Modeling & Verification

Weak Bisimilarity

Max Tschaikowski (tschaikowski@cs.aau.dk)
Slides courtesy of Giorgio Bacci

in the last Lecture

- Value-passing CCS
- Behavioural Equivalences (idea & motivations)
- Strong Bisimilarity
- Game characterisation of Bisimilarity

in this Lecture

- Properties of Strong Bisimilarity (review)
- Example: Buffer implementation in CCS
- Weak Bisimilarity (Properties & Game characterisation)
- Tool: Concurrency Workbench Aalborg Edition (CAAL)

Strong Bisimilarity

Let (Proc, Act, $\{\xrightarrow{\alpha} \mid \alpha \in \text{Act}\}$) be an LTS.

Definition (Strong Bisimulation)

A binary relation $R \subseteq \text{Proc} \times \text{Proc}$ is a *strong bisimulation* iff whenever s R t, for each $\alpha \in \text{Act}$

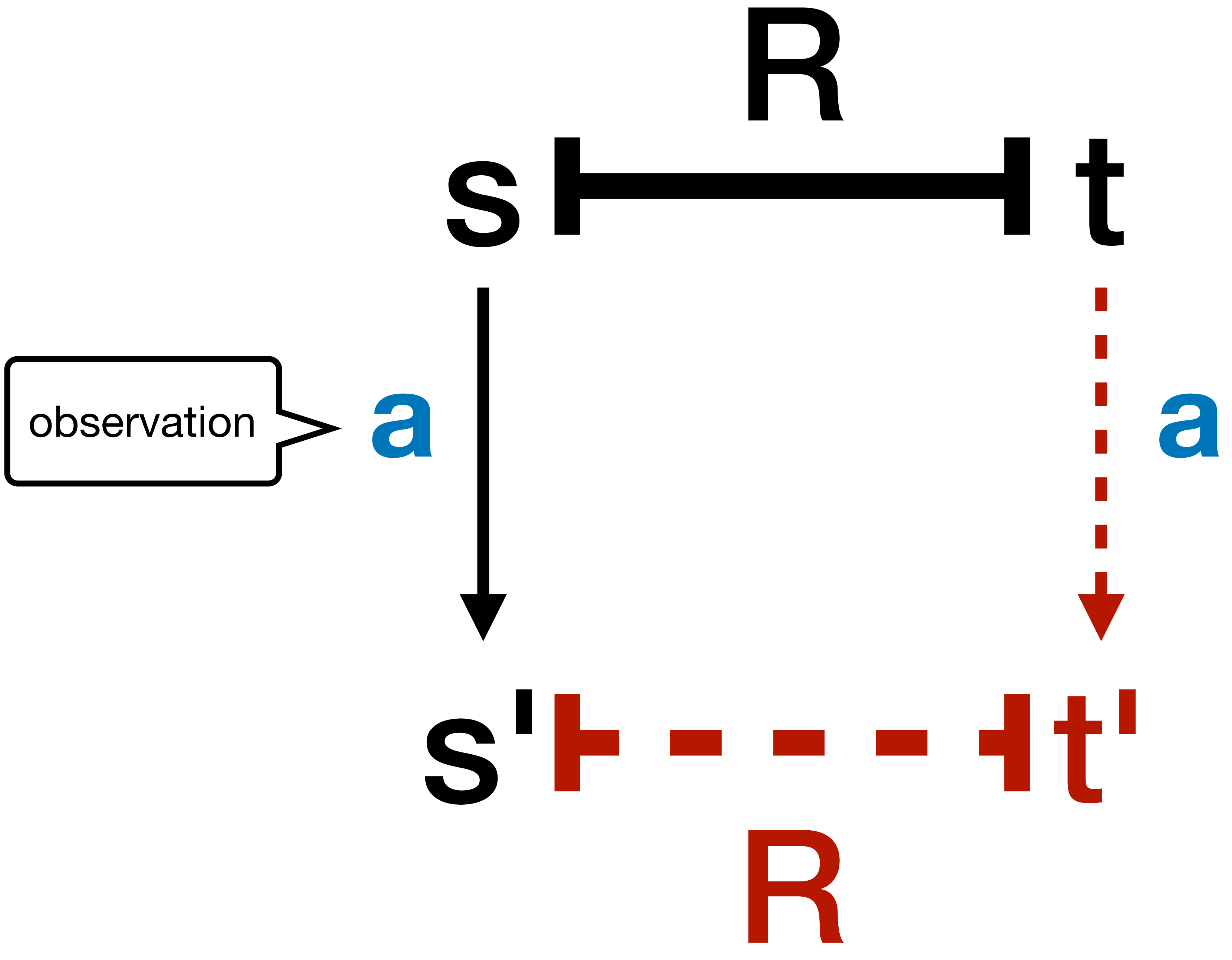
- if $s \xrightarrow{\alpha} s'$, then $t \xrightarrow{\alpha} t'$, for some t' such that s'R t'
- if $t \xrightarrow{\alpha} t'$, then $s \xrightarrow{\alpha} s'$, for some s' such that s'R t'

