Nominal Process Calculi and Modal Logics

Johannes Borgström
Uppsala University

Based on joint work since 2015 with
Ramūnas Gutkovas
Lars-Henrik Eriksson
Joachim Parrow
Tjark Weber

Introduction to Nominal Process Calculi

CCS with restriction

Nominal Process Calculi

- Process calculus: modelling language for systems of communicating processes.

- Three main traditions:
  - CSP (Hoare 1978)
  - CCS (Milner ~1980)
  - ACP (1982) process algebra

What is nominal process algebra?

Calculus of Communicating Systems

- Binary synchronization
  - Action (input) and coaction (output)

0	Nil
$a.P$	Input
$\overline{a}.P$	Output
$P + Q$	Choice
$P \mid Q$	Parallel
$(\nu a)P$	Restriction

Example 1a

Beverage machine $\mathrm{M}(tea,\ coffee,\ coin)$

$$\begin{aligned}\mathrm{M}(tea,\ coffee,\ coin) := {} & coin.\overline{tea}.\mathrm{M}(tea,\ coffee,\ coin) + \\ & coin.coin.\overline{coffee}.\mathrm{M}(tea,\ coffee,\ coin)\end{aligned}$$

Example 1b

Dining philosophers $\mathrm{Philo}(\mathit{left}, \mathit{right}, \mathit{eat})$

$$\mathrm{Philo}(\mathit{left}, \mathit{right}, \mathit{eat}) := \mathit{left}.\mathit{right}.\overline{\mathit{eat}}.\overline{\mathit{left}}.\overline{\mathit{right}}.\mathrm{Philo}(\ldots)$$

$$(\nu\ \mathit{cs1})(\nu\ \mathit{cs2})(\nu\ \mathit{cs3})(\mathrm{Philo}(\mathit{cs1}, \mathit{cs2}, \mathit{eat1}) | \mathrm{Philo}(\mathit{cs2}, \mathit{cs3}, \mathit{eat2}) | \\ \mathrm{Philo}(\mathit{cs3}, \mathit{cs1}, \mathit{eat3}) \mid \overline{\mathit{cs1}} \mid \overline{\mathit{cs2}} \mid \overline{\mathit{cs3}})$$

We write a for $a.0$, and $\bar{a}$ for $\bar{a}.0$

Labelled Semantics

$$\textsc{In}\ \frac{}{a.P \xrightarrow{a} P} \qquad \textsc{Out}\ \frac{}{\overline{a}.P \xrightarrow{\overline{a}} P}$$

$$\textsc{Sum-L}\ \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \textsc{Par-L}\ \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\textsc{Com-L}\ \frac{P \xrightarrow{a} P' \qquad Q \xrightarrow{\overline{a}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\textsc{Scope}\ \frac{P \xrightarrow{\alpha} P'}{(\nu b)P \xrightarrow{\alpha} (\nu b)P'}\ b\#\alpha$$

Example 2

Dining philosophers $\mathrm{Philo}(\mathit{left},\mathit{right},\mathit{eat})$

$$\mathrm{Philo}(\mathit{left},\mathit{right},\mathit{eat}) := \mathit{left}.\mathit{right}.\overline{\mathit{eat}}.\overline{\mathit{left}}.\overline{\mathit{right}}.\mathrm{Philo}(\ldots)$$

$$(\nu\ \mathit{cs1})(\nu\ \mathit{cs2})(\nu\ \mathit{cs3})(\mathrm{Philo}(\mathit{cs1},\mathit{cs2},\mathit{eat1})|\mathrm{Philo}(\mathit{cs2},\mathit{cs3},\mathit{eat2})| \\ \mathrm{Philo}(\mathit{cs3},\mathit{cs1},\mathit{eat3}) \mid \overline{\mathit{cs1}} \mid \overline{\mathit{cs2}} \mid \overline{\mathit{cs3}})$$

$$\mathrm{Philo2}(\mathit{left},\mathit{right},\mathit{eat}) := \mathit{left}.(\mathit{right}.\overline{\mathit{eat}}.(\overline{\mathit{left}}\ |\overline{\mathit{right}}|\mathrm{Philo}(\ldots) \\ + \overline{\mathrm{left}}.\mathrm{Philo}(\ldots))$$

Observational Equivalence

- When can an external observer distinguish between two systems?
- Idea: when either of them can perform an action
  - that the other one cannot perform; or
  - that leads the other system into a state that can be distinguished from the new state of the first system.
- An inductive definition!
- Its negation is coinductive: bisimulation (Park 1981)

Bisimulation

DEFINITION (Strong Bisimulation)

A symmetric relation R on processes satisfying:

if $R(P,Q)$ then

if $P \xrightarrow{\alpha} P'$ then

$\exists Q' . Q \xrightarrow{\alpha} Q'$ and $R(P',Q')$

Simulation

$P \dot{\sim} Q$ if $R(P,Q)$ for some bisimulation R

Examples 3

- Check that $\mathrm{M}(tea,\ coffee,\ coin)$ and $\mathrm{M2}(tea,\ coffee,\ coin)$ below are not bisimilar.

$$\mathrm{M2}(tea,\ coffee,\ coin) := coin.(\overline{tea}.\mathrm{M2}(tea,\ coffee,\ coin) + coin.\overline{coffee}.\mathrm{M2}(tea,\ coffee,\ coin))$$

- Check that the system below is weakly bisimilar to $\mathrm{Spec}(eat1, eat2, eat3) :=$
$eat1.\mathrm{Spec}(...) + eat2.\mathrm{Spec}(...) + eat3.\mathrm{Spec}(...)$

$$(\nu\ cs1)(\nu\ cs2)(\nu\ cs3)(\mathrm{Philo2}(cs1, cs2, eat1) \mid \mathrm{Philo2}(cs2, cs3, eat2) \mid \mathrm{Philo2}(cs3, cs1, eat3) \mid \overline{cs1} \mid \overline{cs2} \mid \overline{cs3})$$

Com-po-si-tio-na-li-ty

- Bisimilarity is an equivalence relation, and a congruence for all operators
  - Allows to substitute bisimilar processes in any context: compositional reasoning

- Structural congruence $\equiv$
  - The smallest congruence relation on processes containing commutative monoid laws for $|$ (parallel) and $+$ (choice) with 0 as unit.
  - $\equiv$ is a bisimulation

The π-calculus

Scope extension, scope extrusion, and residuals

Milner, Parrow, Walker: A calculus of mobile processes.
Information and Computation 100(1) 1992.

The π-calculus

- An extension of CCS with name communication
  - Value-passing can be encoded in CCS using summation
  - General name-passing needs infinite summation: not finitely supported!
- Turing-complete, can easily encode the untyped lambda-calculus
  - Current research on behavioural (session) types

Syntax of π

0	Nil
$a(x).P$	Input
$\overline{a}\ b.P$	Output
$P + Q$	Choice
$P \mid Q$	Parallel
$(\nu a)P$	Restriction

Examples 1

Truth values (at location *l*)

$$\mathrm{True}(l) := l(t,f).(\overline{t} \mid \mathrm{True}(l))$$

$$\mathrm{False}(l) := l(t,f).(\overline{f} \mid \mathrm{False}(l))$$

Let's do lists!

$$\mathrm{Nil}(l) := l(n,c).(\overline{n} \mid \mathrm{Nil}(l))$$

$$\mathrm{Cons}(l,value,tail) := l(n,c).(\overline{c}\ value,tail \mid \mathrm{Cons}(...))$$

What does $(\nu b)\overline{a}\, b.P$ do?

We write $\overline{a}$ for $\overline{a}\ a.0$ and $\overline{a}\ b,c$ for $\overline{a}\ b.\overline{a}\ c$ and $a(b,c)$ for $a(b).a(c)$

Labelled Semantics

$$\textsc{In}\ \frac{}{a(x).P \xrightarrow{a\,b} P\{b/x\}}$$

$$\textsc{Out}\ \frac{}{\overline{a}\,b.P \xrightarrow{\overline{a}\,b} P}$$

But what about $(\nu b)\overline{a}\,b.P$?

$$\textsc{Sum-L}\ \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$$

$$\textsc{Par-L}\ \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\textsc{Com-L}\ \frac{P \xrightarrow{a\,b} P' \qquad Q \xrightarrow{\overline{a}\,b} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\textsc{Scope}\ \frac{P \xrightarrow{\alpha} P'}{(\nu b)P \xrightarrow{\alpha} (\nu b)P'}\ b\#\alpha$$

Structural congruence $\equiv$

- The smallest congruence relation containing
  - commutative monoid laws for $|$ (parallel) and $+$ (choice) with 0 as unit;
  - and the scope extension laws

$$
\begin{array}{rcl}
P \mid (\nu b)Q & \equiv & (\nu b)(P \mid Q) \text{ when } b\#P \\
P + (\nu b)Q & \equiv & (\nu b)(P + Q) \text{ when } b\#P \\
(\nu a)(\nu b)P & \equiv & (\nu b)(\nu a)P
\end{array}
$$

Reduction Semantics

$$\textsc{Red}\ \frac{}{(a(x).P + P') \mid (\overline{a}\, b.Q + Q') \rightarrow P\left\{{}^{b}/_{x}\right\} \mid Q}$$

$$\textsc{Struct}\ \frac{P \equiv P' \qquad P' \rightarrow Q \qquad Q \equiv Q'}{P \rightarrow Q'}$$

$$\textsc{Ctx-Par}\ \frac{P \rightarrow P'}{P \mid Q \rightarrow P' \mid Q} \qquad \textsc{Ctx-Res}\ \frac{P \rightarrow P'}{(\nu b)P \rightarrow P'}$$

Examples 2

`if true then P else Q`

$(\nu\ l)(\nu\ t)(\nu\ f)(\mathrm{True}(l) \mid l(t,f) \mid t.P \mid f.Q)$

`case l of Nil -> P | Cons(v,l') -> Q`

$(\nu\ n)(\nu\ c)(\ l(n,c) \mid n.P \mid c(v,l').Q)$

Set binders

$$(as, x) \approx_{set}^{R, fa, p} (bs, y) \stackrel{def}{=}$$

$$\begin{array}{ll} (i) & fa\ x - as = fa\ y - bs \\ (ii) & fa\ x - as \#^* p \\ (iii) & (p \bullet x)\ R\ y \\ (iv) & p \bullet as = bs \end{array}$$

Urban, Kaliszyk: General Bindings and Alpha-Equivalence in Nominal Isabelle. ESOP 2011

NTS Labelled Semantics

But what about $(\nu b)\overline{a}\,b.P$?

$$\textsc{In}\ \frac{}{a(x).P \to \langle\emptyset\rangle(a\,b,\ P\left\{{}^{b}/_{x}\right\})} \qquad \textsc{Out}\ \frac{}{\overline{a}\,b.P \to \langle\emptyset\rangle(\overline{a}\,b,\ P)}$$

$$\textsc{Sum-L}\ \frac{P \to S}{P + Q \to S} \qquad \textsc{Par-L}\ \frac{P \to \langle C\rangle(\alpha,\ P')}{P \mid Q \to \langle C\rangle(\alpha,\ P' \mid Q)}\ C\#Q$$

$$\textsc{Com-L}\ \frac{P \to \langle\emptyset\rangle(a\,b,\ P') \qquad Q \to \langle\emptyset\rangle(\overline{a}\,b,\ Q')}{P \mid Q \to \langle\emptyset\rangle(\tau,\ P' \mid Q')}$$

$$\textsc{Scope}\ \frac{P \to \langle C\rangle(\alpha,\ P')}{(\nu b)P \to \langle C\rangle(\alpha,\ (\nu b)P')}\ b\#\alpha$$

NTS Labelled Semantics

$$\textsc{Com-L}\ \frac{P \to \langle\emptyset\rangle(a\,b,\ P') \qquad Q \to \langle\emptyset\rangle(\overline{a}\,b,\ Q')}{P \mid Q \to \langle\emptyset\rangle(\tau,\ P' \mid Q')}$$

$$\textsc{Scope}\ \frac{P \to \langle C\rangle(\alpha,\ P')}{(\nu b)P \to \langle C\rangle(\alpha,\ (\nu b)P')}\ b\#\alpha$$

$$\textsc{Close-L}\ \frac{P \to \langle\emptyset\rangle(a\,b,\ P') \qquad Q \to \langle\{b\}\rangle(\overline{a}\,\underline{b},\ Q')}{P \mid Q \to \langle\emptyset\rangle(\tau,\ (\nu b)(P' \mid Q'))}\ b\#P$$

$$\textsc{Open}\ \frac{P \to \langle\emptyset\rangle(\overline{a}\,b,\ P')}{(\nu b)P \to \langle\{b\}\rangle(\overline{a}\,\underline{b},\ P')}\ b\#a$$

Based on Gabbay: The π-Calculus in FM,
in "Thirty Five Years of Automating Mathematics", Kluwer 2004

Examples 3

`if true then a else b`

$$(\nu\ l)(\nu\ t)(\nu\ f)(\mathrm{True}(l) \mid l(t,f) \mid t.\overline{a} \mid f.\overline{b})$$

$$\mathrm{Connect}(c,P(l)) := (\nu\ l)\overline{c}\ l.P(l)$$

$$\mathrm{Connect}(c,(\overline{l}\ a)(l)) \mid c(b).b(x).\overline{x}$$

What are the transitions of $(\nu a)\overline{c}\ a \mid (\nu c)\overline{c}\ a$?

