \partial_H(y)</math></p> <p>D4 <math>\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)</math></p> <p>D5 <math>\partial_H(x \boxplus_\pi y) = \partial_H(x) \boxplus_\pi \partial_H(y)</math></p>
<p>PM1 <math>x \parallel (y \boxplus_\pi z) = (x \parallel y) \boxplus_\pi (x \parallel z)</math></p> <p>PM2 <math>(x \boxplus_\pi y) \parallel z = (x \parallel z) \boxplus_\pi (y \parallel z)</math></p> <p>LM2 <math>a \llcorner x = a \cdot x</math></p> <p>LM3 <math>a \cdot x \llcorner y = a \cdot (x \parallel y)</math></p> <p>LM4 <math>(x + y) \llcorner z = x \llcorner z + y \llcorner z</math></p>	<p>CF <math>a \mid b = \gamma(a, b)</math>, if <math>\gamma(a, b)</math> defined <math>a \mid b = \delta</math>, otherwise</p> <p>CM1 <math>x \mid (y + z) = x \mid y + x \mid z</math></p> <p>CM2 <math>(x + y) \mid z = x \mid z + y \mid z</math></p> <p>CM3 <math>a \mid b \cdot x = (a \mid b) \cdot x</math></p> <p>CM4 <math>a \cdot x \mid b = (a \mid b) \cdot x</math></p> <p>CM5 <math>a \cdot x \mid b \cdot y = (a \mid b)(x \parallel y)</math></p>
<p>PrB <math>x \cdot ((y + \tau \cdot (y + z)) \boxplus_\pi w) = x \cdot ((y + z) \boxplus_\pi w)</math> if <math>y = y + y</math> and <math>z = z + z</math></p>	

---