Definition (Strong Bisimilarity)

Two states $s,t \in \text{Proc}$ are *strongly bisimilar* (s ~ t) iff there exists a strong bisimulation R such that s R t.

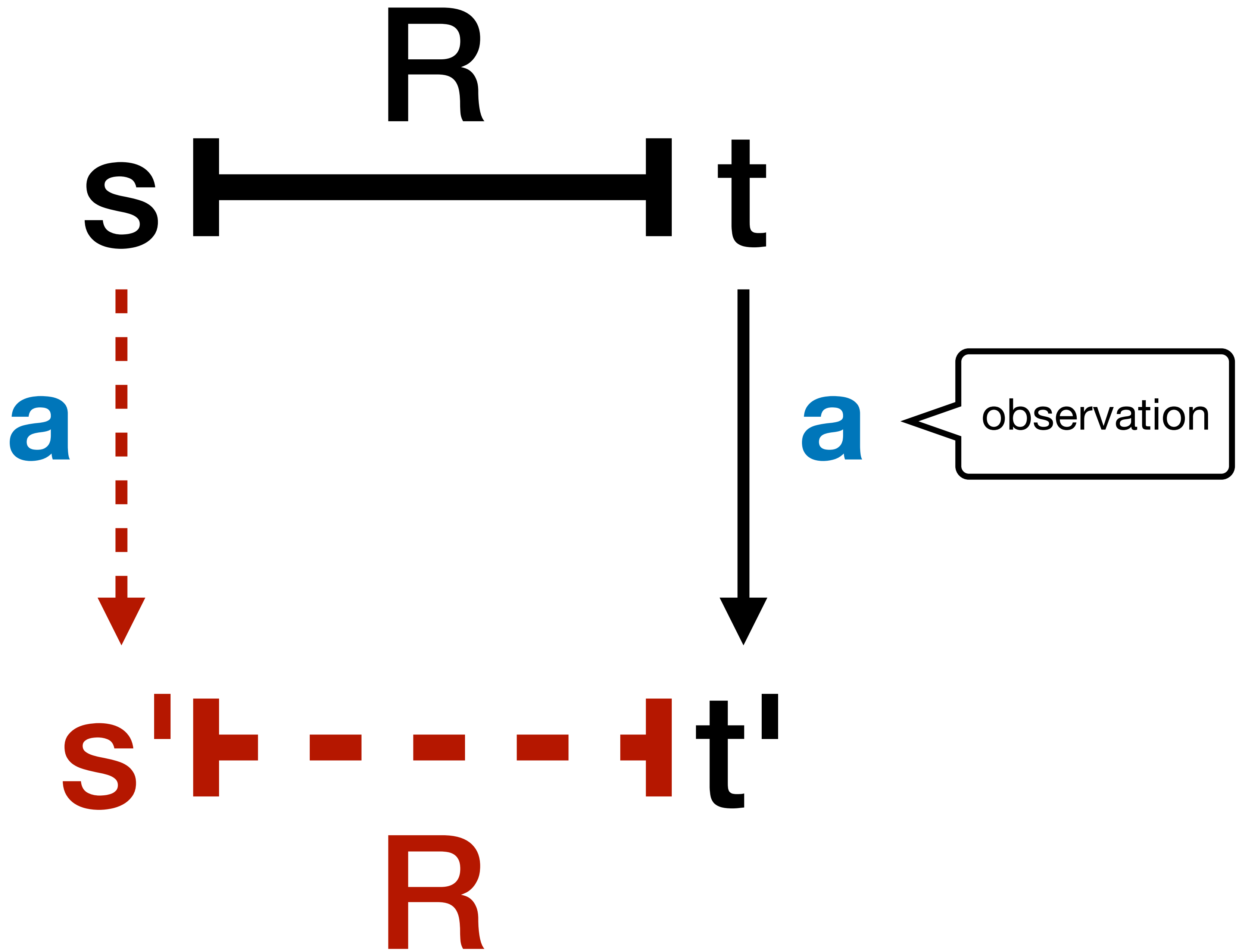
$$\sim \; = \bigcup \{ R \mid R \text{ is a strong bisimulation} \}$$

The intuition...