Bisimulation

DEFINITION (Strong Bisimulation)

A symmetric relation R on processes satisfying:

if $R(P,Q)$ then

$$\text{If } P \xrightarrow{\alpha} P' \text{ and } \boxed{\text{bn}(\alpha)\#Q} \text{ then}$$

$$\exists Q' . \, Q \xrightarrow{\alpha} Q' \text{ and } R(P',Q')$$

Simulation

$P \dot{\sim} Q$ if $R(P,Q)$ for some bisimulation R

Examples 4

- Check that $(\nu c)\overline{c}\ a$ is bisimilar to 0.
- Check that $(\nu \mathrm{a})\overline{c}\ a$ is bisimilar to $(\nu a)\overline{c}\ a \mid (\nu c)\overline{c}\ a$

Com-po-si-tio-na-li-ty

- Bisimilarity is an equivalence relation,
  and a congruence for all operators **except input**
  - Allows to substitute bisimilar processes in any **non-input** context: compositional reasoning

- Structural congruence $\equiv$ is a bisimulation

Nominal Transition Systems

Based on slides by Joachim Parrow, OPCT 2017
(I omit predicates for now.)

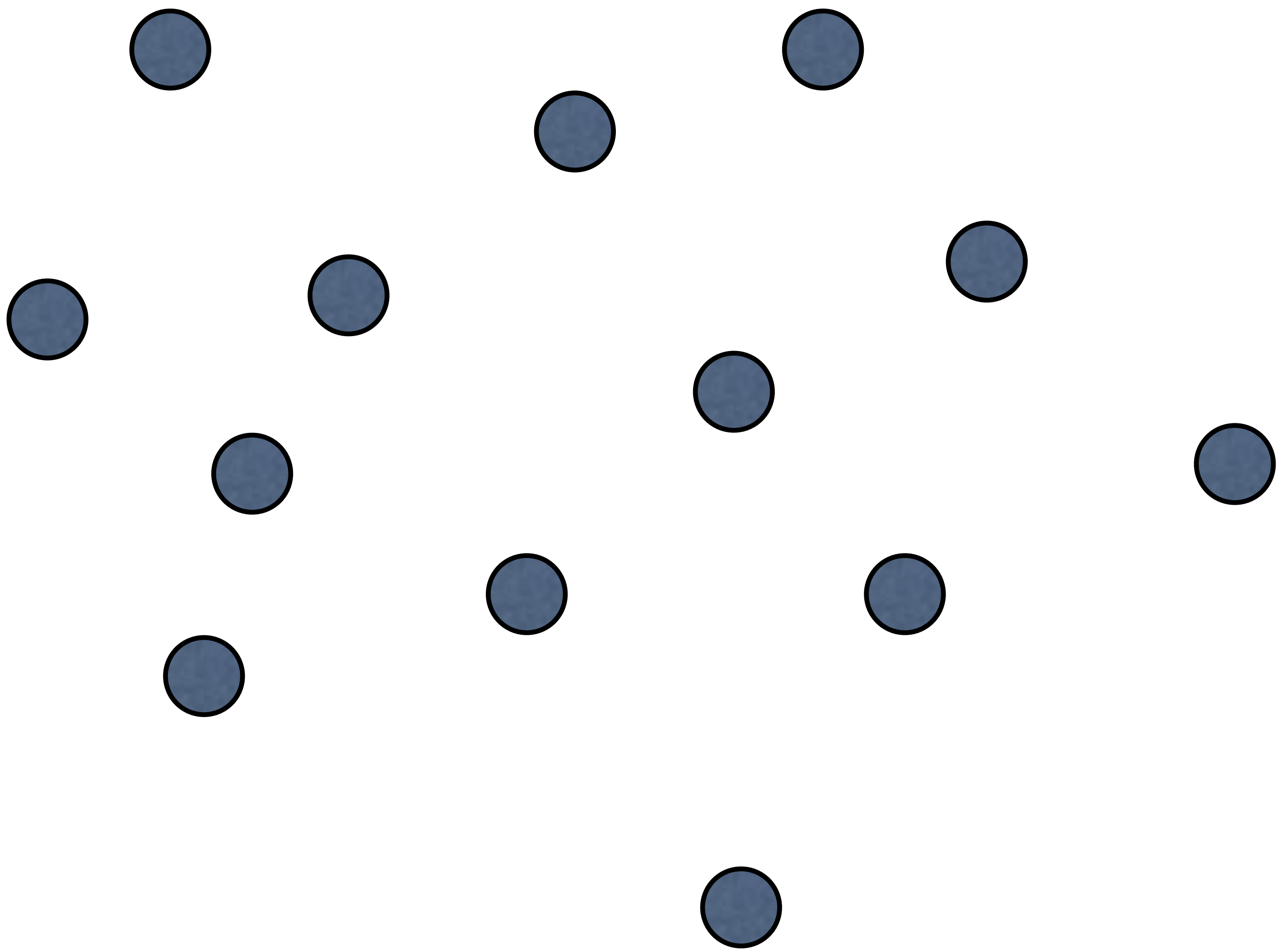
Nominal Transition Systems

What are NTS? Why?

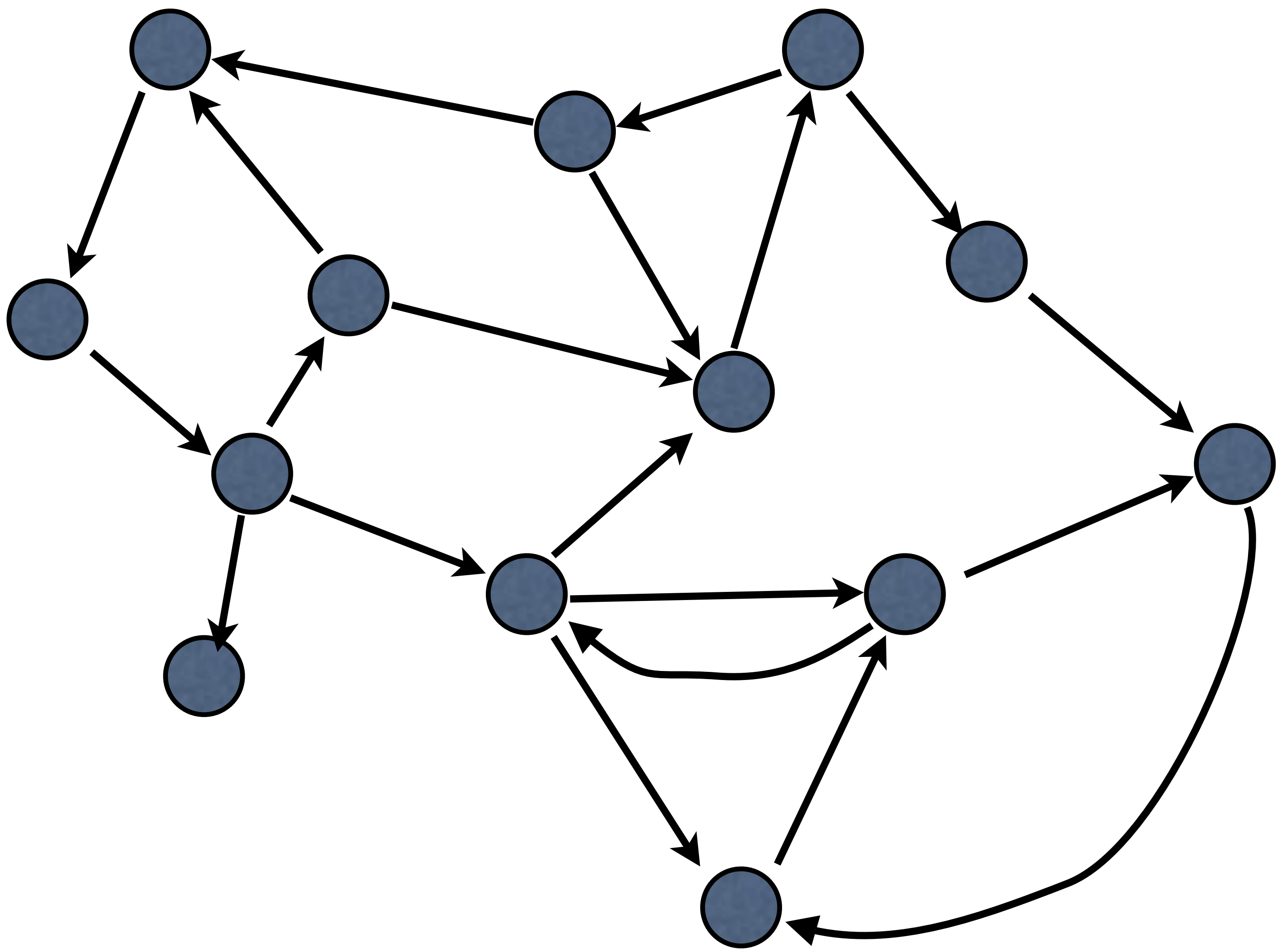
NTS are a **general framework** that fits almost all **advanced process algebras**,

by generalising standard transition systems to include **binders in actions**

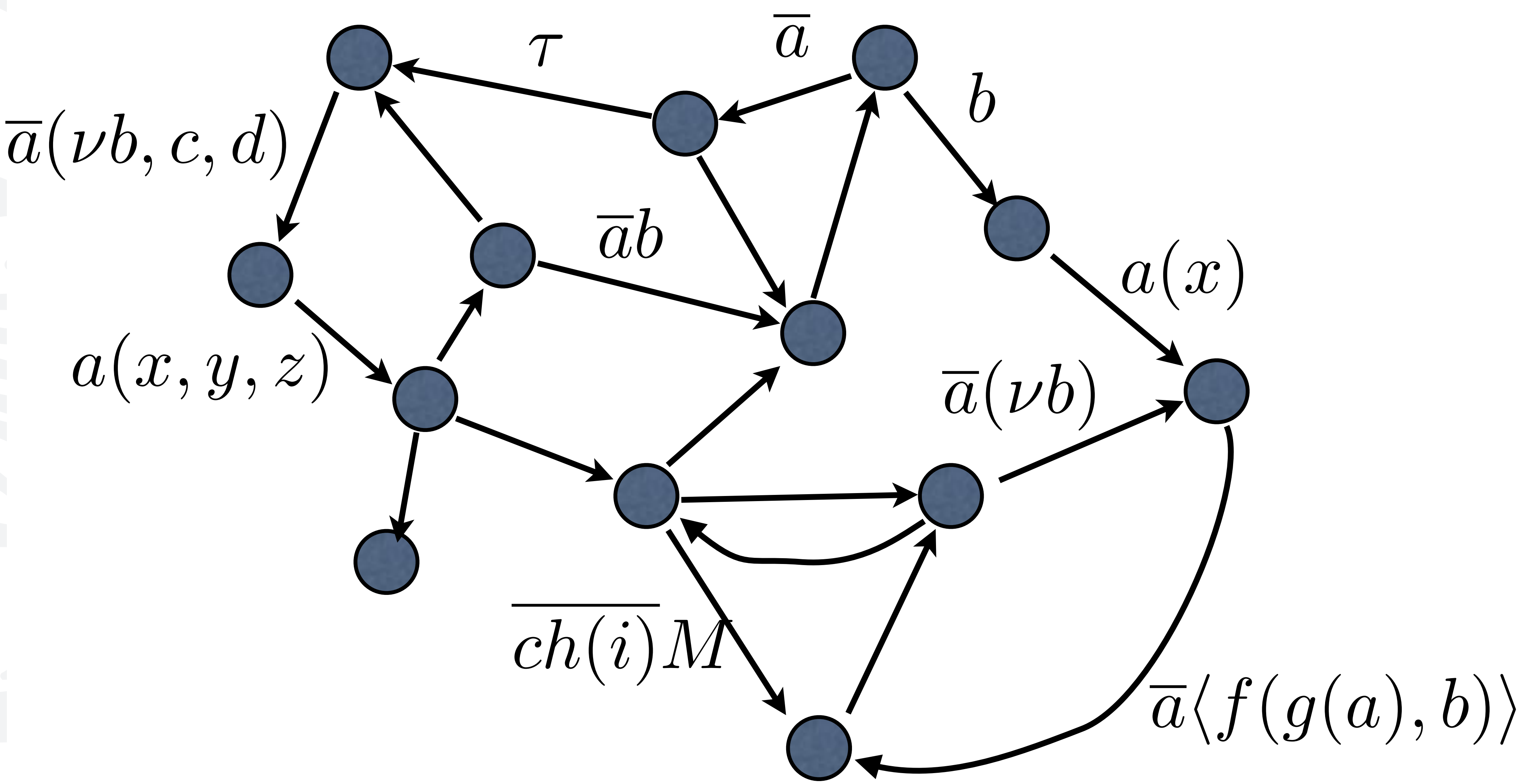
States



Transitions

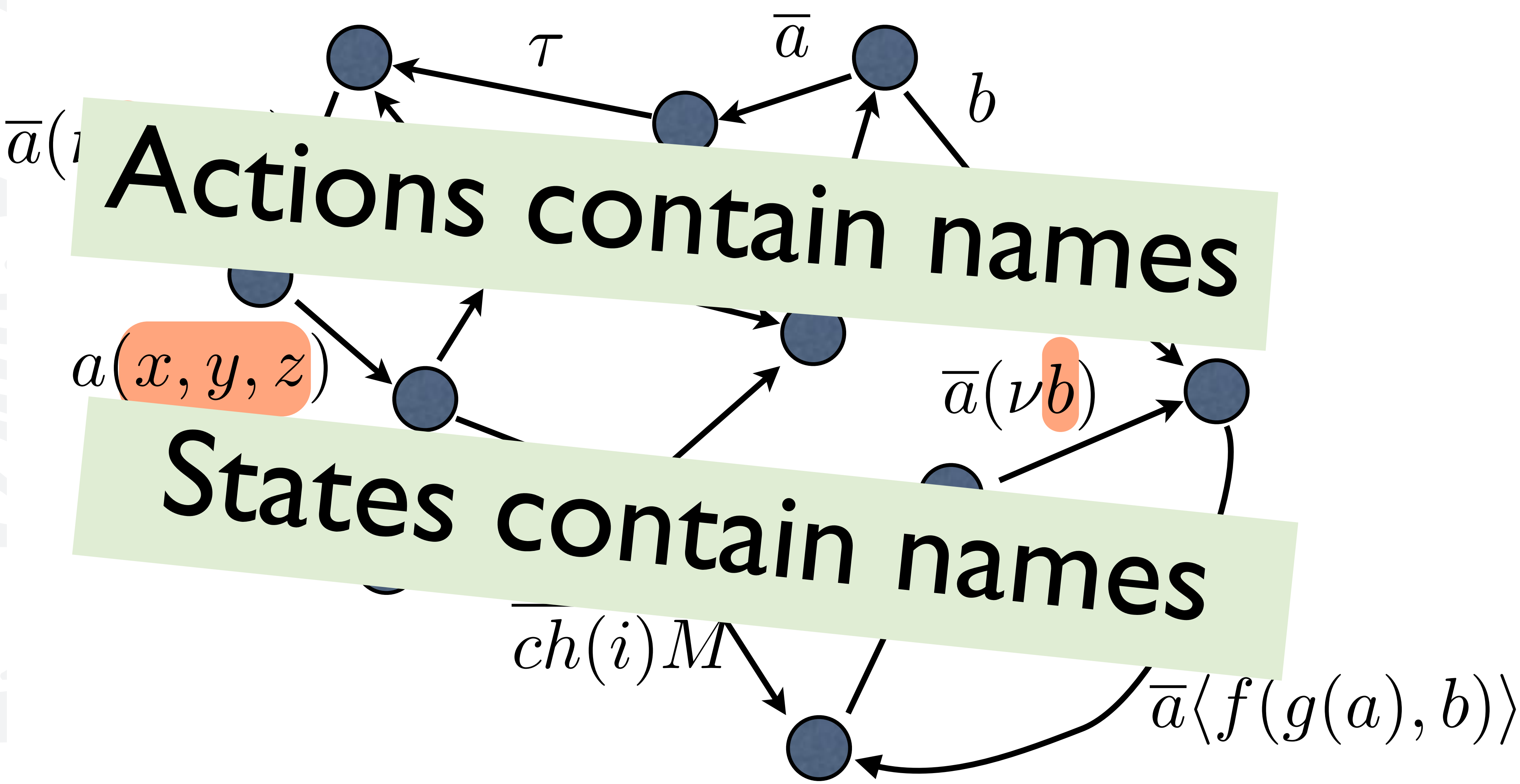


Actions



τ
$\overline{a}$
b
$\overline{a}(\nu b, c, d)$
$\overline{a}b$
$a(x)$
$a(x, y, z)$
$\overline{a}(\nu b)$
$\overline{ch(i)}M$
$\overline{a}\langle f(g(a), b)\rangle$

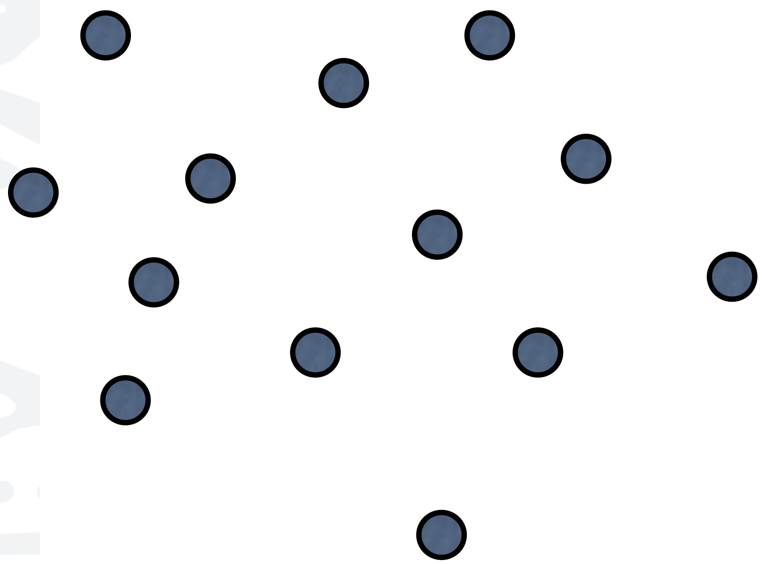
Binding names



Actions contain names

States contain names

States and actions



$\texttt{STATES}$: A nominal set $\quad P, \; Q$

$\tau \quad \overline{a} \quad b$

$\overline{a}b \quad a(x)$

$a(x, y, z) \quad \overline{a}(\nu b)$

$\overline{ch(i)}M \quad \overline{a}\langle f(g(a), b)\rangle$

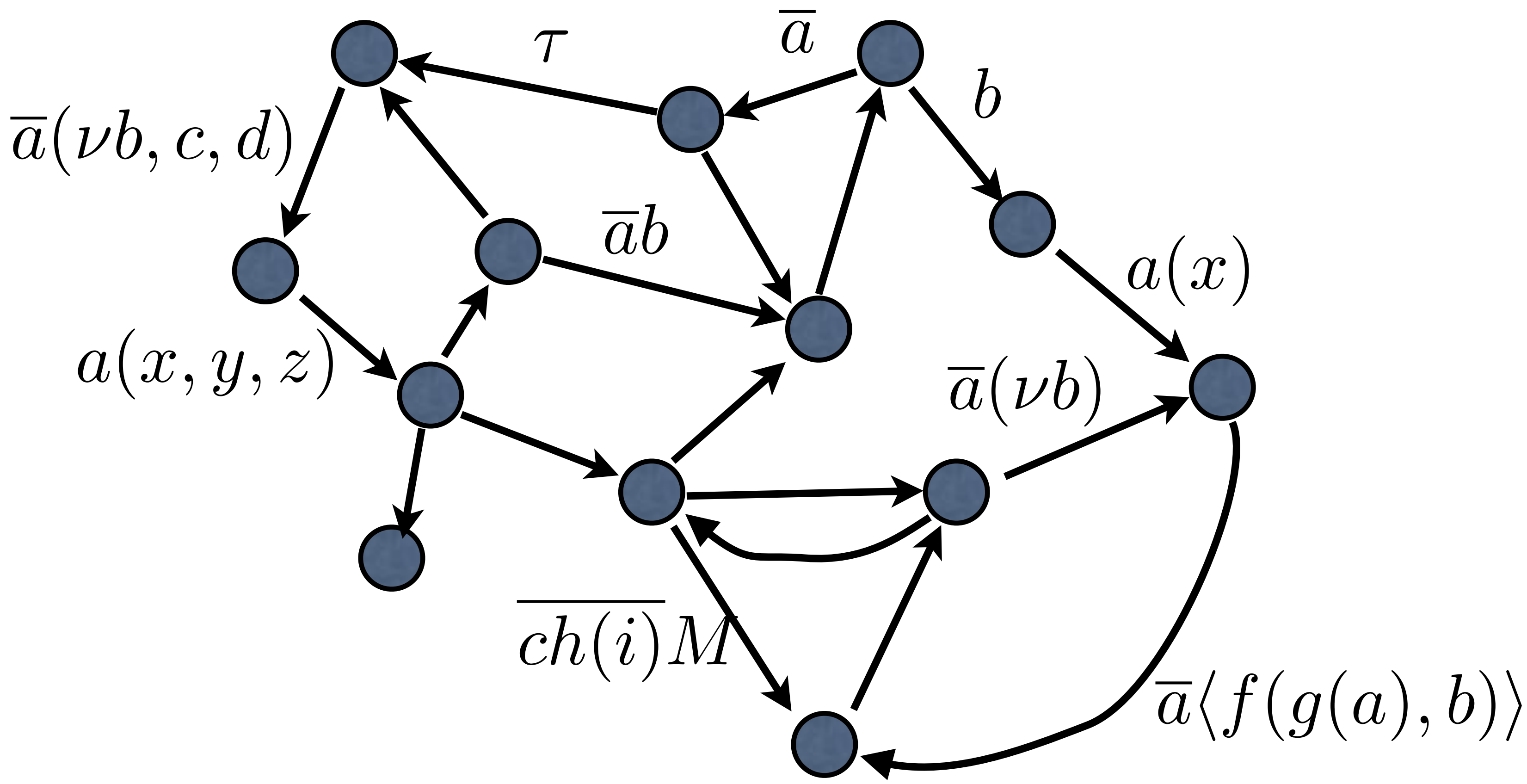
$\overline{a}(\nu b, c, d)$

$\texttt{ACT}$: A nominal set $\quad \alpha$

$$\mathrm{bn} : \mathrm{ACT} \to P_{\mathrm{fin}}(\mathcal{N}) \quad \text{equivariant}$$

$$\mathrm{bn}(\alpha) \subseteq \mathrm{supp}(\alpha)$$

Transitions



$\to \;\subseteq\; \textsc{states} \times [P_{\text{fin}}(\mathcal{N})](\textsc{act} \times \textsc{states})$ equivariant

$(P, <\tilde{b}> (\alpha, Q)) \in \to$ implies $\tilde{b} = \text{bn}(\alpha)$

We write $P \xrightarrow{\alpha} Q$ for $(P, \langle \text{bn}(\alpha) \rangle (\alpha, Q)) \in \to$

Bisimulation

DEFINITION (Strong Bisimulation)

A symmetric relation R on processes satisfying:

if $R(P,Q)$ then

$$\text{If } P \xrightarrow{\alpha} P' \text{ and } \mathrm{bn}(\alpha)\#Q \text{ then}$$

$$\exists Q'. \, Q \xrightarrow{\alpha} Q' \text{ and } R(P',Q')$$

Simulation

$$P \dot{\sim} Q \text{ if } R(P,Q) \text{ for some bisimulation } R$$

Summary

- Three process calculi: CCSish, pi, ~~fusion~~
- Reduction semantics
- Residual-based labelled semantics
- Bisimulation
- Generalization: Nominal Transition Systems (NTS)

- **Saturday: Psi-calculi, modal logic for NTSs**
  - **Weak bisimilarity, weak logic, effects**

The Ψ-calculus

Jesper Bengtson, Magnus Johansson, Joachim Parrow, Björn Victor, Johannes Åman Pohjola, et al.