The intuition...

(and symmetrically)



Bisimilarity (Properties)

Theorem

Let P and Q be CCS processes such that $P \sim Q$. Then

- $\alpha.P \sim \alpha.Q$, for each $\alpha \in Act$
- $P+R \sim Q+R$ and $R+P \sim R+Q$, for each CCS process R
- $P|R \sim Q|R$ and $R|P \sim R|Q$, for each CCS process R
- $P[f] \sim Q[f]$, for each relabelling function f
- $P\backslash L \sim Q\backslash L$, for each set of labels $L \subseteq \mathbb{A}$

Theorem

For any P, Q, and R CCS processes, the following hold

- $P+Q \sim Q+P$
- $P+0 \sim P$
- $(P+Q)+R \sim P+(Q+R)$
- $P|Q \sim Q|P$
- $P|0 \sim P$
- $(P|Q)|R \sim P|(Q|R)$

Buffer of capacity n
(CCS implementation!)

Buffer of capacity n

A buffer of capacity n≥1, should satisfy the following:

- if *full*, it should not have any **input** capability;
- if *empty*, it should not have any **output** capability;
- otherwise, it should be able of **inputting** or **outputting**;

$$B_{i,n} \quad (\text{for } 0 \le i \le n)$$

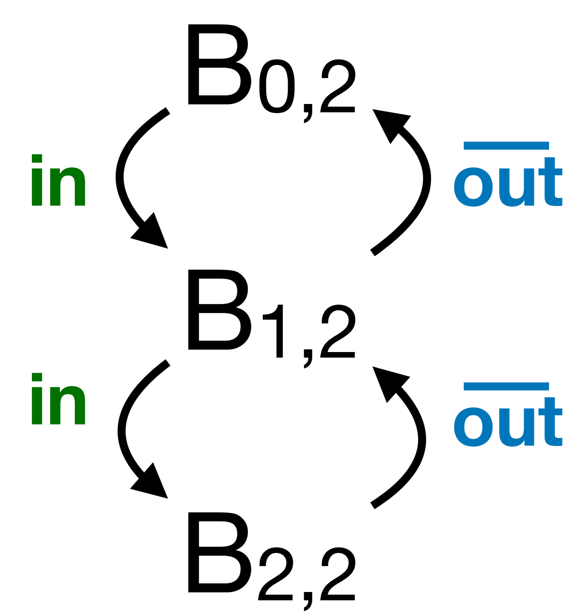
used space

total capacity

$$B_{0,n} \stackrel{\text{def}}{=} \mathbf{in}.B_{1,n}$$

$$B_{i,n} \stackrel{\text{def}}{=} \mathbf{in}.B_{i+1,n} + \overline{\mathbf{out}}.B_{i-1,n} \quad (\text{for } 0 < i < n)$$

$$B_{n,n} \stackrel{\text{def}}{=} \overline{\mathbf{out}}.B_{n-1,n}$$



Buffer (continued)

simpler and **modular**!