From pi to psi

$(\nu z)(\overline{a}z) \mid a(x).\,[x=b]P$

Ordinary pi-calculus

arbitrary set of data

$(\nu z)(\overline{a}M) \mid a(x).\,[x=b]P$

Data structures can be sent

$(\nu z)(\overline{a}M) \mid a(\lambda\tilde{x})N.\,[x=b]P$

Pattern matching

$(\nu z)(\overline{K}M) \mid L(\lambda\tilde{x})N.\,[x=b]P$

Channels can be arbitrary structures

arbitrary logic

$(\nu z)(\overline{K}M) \mid L(\lambda\tilde{x})N.\,\text{if } \varphi \text{ then } P$

Tests can be arbitrary predicates

new construct

$(\nu z)(\overline{K}M).\,(\!|\Psi|\!) \mid L(\lambda\tilde{x})N.\,\text{if } \varphi \text{ then } P$

Facts about data

Cook a psi-calculus

Define terms $\mathbf{T}$ (data terms, channels) M, N

and conditions $\mathbf{C}$ (used in case stmt) φ

and assertions $\mathbf{A}$ (facts about data) Ψ

can be any nominal set (not syntactic)

Define term substitution, and operators:

$\dot{\leftrightarrow} : \mathbf{T} \times \mathbf{T} \to \mathbf{C}$ Channel equivalence

$\otimes : \mathbf{A} \times \mathbf{A} \to \mathbf{A}$ Composition

$\mathbf{1} : \mathbf{A}$ Unit assertion

$\vdash \subseteq \mathbf{A} \times \mathbf{C}$ Entailment

(practically anything)

Axioms for substitution

Assume all the $\tilde{a}$ distinct, all the $\tilde{b}$ distinct.

if $\tilde{a} \subseteq \mathrm{n}(X)$ and $b \in \mathrm{n}(\tilde{T})$ then $b \in \mathrm{n}(X[\tilde{a} := \tilde{T}])$

if $\tilde{b} \# X$, $\tilde{a}$ then $X[\tilde{a} := \tilde{T}] = ((\tilde{b}\ \tilde{a}) \cdot X)[\tilde{b} := \tilde{T}]$

Easy as pi!

$$\textsc{In}\ \frac{\Psi \vdash M \dot{\leftrightarrow} K}{\Psi \rhd \underline{M}(\lambda\tilde{y})N.P \xrightarrow{\underline{K}\,N[\tilde{y}:=\tilde{L}]} P[\tilde{y} := \tilde{L}]}$$

$$\textsc{Out}\ \frac{\Psi \vdash M \dot{\leftrightarrow} K}{\Psi \rhd \overline{M}\,N.P \xrightarrow{\overline{K}\,N} P}$$

$$\textsc{Case}\ \frac{\Psi \rhd P_i \xrightarrow{\alpha} P' \qquad \Psi \vdash \varphi_i}{\Psi \rhd \mathbf{case}\ \tilde{\varphi} : \tilde{P} \xrightarrow{\alpha} P'}$$

$$\textsc{Com}\ \frac{\Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \dot{\leftrightarrow} K \quad \Psi_Q \otimes \Psi \rhd P \xrightarrow{\overline{M}\,(\nu\tilde{a})N} P' \qquad \Psi_P \otimes \Psi \rhd Q \xrightarrow{\underline{K}\,N} Q'}{\Psi \rhd P \mid Q \xrightarrow{\tau} (\nu\tilde{a})(P' \mid Q')}\ \tilde{a}\#Q$$

$$\textsc{Par}\ \frac{\Psi_Q \otimes \Psi \rhd P \xrightarrow{\alpha} P'}{\Psi \rhd P|Q \xrightarrow{\alpha} P'|Q}\ \mathrm{bn}(\alpha)\#Q$$

$$\textsc{Scope}\ \frac{\Psi \rhd P \xrightarrow{\alpha} P'}{\Psi \rhd (\nu b)P \xrightarrow{\alpha} (\nu b)P'}\ b\#\alpha, \Psi$$

$$\textsc{Open}\ \frac{\Psi \rhd P \xrightarrow{\overline{M}\,(\nu\tilde{a})N} P'}{\Psi \rhd (\nu b)P \xrightarrow{\overline{M}\,(\nu\tilde{a}\cup\{b\})N} P'}\ \begin{array}{l} b\#\tilde{a}, \Psi, M \\ b \in \mathrm{n}(N) \end{array}$$

$$\textsc{Rep}\ \frac{\Psi \rhd P \mid !P \xrightarrow{\alpha} P'}{\Psi \rhd !P \xrightarrow{\alpha} P'}$$

Results

- Generic results for all instances: Machine-checked proofs
  - compositional semantics
  - bisimulation theory (strong and weak)
  - algebraic properties, congruence

LICS'09
LICS'10
LMCS 2011

- Results for many instances
  - symbolic semantics and bisimulation
  - procedure for computing bisimilarity constraint

SOS'09
JLAP 2012

Algebraic properties

The usual structural laws, in particular

Scope extension

$$P \mid (\nu a)Q \quad \sim \quad (\nu a)(P \mid Q) \qquad a \# P$$

The usual congruence properties, in particular

Compositionality, congruence

Machine-checked proofs

$$P \mathrel{\dot{\sim}_\Psi} Q \Longrightarrow P \mid R \mathrel{\dot{\sim}_\Psi} Q \mid R$$

$$(\forall \widetilde{L}.\ P[\widetilde{a} := \widetilde{L}] \mathrel{\dot{\sim}_\Psi} Q[\widetilde{a} := \widetilde{L}]) \Longrightarrow \underline{M}(\lambda\widetilde{a})N\,.\,P \mathrel{\dot{\sim}_\Psi} \underline{M}(\lambda\widetilde{a})N\,.\,Q$$

Nominal Isabelle Formalization

Mainly by
Jesper Bengtson and Johannes Åman Pohjola

Making it this simple is hard work!

- Easy to get things wrong, even when they are "obviously right"
- Easy to miss a requirement
- Easy to miss generalisations
- Especially true when (name) binding is involved

Easy to get worried!

Isabelle from day 1

- use Interactive theorem prover Isabelle with Nominal package
  - supports nominal datatypes, under active development, produces readable proofs
- use **during** development, not only afterwards!

Adaptable proofs: **case** example

Original rule, tau action: easy induction proofs

$$\textsc{Old-Case}\ \frac{\Psi \vdash \varphi_i}{\Psi \rhd \mathbf{case}\ \widetilde{\varphi} : \widetilde{P} \xrightarrow{\tau} P_i}$$

New rule: more standard, can express the above

$$\textsc{Case}\ \frac{\Psi \rhd P_i \xrightarrow{\alpha} P' \qquad \Psi \vdash \varphi_i}{\Psi \rhd \mathbf{case}\ \widetilde{\varphi} : \widetilde{P} \xrightarrow{\alpha} P'}$$

Change requires re-checking all proofs!

With Isabelle: took a day

Adaptable proofs: higher-order

To get higher-order psi-calculi, just add the following:

Invocation agent $\mathbf{run}\ M$

Clauses $M \Leftarrow P \qquad \mathrm{n}(M) \supseteq \mathrm{n}(P)$

Invocation rule

$$\frac{\Psi \vdash M \Leftarrow P \qquad \Psi \rhd P \xrightarrow{\alpha} P'}{\Psi \rhd \mathbf{run}\ M \xrightarrow{\alpha} P'}$$

Parrow, Borgström, Raabjerg, Åman Pohjola, MSCS 2016

Now prove all meta-theory again!

With Isabelle: meta-theory took a day and a night
More effort: locales, canonical instances, encodings

Broadcast: harder

To get broadcast communication:

Output connectivity $M \mathrel{\dot{\prec}} K$

Input connectivity $K \mathrel{\dot{\succ}} M$

Five new semantics rules, two new actions

$$\textsc{BrOut}\ \frac{\Psi \vdash M \mathrel{\dot{\prec}} K}{\Psi \rhd \overline{M}\,N\,.\,P \xrightarrow{!\overline{K}\ N} P} \qquad \textsc{BrIn}\ \frac{\Psi \vdash K \mathrel{\dot{\succ}} M}{\Psi \rhd \underline{M}(\lambda\tilde{y})N\,.\,P \xrightarrow{?\underline{K}\ N[\tilde{y}:=\tilde{L}]} P[\tilde{y}:=\tilde{L}]}$$

$$\textsc{BrMerge}\ \frac{\Psi_Q \otimes \Psi \rhd P \xrightarrow{?\underline{K}\ N} P' \qquad \Psi_P \otimes \Psi \rhd Q \xrightarrow{?\underline{K}\ N} Q'}{\Psi \rhd P \mid Q \xrightarrow{?\underline{K}\ N} P' \mid Q'}$$

$$\textsc{BrCom}\ \frac{\Psi_Q \otimes \Psi \rhd P \xrightarrow{!\overline{K}\ (\nu\tilde{a})N} P' \qquad \Psi_P \otimes \Psi \rhd Q \xrightarrow{?\underline{K}\ N} Q'}{\Psi \rhd P \mid Q \xrightarrow{!\overline{K}\ (\nu\tilde{a})N} P' \mid Q'}\ \tilde{a}\#Q$$

$$\textsc{BrClose}\ \frac{\Psi \rhd P \xrightarrow{!\overline{K}\ (\nu\tilde{a})N} P' \qquad b \in \mathsf{n}(K)}{\ldots}$$

Quite some work getting it right!
Adds about 12700 lines of Isabelle proofs,
reuses entire Psi codebase of about 20500 lines.

SEFM'11
SoSyM 2015

Even with Isabelle: two years, seven coauthors

The power of Isabelle

What about **combining** higher-order and broadcast?

Re-prove all the meta-theory…

With Isabelle: took HALF a day, mostly waiting!

"could be done by a clever shell script"

Effort

It must take a lot of time to use Isabelle, surely?

- Theory development is not only about doing proofs – most time spent elsewhere
- Doing false proofs is a waste of time
- Correct proofs make it worthwhile!

No worries!