Theorem

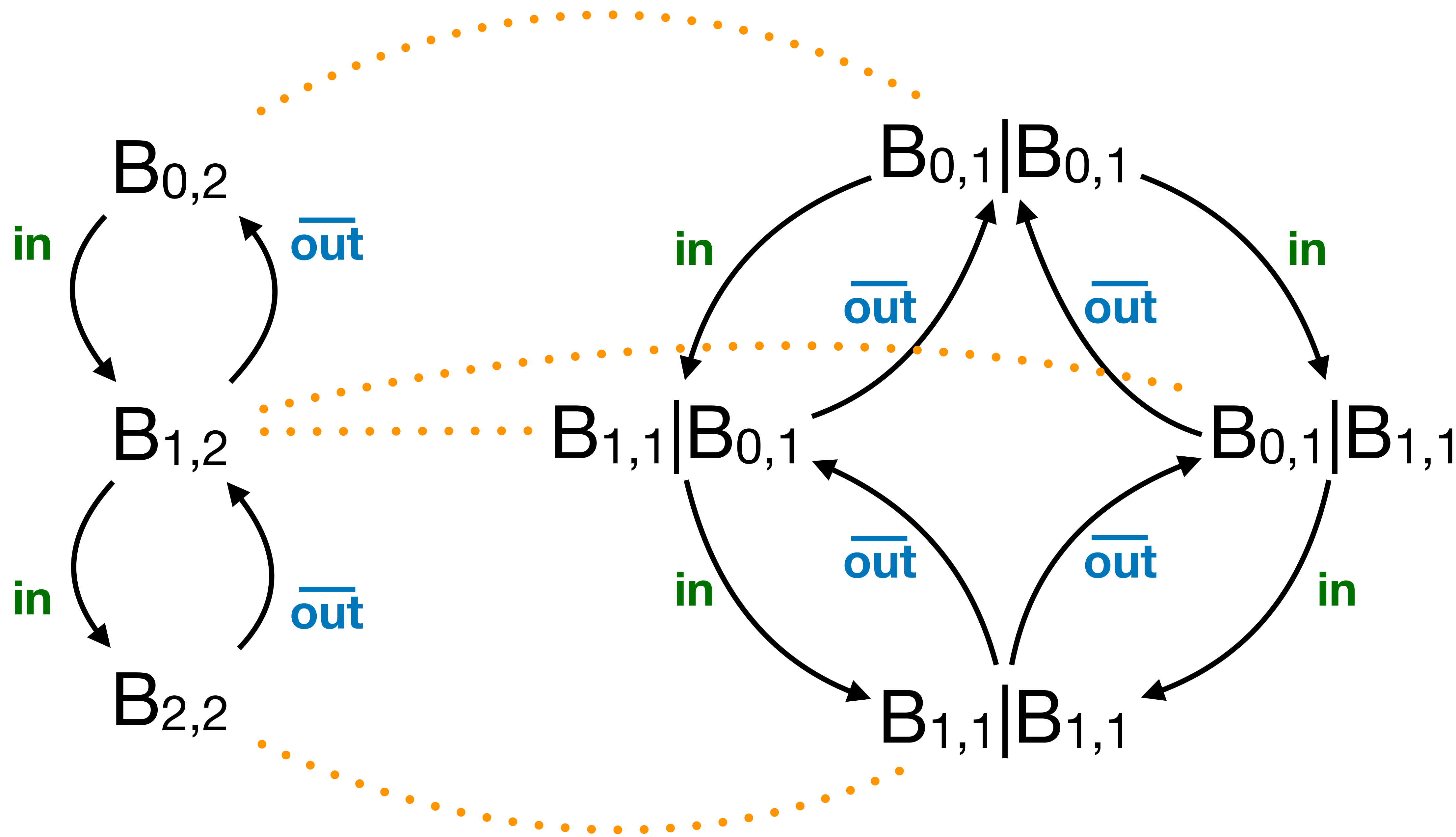
For all integers $n \geq 1$, $B_{0,n} \sim \underbrace{B_{0,1} \mid B_{0,1} \mid \ldots \mid B_{0,1}}_{\textbf{n-times}}$.

proof

Construct the following binary relation, where $i_1, \ldots, i_n \in \{0,1\}$,

$$R = \{ (B_{i,n}, B_{i_1,1} \mid \ldots \mid B_{i_n,1}) \mid i_1 + \ldots + i_n = i \}.$$

- $(B_{0,n} , B_{0,1} \mid B_{0,1} \mid \ldots \mid B_{0,1}) \in R$;
- R is a strong bisimulation .



$B_{0,2}$
$B_{1,2}$
$B_{2,2}$
in
$\overline{\text{out}}$
in
$\overline{\text{out}}$
$B_{0,1}|B_{0,1}$
$B_{1,1}|B_{0,1}$
$B_{0,1}|B_{1,1}$
$B_{1,1}|B_{1,1}$
in
in
$\overline{\text{out}}$
$\overline{\text{out}}$
in
in
$\overline{\text{out}}$
$\overline{\text{out}}$

Summary of properties

Properties of ~

- is an equivalence relation
- is the largest strong bisimulation
- is a congruence
- enough to prove some natural equivalences, like $P|0 \sim P$, $P|Q \sim Q|P$, ...

should we look any further?

Internal actions...

Implementation

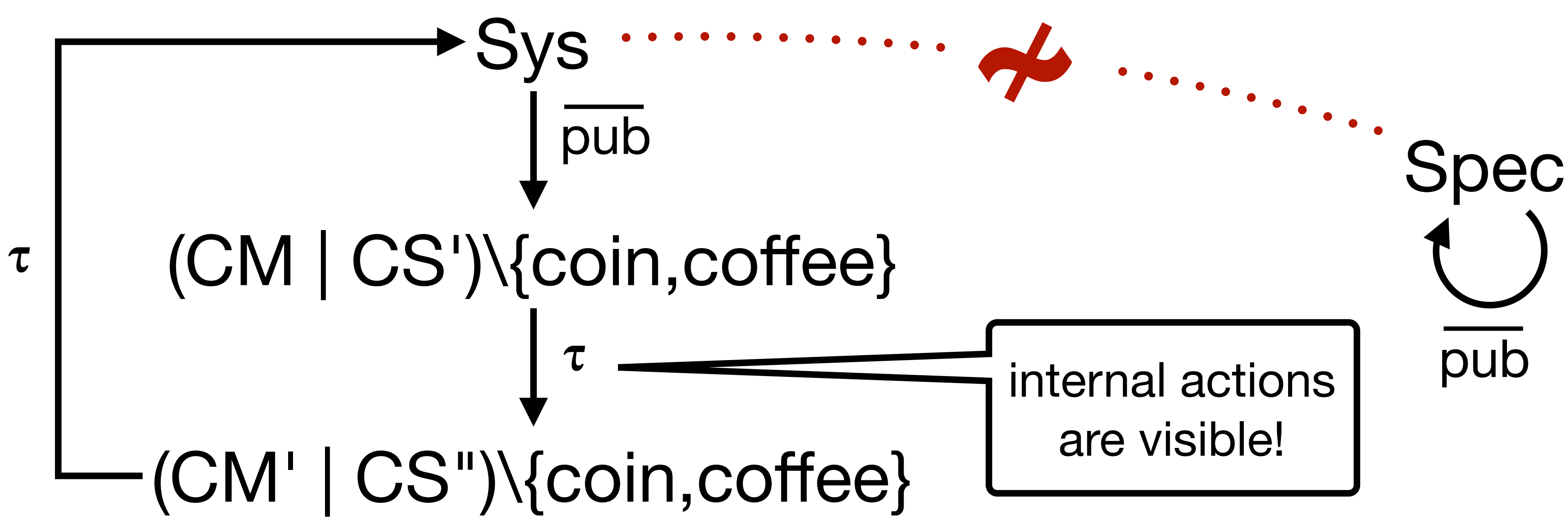
$$CS \stackrel{\text{def}}{=} \overline{pub}.\overline{coin}.coffee.CS$$

$$CM \stackrel{\text{def}}{=} coin.\overline{coffee}.CM$$

$$Sys \stackrel{\text{def}}{=} (CM \mid CS)\backslash\{coin,coffee\}$$

Specification

$$Spec \stackrel{\text{def}}{=} \overline{pub}.Spec$$



Weak Bisimulation

a way to abstract
from internal actions

Weak transition relation

Let (Proc, Act, $\{\xrightarrow{\alpha} \mid \alpha \in \text{Act}\}$) be an LTS such that $\tau \in$Act.

Definition (Weak Transition)

$$\overset{\alpha}{\Longrightarrow} = \begin{cases} (\xrightarrow{\tau})^* \circ \xrightarrow{\alpha} \circ (\xrightarrow{\tau})^* & \text{if } \alpha \neq \tau \\ (\xrightarrow{\tau})^* & \text{if } \alpha = \tau \end{cases}$$

- if $\alpha \neq \tau$, then $s \overset{\alpha}{\Longrightarrow} t$ means that from s we can get to t by doing zero of more τ actions, followed by an action α, followed by zero or more τ actions.
- if $\alpha = \tau$, then $s \overset{\alpha}{\Longrightarrow} t$ means that from s we can get to t by doing zero or more τ actions.

Weak Bisimilarity

Let (Proc, Act, $\{\xrightarrow{\alpha} \mid \alpha \in \text{Act}\}$) be an LTS such that $\tau \in$ Act.

Definition (Weak Bisimulation)

A binary relation R⊆Proc×Proc is a *weak bisimulation* iff whenever s R t, for each $\alpha \in$ Act (including τ)

- if $s \xrightarrow{\alpha} s'$, then $t \overset{\alpha}{\Longrightarrow} t'$, for some t' such that s'R t'
- if $t \xrightarrow{\alpha} t'$, then $s \overset{\alpha}{\Longrightarrow} s'$, for some s' such that s'R t'

Definition (Weak Bisimilarity)

Two states s,t∈Proc are *weakly bisimilar* (s ≈ t) iff there exists a weak bisimulation R such that s R t.

$$\approx \; = \bigcup \{R \mid R \text{ is a weak bisimulation}\}$$

The Game Characterisation

We define a *weak bisimulation game* in the same way we did in the case of strong bisimulation, except that

defender now answers using $\stackrel{\alpha}{\Longrightarrow}$-moves

(**attacker** still uses only $\xrightarrow{\alpha}$-moves)

Theorem

- The states s and t are weakly bisimilar iff the **defender** has a *universal* winning strategy starting from the configuration (s,t).
- The states s and t are *not* weakly bisimilar iff the **attacker** has a *universal* winning strategy starting from the configuration (s,t).