Nominal Transition Systems

Based on slides by Joachim Parrow, OPCT 2017

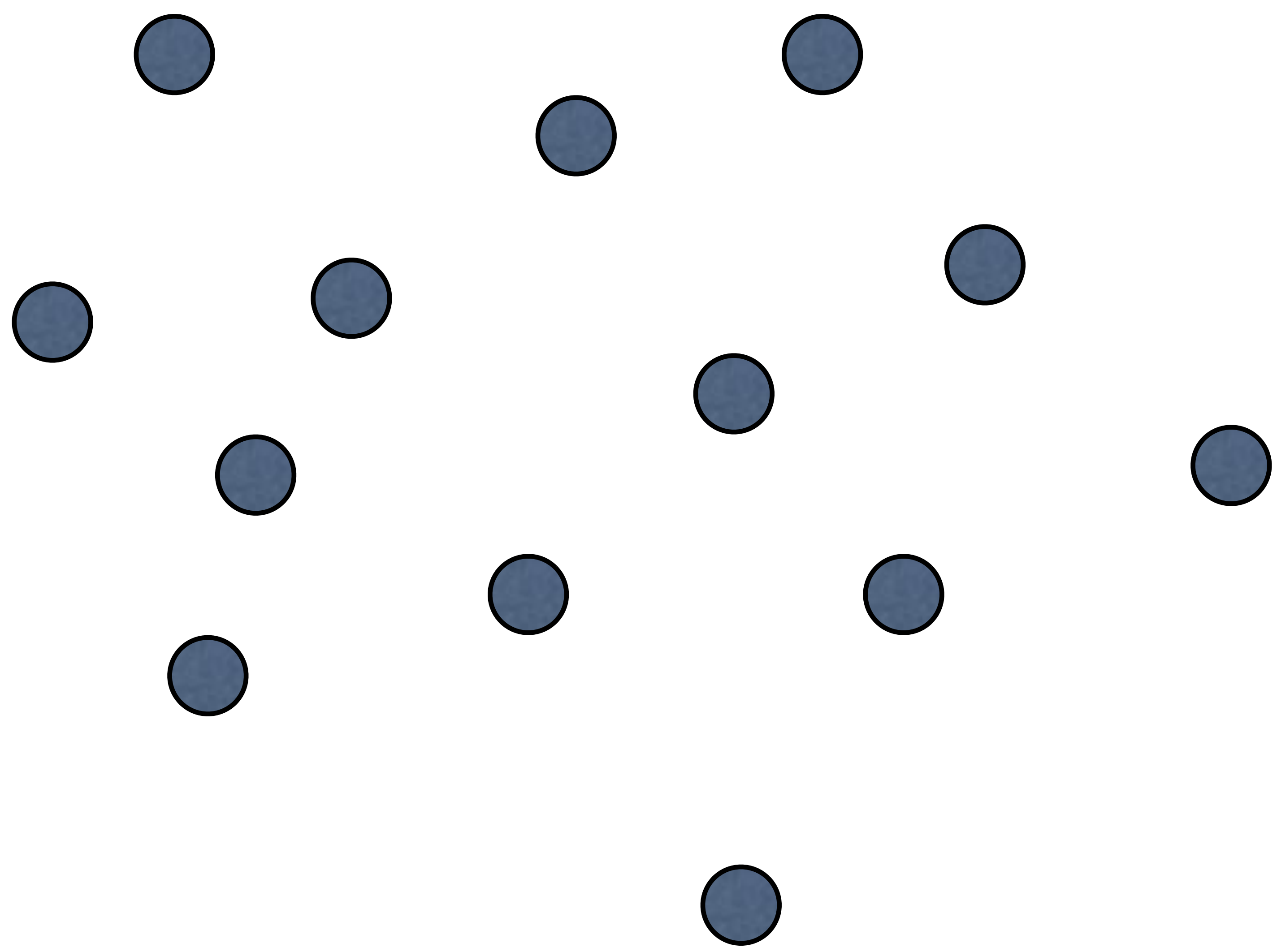
Nominal Transition Systems

What are NTS? Why?

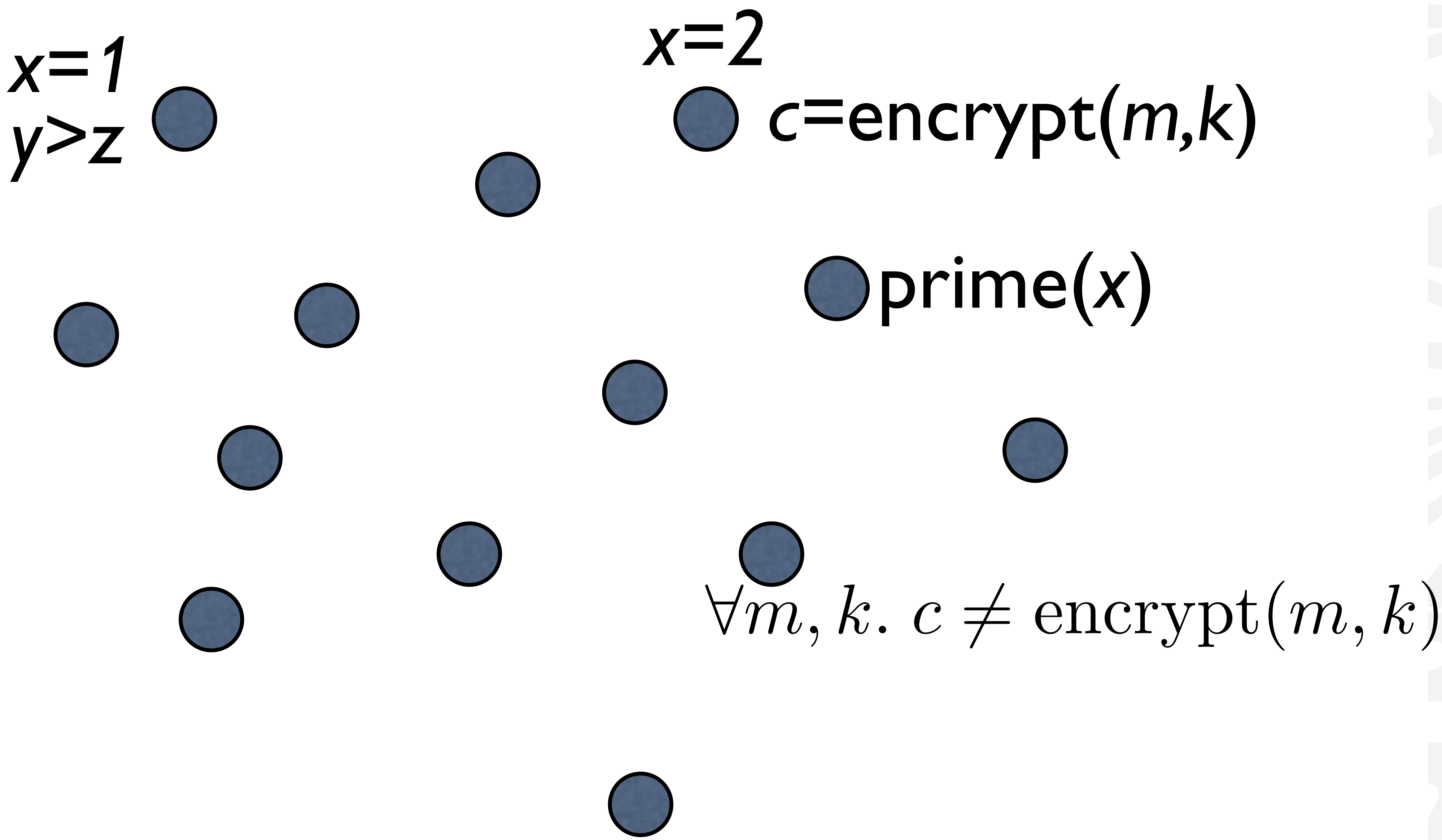
NTS are a **general framework** that fits almost all **advanced process algebras**,