Internal actions...

Implementation

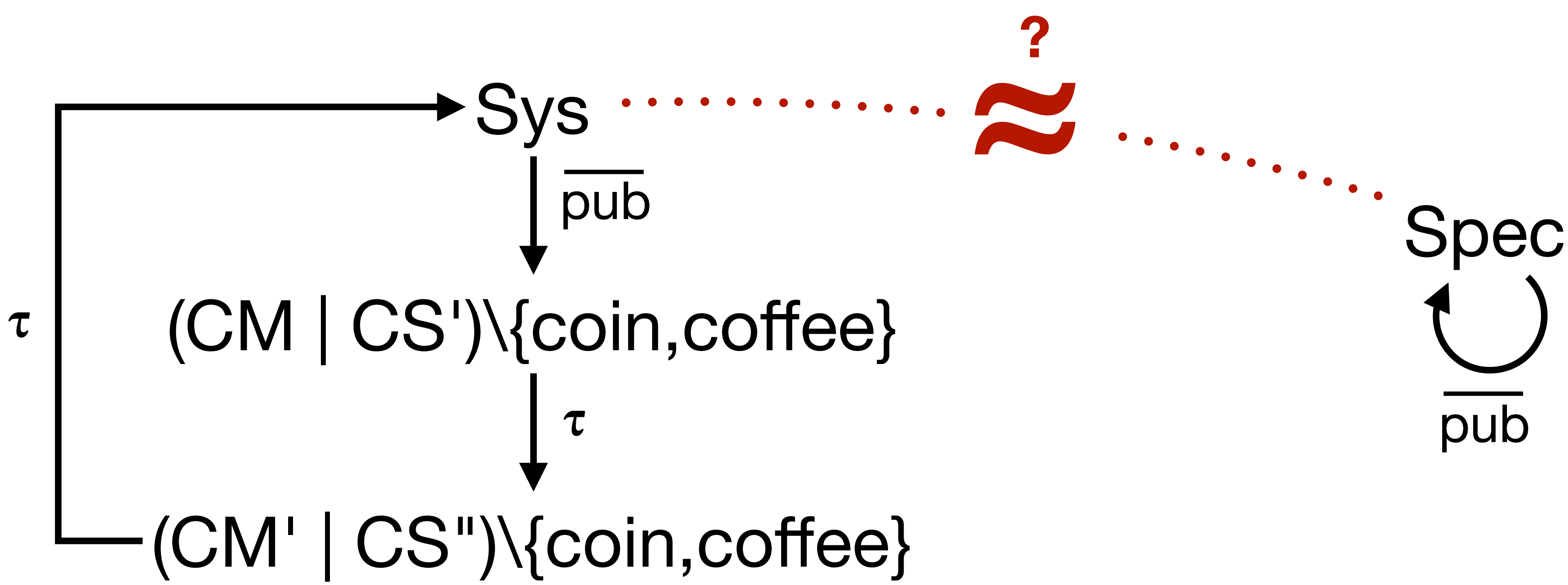
$$CS \stackrel{\text{def}}{=} \overline{\text{pub}}.\overline{\text{coin}}.\text{coffee}.CS$$

$$CM \stackrel{\text{def}}{=} \text{coin}.\overline{\text{coffee}}.CM$$

$$Sys \stackrel{\text{def}}{=} (CM \mid CS)\backslash\{\text{coin},\text{coffee}\}$$

Specification

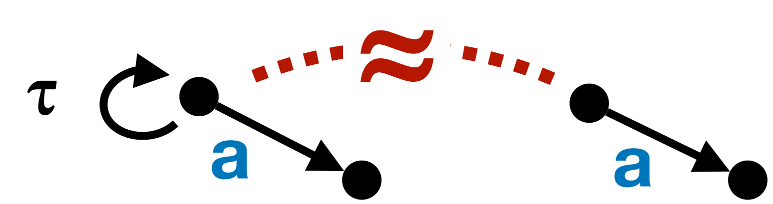
$$Spec \stackrel{\text{def}}{=} \overline{\text{pub}}.Spec$$



Properties of Weak Bisimilarity

Properties of ≈

- is an equivalence relation
- is the largest weak bisimulation
- validates lots of natural laws, e.g,
  - $a.\tau.P \approx a.P$; $P + \tau.P \approx \tau.P$;
  - $a.(P+ \tau.Q) \approx a.(P+ \tau.Q) + a.Q$
  - $P+Q \approx Q+P$; $P|Q \approx Q|P$; $P+0 \approx P$; etc...
- strong bisimilarity implies weak bisimilarity ($\sim \subseteq \approx$)
- abstract from τ-loops



Is it a Congruence?

Theorem

Let P and Q be CCS processes such that $P \approx Q$. Then

- $\alpha.P \approx \alpha.Q$, for each $\alpha \in Act$
- $P|R \approx Q|R$ and $R|P \approx R|Q$, for each CCS process R
- $P[f] \approx Q[f]$, for each relabelling function f
- $P\backslash L \approx Q\backslash L$, for each set of labels $L \subseteq \mathbb{A}$

what about nondeterministic choice?

$\tau.a.0 \approx a.0$ but $\tau.a.0 + b.0 \not\approx a.0 + b.0$

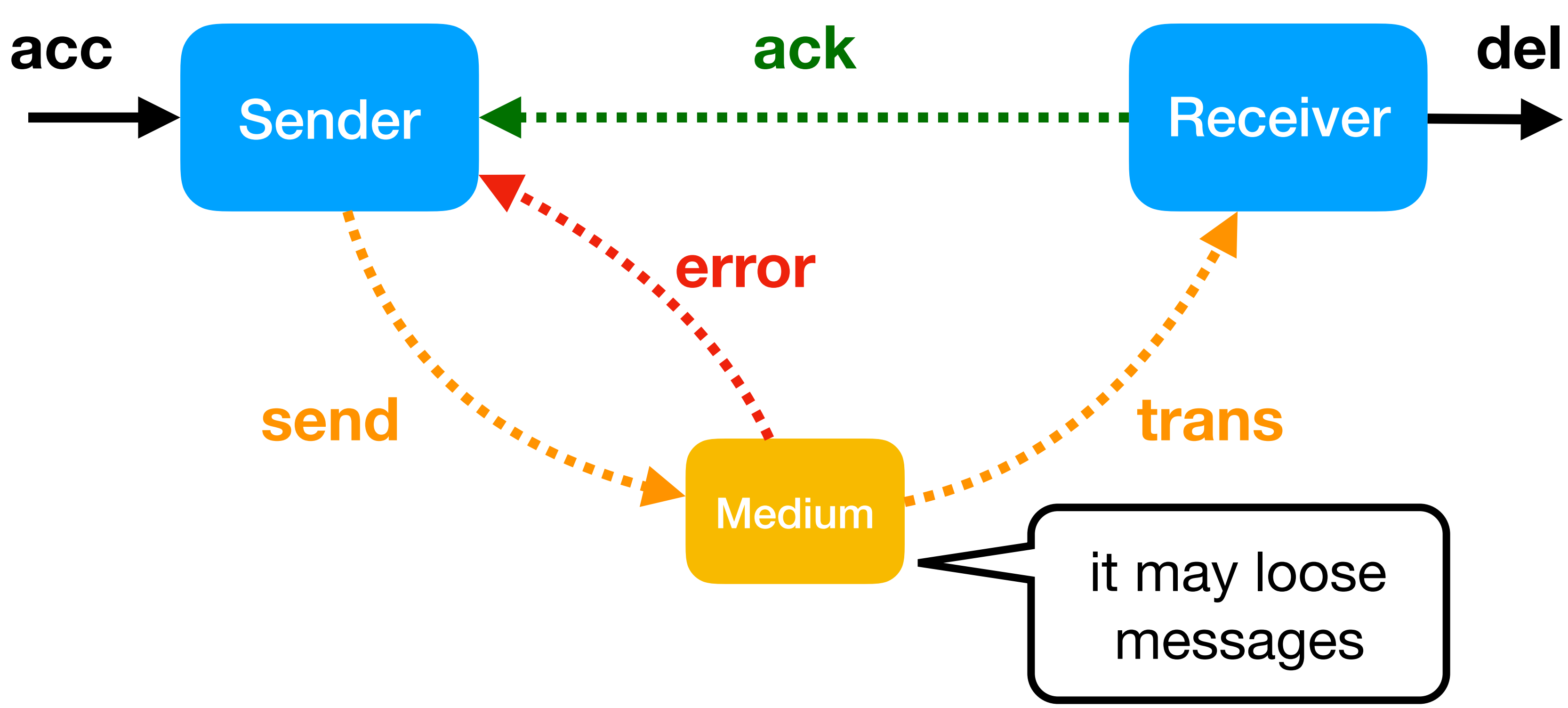
Conclusion

Weak bisimilarity is not a congruence for CCS

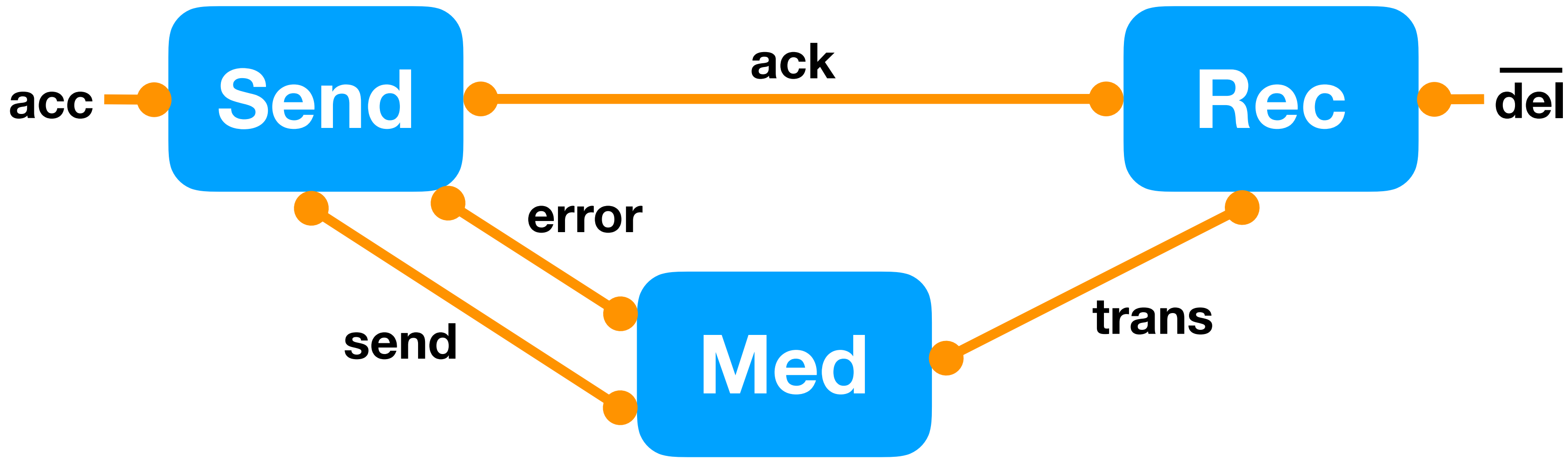
Case Study
(Communication Protocol)

Communication Protocol

A communication protocol is a discipline for transmission of messages from a source to destination. Sometimes it is designed to ensure reliable transmission under possible adverse conditions, such as loss messages caused by the transmission medium.



CCS Implementation



$$\text{Send} \stackrel{\text{def}}{=} \text{acc.Sending}$$
$$\text{Sending} \stackrel{\text{def}}{=} \overline{\text{send}}.\text{Wait}$$