by generalising standard transition systems to include **binders in actions**

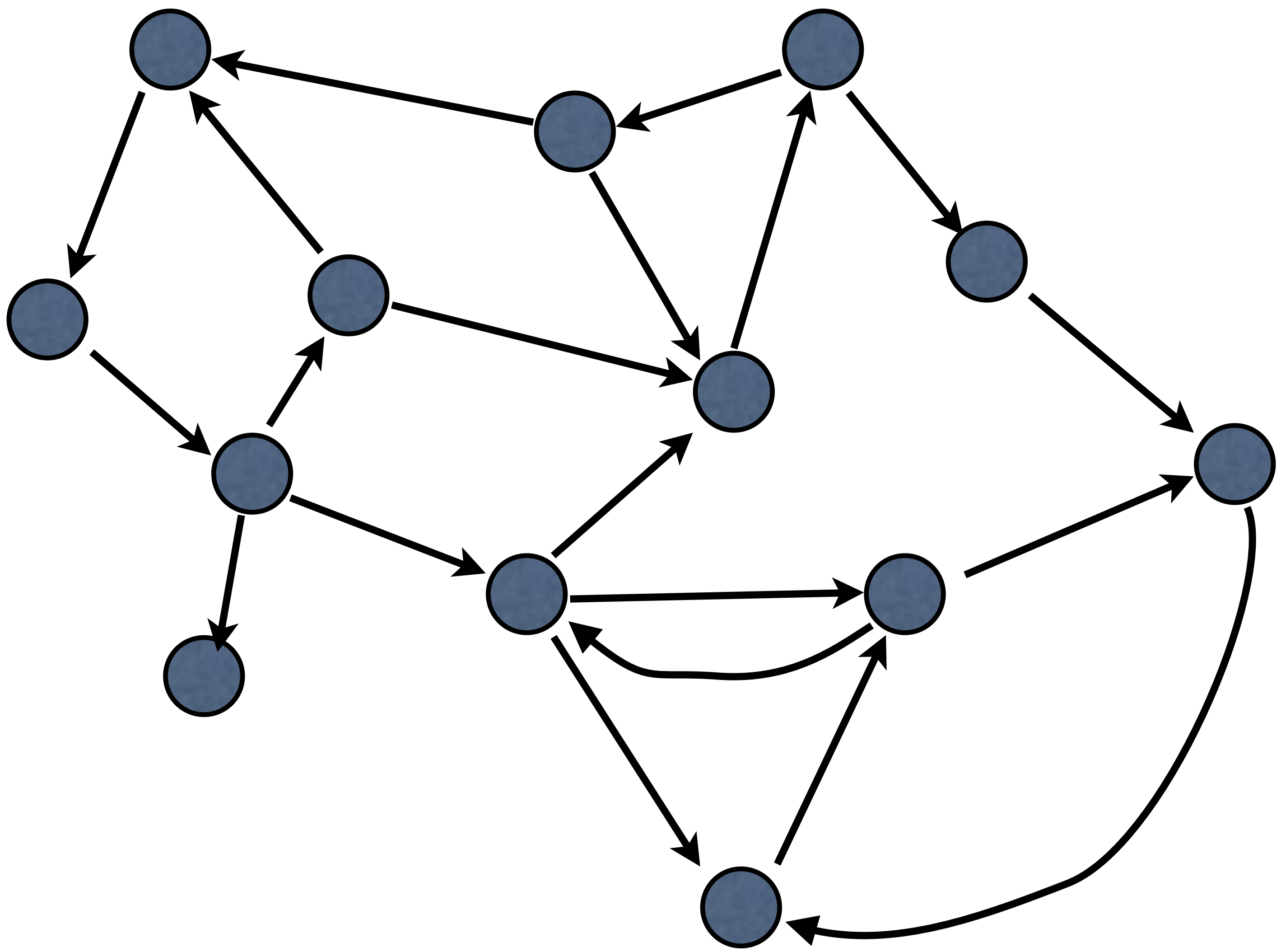
States



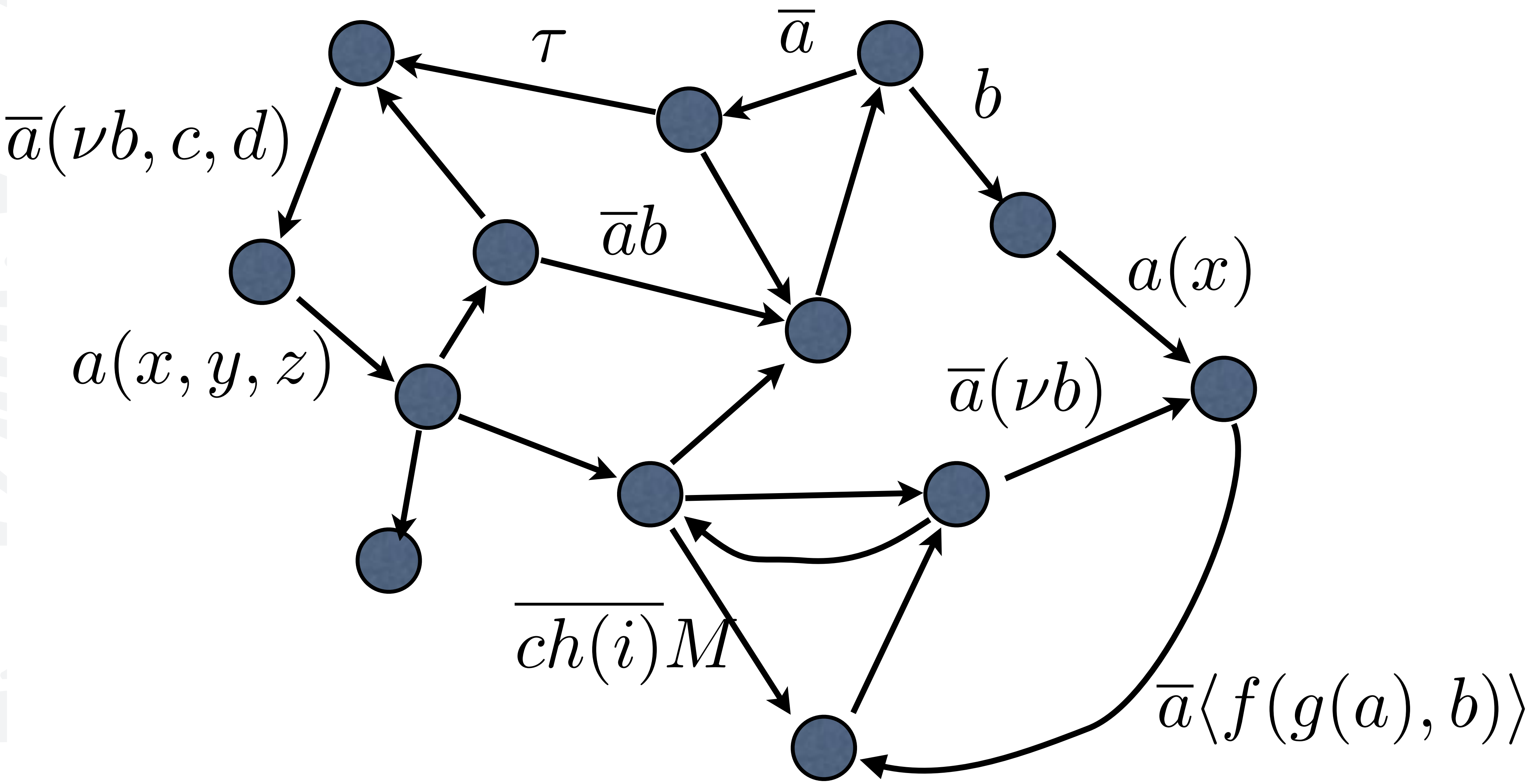
State predicates



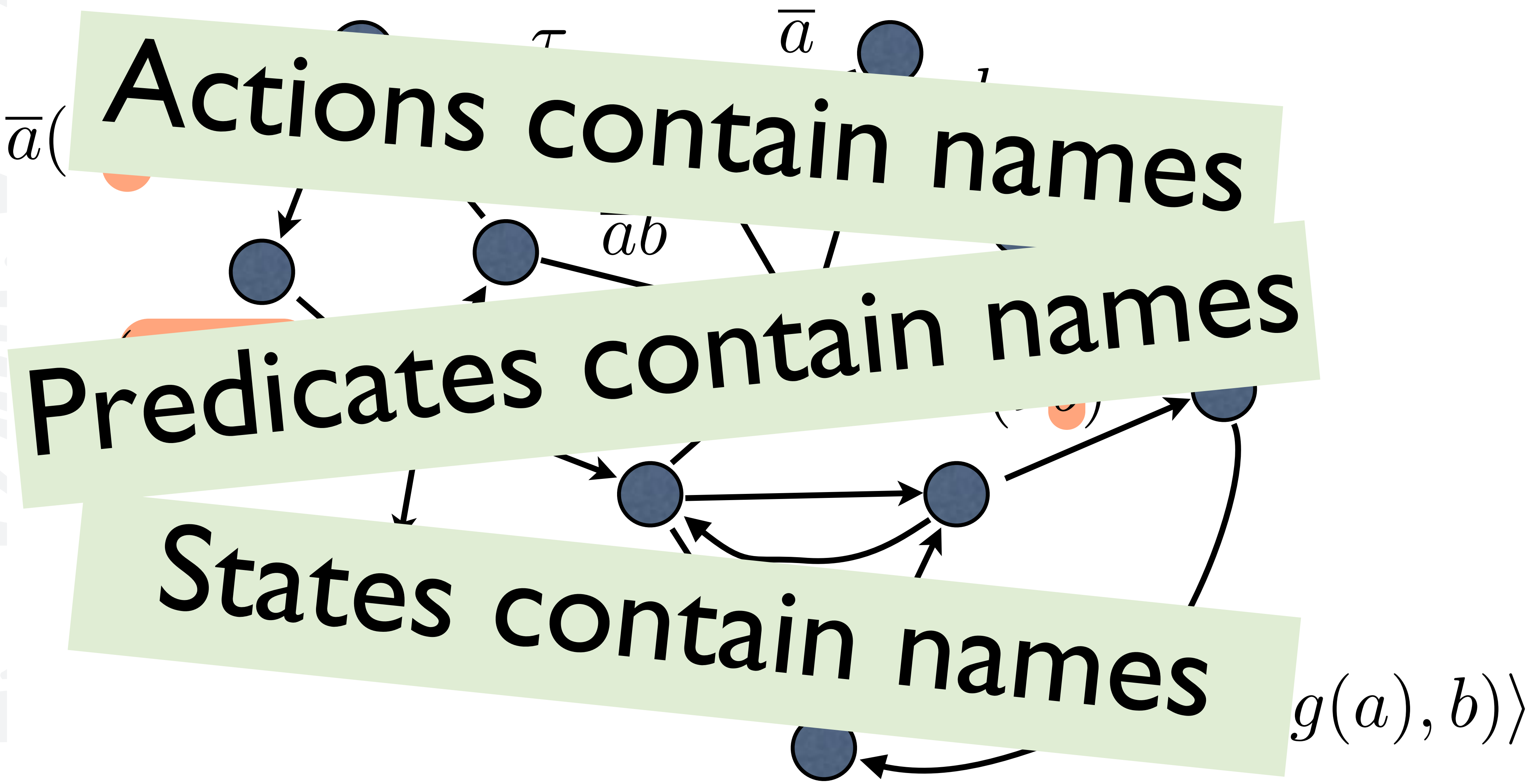
Transitions



Actions


τ
$\overline{a}$
b
$\overline{a}(\nu b, c, d)$
$\overline{a}b$
$a(x)$
$a(x, y, z)$
$\overline{a}(\nu b)$
$\overline{ch(i)}M$
$\overline{a}\langle f(g(a), b)\rangle$

Binding names

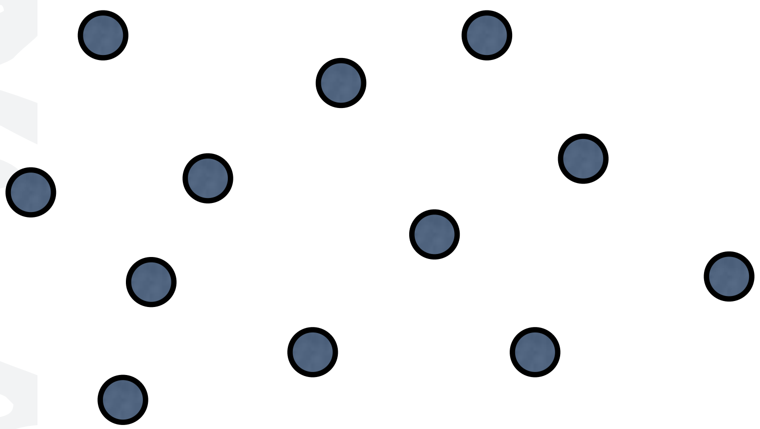


Actions contain names

Predicates contain names

States contain names

States, predicates, and actions



STATES: A nominal set $P, \; Q$

$x = 1$ $x = 2$

$y > z$

$\mathrm{prime}(x)$

$c = \mathrm{encrypt}(m,k)$

$\forall m, k.\ c \neq \mathrm{encrypt}(m,k)$

PRED: A nominal set φ

$\vdash \subseteq$ STATES $\times$ PRED equivariant

τ $\overline{a}$ b

$\overline{a}b$ $a(x)$

$a(x,y,z)$ $\overline{a}(\nu b)$

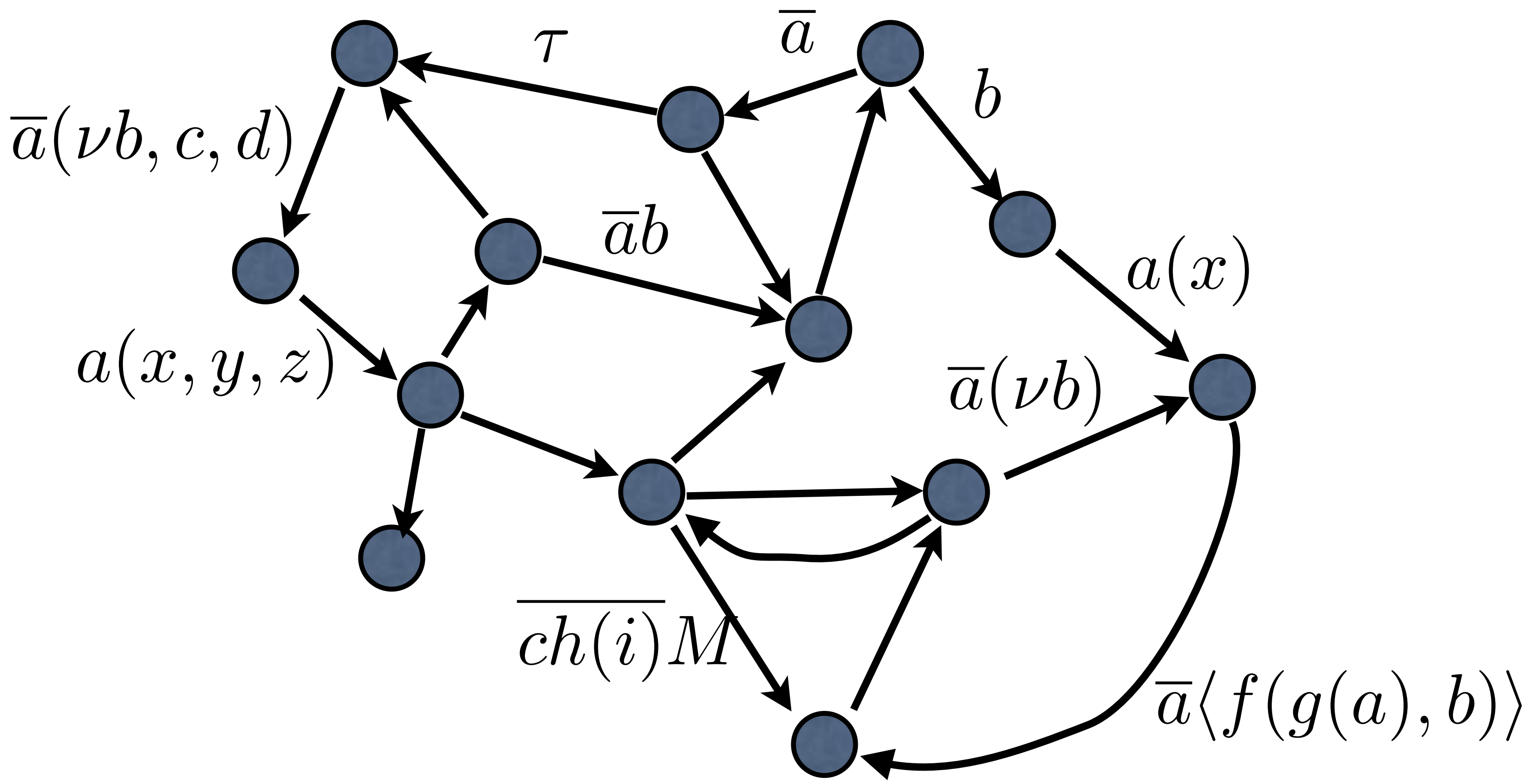
$\overline{ch(i)}M$ $\overline{a}\langle f(g(a),b)\rangle$

ACT: A nominal set α

$\mathrm{bn} : \mathrm{ACT} \to P_{\mathrm{fin}}(\mathcal{N})$ equivariant

$\mathrm{bn}(\alpha) \subseteq \mathrm{supp}(\alpha)$

Transitions



$$\rightarrow \;\subseteq\; \text{STATES} \times [P_{\text{fin}}(\mathcal{N})](\text{ACT} \times \text{STATES}) \quad \text{equivariant}$$

$$(P, <\tilde{b}> (\alpha, Q)) \in \rightarrow \quad \text{implies} \quad \tilde{b} = \text{bn}(\alpha)$$

We write $P \xrightarrow{\alpha} Q$ for $(P, \langle \text{bn}(\alpha) \rangle (\alpha, Q)) \in \rightarrow$

Bisimulation

DEFINITION (Strong Bisimulation)

A symmetric relation R on processes satisfying:

if $R(P,Q)$ then

$$\text{If } P \xrightarrow{\alpha} P' \text{ and } \mathrm{bn}(\alpha)\#Q \text{ then}$$

$$\exists Q'.\, Q \xrightarrow{\alpha} Q' \text{ and } R(P',Q')$$

Simulation

$$\text{If } P \vdash \varphi \text{ then } Q \vdash \varphi$$

Static implication

$$P \dot{\sim} Q \text{ if } R(P,Q) \text{ for some bisimulation } R$$

Modal Logics for Nominal Transition Systems

Based on CONCUR 2015 paper with
Ramūnas Gutkovas
Lars-Henrik Eriksson
Joachim Parrow
Tjark Weber

Presentation based on slides by
Joachim Parrow

Logic

Our objectives:

A set of **formulas** $A, \ B$

A **satisfaction** relation between states and formulas $P \models A$

Expressive wrt existing work

Fully **formal**

Simple

Not objectives: decidability, model checking

Formulas

$$A \quad := \quad \varphi \quad | \quad \langle\alpha\rangle A \quad | \quad \neg A \quad | \quad \bigwedge_{i\in I} A_i$$

Four basic constructors

State Predicates

$P \models \varphi$ *P* satisfies the formula

holds if

$P \vdash \varphi$ the state predicate holds in *P*

Action modality

P can do α and then satisfy A

$$P \models \langle\alpha\rangle A$$

holds if

$$\exists P'.\ P \xrightarrow{\alpha} P' \text{ and } P' \models A$$

we consider formulas up to alpha equivalence, ie

If $a \in \mathrm{bn}(\alpha), b\#\alpha, A$

then $\langle\alpha\rangle A = (a\, b) \cdot (\langle\alpha\rangle A)$

Negation

$$P \models \neg A$$

holds if

$$\text{not} \quad P \models A$$

Conjunction

Assume A_i a formula for each $i \in I$

$$P \models \bigwedge_{i \in I} A_i \qquad \text{if for all } i \in I \text{ it holds } P \models A_i$$

The million dollar question: **which** such conjunctions should be **allowed**?

Safe but not enough

Finite conjunction

As in Hennessy Milner 1985

$$P \models \bigwedge_{i \in I} A_i$$

Allowed only for finite *I*

Same as binary conjunction $A_1 \wedge A_2$

Easy to make fully formal

Quite **limited expressiveness** (suitable only for finite-branching transition systems)

Arbitrary conjunction

Needs substantial restrictions

As in Milner 1989

$P \models \bigwedge_{i \in I} A_i$ Allowed for any *I*

Enormous expressiveness:
greater than the systems we study!

Formulas might not be finitely supported,
alpha-conversion might be impossible

Uniformly bounded conjun…

As in Abramsky 1991

Standard but not enough

$$P \models \bigwedge_{i \in I} A_i$$

Allowed for any *I* such that conjuncts have common finite support

for some finite set of names *S*

$$\forall i \in I.\, \mathrm{supp}(A_i) \subseteq S$$

Still of limited expressiveness

OK to make fully formal?

Example: quantifiers

$$P \models \forall x \in \mathcal{N}.\ A$$

holds if

some substitution function

$$\text{for all } z \in \mathcal{N} \text{ it holds } P \models A[x := z]$$

Can this be represented as

$$\forall x \in \mathcal{N}.\ A = \bigwedge_{z \in \mathcal{N}} A[x := z] \qquad ?$$

$$\forall x \in \mathcal{N}.\ A = \bigwedge_{z \in \mathcal{N}} A[x := z]$$

Is this conjunction uniformly bounded?

No. At least not if

$$z \in \mathrm{supp}(A[x := z])$$

Quantification cannot be expressed by uniformly bounded conjunction!

Finitely supported conjunction

Our contribution

$\bigwedge_{i \in I} A_i$ requires that the set of formulas $\{A_i \,|\, i \in I\}$ has finite support S

Assume $\boldsymbol{F}$ is the set of formulas supported by S.

Consider the different formulas $\bigwedge\{A \mid A \in \boldsymbol{B}\}$

where $\boldsymbol{B}$ ranges over the subsets of $\boldsymbol{F}$.

By Cantor's Theorem, we have a contradiction.

Solution: cardinality bound on conjunction width

$$\forall x \in \mathcal{N}.\ A = \bigwedge_{z \in \mathcal{N}} A[x := z] \quad ?$$

Is this conjunction finitely supported?

Yes!
Assuming substitution is equivariant.

Expressiveness

Dualities

$$\bigvee_{i \in I} A_i = \neg \bigwedge_{i \in I} \neg A_i$$

$$[\alpha]A = \neg\langle\alpha\rangle\neg A$$

Expressiveness

Quantifiers

$$\forall x.\ A = \bigwedge_{z \in V} A[x := z]$$

$$\exists x.\ A = \bigvee_{z \in V} A[x := z]$$

Assumes V is finitely supported and substitution is equivariant

Expressiveness

Fresh Quantifier

$$P \models \text{И} x.\, A \text{ if for some } n \# P \text{ it holds } P \models (x\, n) \cdot A$$

$$\text{И} x.\, A = \bigvee_{S \in \mathrm{COF}} \bigwedge_{n \in S} (x\, n) \cdot A$$

COF is the set of cofinite sets of names

There is a cofinite set such that A holds for all its members

Expressiveness

Next step modality

$$\langle\rangle A = \bigvee_{\alpha \in \mathrm{ACT}} \langle\alpha\rangle A$$

$$\mathrm{bn}(\alpha) \# A$$

Fixpoints

minimal fixpoint defined as disjunction of all unfoldings

With next and fixpoints
we get all of CTL* Emerson 1997

Applications

F Finite conjunction
A Arbitrary conjunction
U Uniformly bounded conjunction

Hennessy-Milner Logic for CCS — *Hennessy, Milner 1985* F; *Milner 1989* A; *Abramsky 1991* U

for pi-calculus — *Milner, Parrow, Walker 1993* U

for value passing — *Hennessy, Liu 1995* F + quantifiers

for spi-calculus — *Frendrup, Huttel, Jensen 2002* A

for applied pi-calculus — *Pedersen, 2006* F

for fusion calculus — *Haugstad, Terkelsen, Vindum 2006* A

for multi-labelled systems — *De Nicola, Loreti 2008* F + quantifiers

---

for concurrent constraint calculus — *Buscemi, Montanari 2007*

for psi-calculi — *Bengtson et al 2011*

Yet no modal logic

Adequacy

A kind of sanity check:

Most often: bisimulation

If two states ``**behave the same**´´ then they satisfy exactly the **same** formulas

If two states do **not** ``behave the same´´ then there is a formula satisfied by **one and not the other**

Bisimulation

DEFINITION (Bisimulation)
A symmetric relation R on states satisfying:

if $R(P,Q)$ then

$$\text{If } P \xrightarrow{\alpha} P' \text{ and } \mathrm{bn}(\alpha)\#Q \text{ then } \exists Q'.\, Q \xrightarrow{\alpha} Q' \text{ and } R(P',$$

$$\text{If } P \models \varphi \text{ then } Q \models \varphi$$

$P \mathrel{\dot\sim} Q$ if $R(P,Q)$ for some bisimulation Q

THEOREM (Adequacy)
$P \mathrel{\dot\sim} Q$ iff for all formulas A: $P \models A$ iff $Q \models A$

$P \dot{\sim} Q$ iff for all formulas A: $P \models A$ iff $Q \models A$

In direction $\Leftarrow$ show that

logical equivalence

$$\doteq \text{ defined as } \{(P,Q) \mid \forall A.\, P \models A \text{ iff } Q \models A\}$$

is a bisimulation.

Assume not, then P has an α-transition to P'

that Q cannot simulate:

For each α-derivative Q' there is a distinguishing

formula A between P' and Q'.

Let B be the conjunction of all these A (one for each Q')

Then $P \models \langle\alpha\rangle B$ and not $Q \models \langle\alpha\rangle B$

Contradiction!

Let B be the conjunction of all these A (one for each Q')

Can this conjunction be defined in the logic?

If the transition system is **finitely branching**

then there are finitely many Q'

so **finite conjunction** suffices

Eg CCS with guarded recursion

If all the formulas A have

a **common finite** support then

uniformly bounded conjunction suffices

Eg pi-calculus

In general use finitely supported conjunction

Arbitrary nominal transition systems

In general use finitely supported conjunction

Lemma: If $P' \models A \ \wedge\ Q' \not\models A$ then

$$\exists B.\ \ P' \models B \ \wedge\ Q' \not\models B \ \ \wedge\ \ \mathrm{supp}(B) \subseteq \mathrm{supp}(P')$$

If there is a distinguising formula for *P*' and *Q*',

then there is one with the support bounded by *P*'

Proof idea:

Let `PERM` be the name permutations that fix *P'*

$$B = \bigwedge_{\pi \in \mathrm{PERM}} \pi \cdot A$$

Formalisation

Out of which 150 loc are definitions and theorems

All definitions and the adequacy theorem formalised in **Nominal Isabelle** (~2700 loc)

Significant new ideas for alpha-equivalence and finite support in data types with **infinitary** constructors.

First ever mechanisation of an **infinitely branching nominal datatype.**

Equivalences and Modal Logics for Unobservable Actions

Presentation based on slides by
Joachim Parrow

Based on FORTE 2017 paper with
Ramūnas Gutkovas
Lars-Henrik Eriksson
Joachim Parrow
Tjark Weber

Weak = disregard *silent* transitions

τ action with empty support (implies $\mathrm{bn}(\tau)=\varnothing$) representing an unobservable action

$$P \xrightarrow{\tau} P'$$

P can evolve to P'

without the environment noticing

without interacting with the environment

spontaneously

silently

Weak transitions

$P \Rightarrow P'$ defined inductively as

$$P = P' \;\vee\; P \xrightarrow{\tau} \circ \Rightarrow P'$$

$P \overset{\alpha}{\Rightarrow} P'$ defined as $P \Rightarrow \circ \xrightarrow{\alpha} \circ \Rightarrow P'$

$P \overset{\hat{\alpha}}{\Rightarrow} P'$ defined as $\begin{cases} P \Rightarrow P' & \text{if } \alpha = \tau \\ P \overset{\alpha}{\Rightarrow} P' & \text{otherwise} \end{cases}$

P can evolve to P' through zero or more transitions with observable content α

Simulation

DEFINITION (simulation)

A relation *R* on states satisfying: if *R*(*P*,*Q*) then

$$\text{If } P \xrightarrow{\alpha} P' \text{ and } \mathrm{bn}(\alpha)\#Q \text{ then } \exists Q'.\, Q \xrightarrow{\alpha} Q' \text{ and } R(P', Q$$

Weak simulation

DEFINITION (weak simulation)

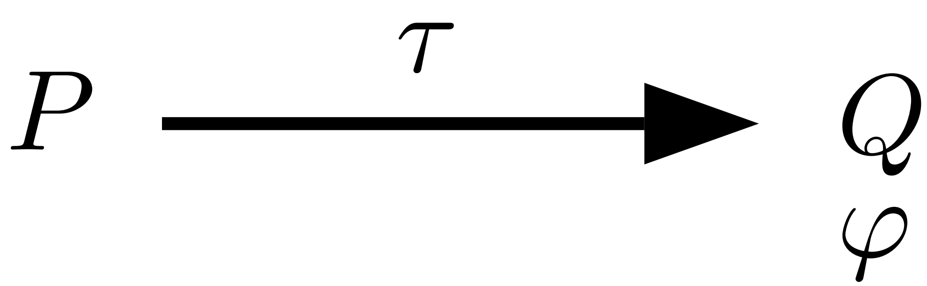
A relation *R* on states satisfying: if *R*(*P*,*Q*) then

$$\text{If } P \overset{\alpha}{\to} P' \text{ and } \mathrm{bn}(\alpha)\#Q \text{ then } \exists Q'.\, Q \overset{\hat{\alpha}}{\Rightarrow} Q' \text{ and } R(P',Q'$$

Static implication?