$$\text{Wait} \stackrel{\text{def}}{=} \text{ack.Send} + \text{error.Sending}$$

$$\text{Rec} \stackrel{\text{def}}{=} \text{trans.Del}$$
$$\text{Del} \stackrel{\text{def}}{=} \overline{\text{del}}.\text{Ack}$$
$$\text{Ack} \stackrel{\text{def}}{=} \overline{\text{ack}}.\text{Rec}$$

$$\text{Med} \stackrel{\text{def}}{=} \text{send.Med'}$$
$$\text{Med'} \stackrel{\text{def}}{=} \tau.\text{Err} + \overline{\text{trans}}.\text{Med}$$
$$\text{Err} \stackrel{\text{def}}{=} \overline{\text{error}}.\text{Med}$$

Specification Checking

$$\text{Impl} \stackrel{\text{def}}{=} (\text{Send} \mid \text{Med} \mid \text{Rec}) \setminus \{\text{send},\text{trans},\text{ack},\text{error}\}$$

$$\text{Spec} \stackrel{\text{def}}{=} \text{acc}.\overline{\text{del}}.\text{Spec}$$

Question

$$\textbf{Impl} \stackrel{?}{\approx} \textbf{Spec}$$

- Draw the LTS of Impl and Spec and prove ≈ (by hand)
- Use *Concurrency WorkBench Aalborg Edition* (CAAL)

Concurrency WorkBench Aalborg Edition (CAAL)

http://caal.cs.aau.dk