Can we re-use the static implication **NO!**

If $P \vdash \varphi$ then $Q \vdash \varphi$



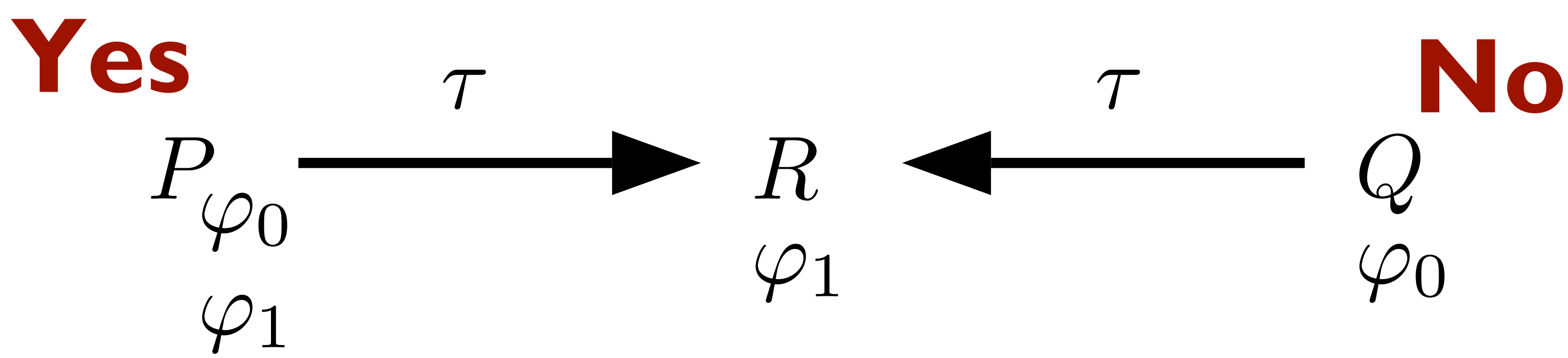
Example: transition system with two states, one transition, and one state predicate

Should *P* and *Q* be equivalent?

YES!

Weak static implication?

$$\text{If } P \vdash \varphi \text{ then } Q \Rightarrow Q' \vdash \varphi \qquad (*)$$



P and Q are weakly similar and satisfy (*)

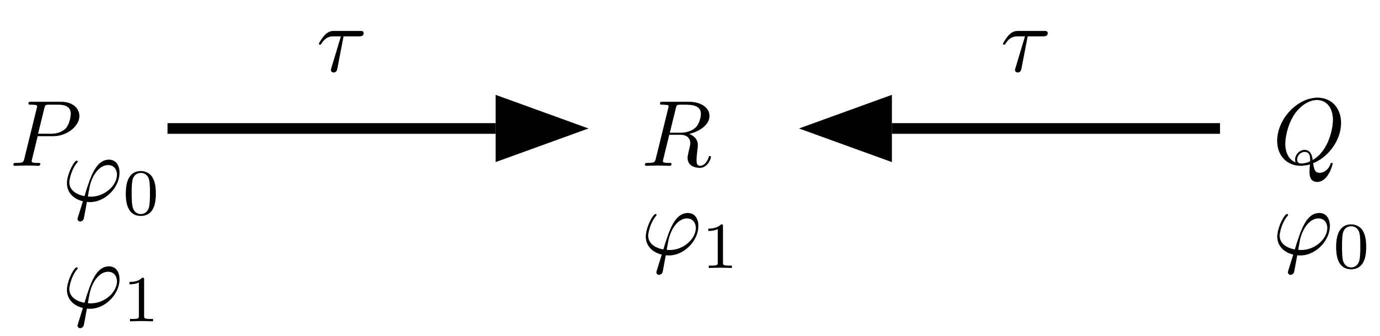
Are P and Q observationally equivalent?

Observe φ_1 and then observe φ_0

Weak static implication!

S is a weak static implication if *S*(*P*,*Q*) implies

$$\text{If } P \vdash \varphi \text{ then } Q \Rightarrow Q' \vdash \varphi \text{ and } S(P, Q')$$

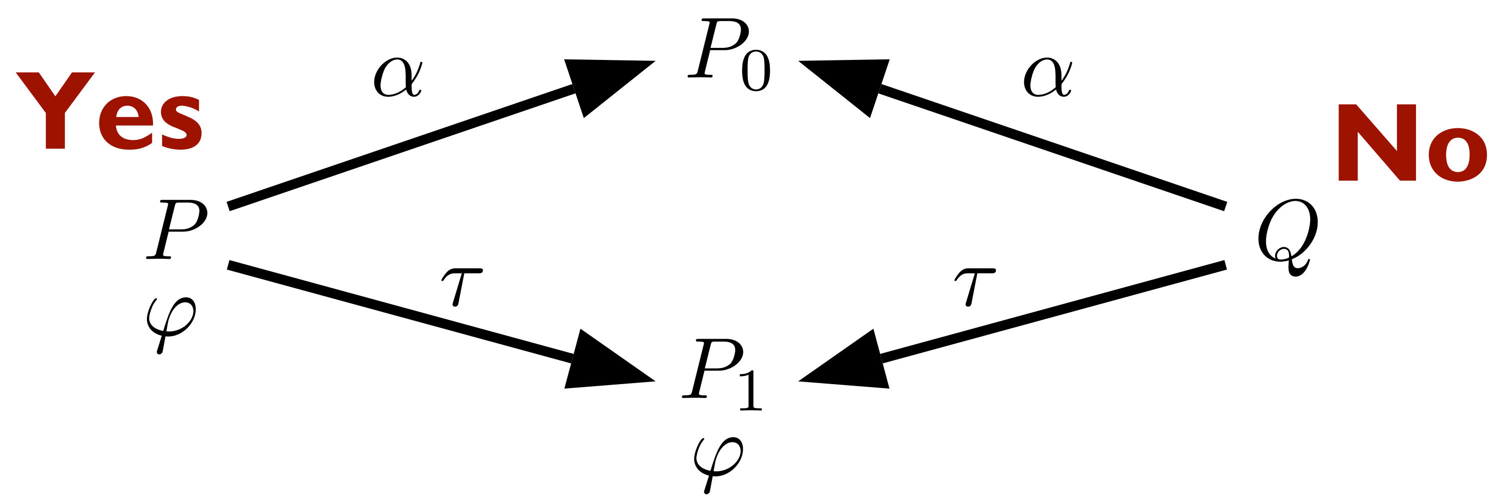


$\{(P, Q), (P, R)\}$ NOT a WSI

Weak static implication

If $P \vdash \varphi$ then $Q \Rightarrow Q' \vdash \varphi$ and $S(P, Q')$

Not enough by itself!

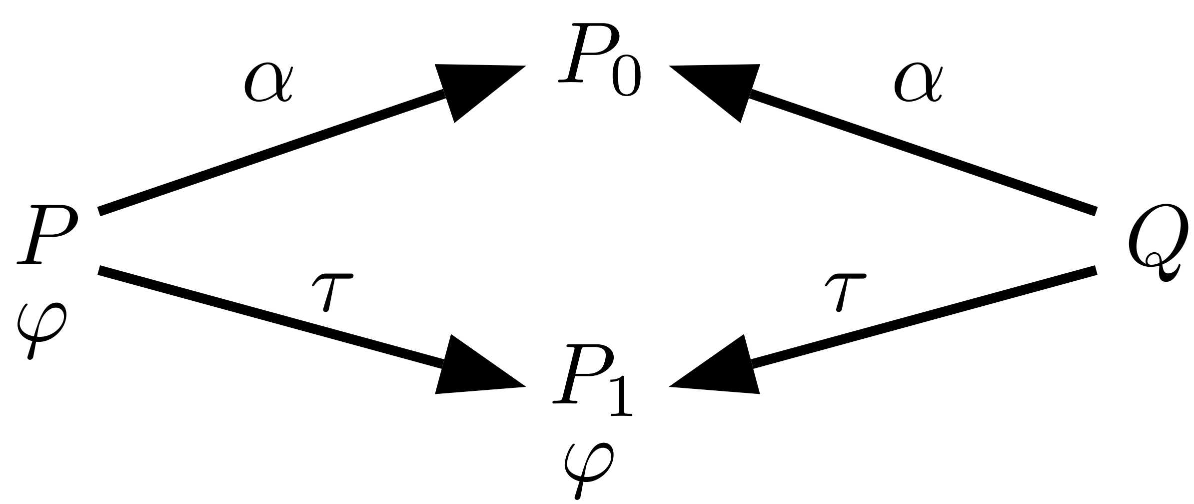


P and Q are weakly similar and the relation $\{(P, Q), (P, P_1)\}$ satisfies (*)

Are P and Q observationally equivalent?

Observe φ and then perform α

Weak static implication!



$\{(P,Q),(P_0,P_0),(P_1,P_1)\}$ is a weak simulation
is NOT a WSI

$\{(P,Q),(P,P_1)\}$ is a WSI
is NOT a weak simulation

Must require the relation to be **both** WSI **and** weak simulation!

Weak bisimulation

DEFINITION

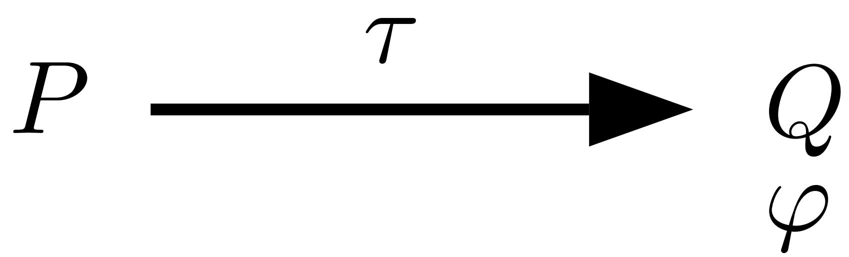
A *weak bisimulation* is a symmetric relation *R* on states which is **both** a weak simulation **and** a weak static implication

$R(P, Q)$ implies:

If $P \xrightarrow{\alpha} P'$ and $\mathrm{bn}(\alpha)\#Q$ then $\exists Q'.\, Q \overset{\hat{\alpha}}{\Rightarrow} Q'$ and $R(P', Q')$

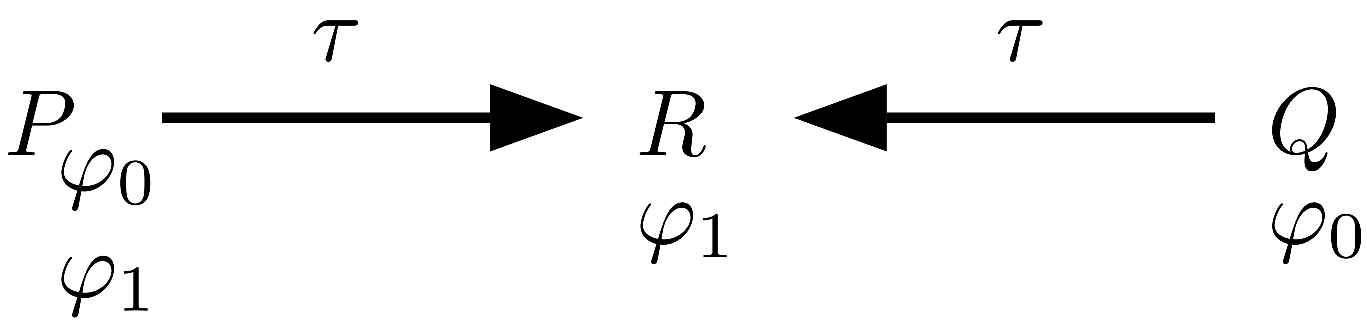
If $P \vdash \varphi$ then $Q \Rightarrow Q' \vdash \varphi$ and $R(P, Q')$

$P \dot{\approx} Q$ if $R(P, Q)$ for some weak bisimulation R



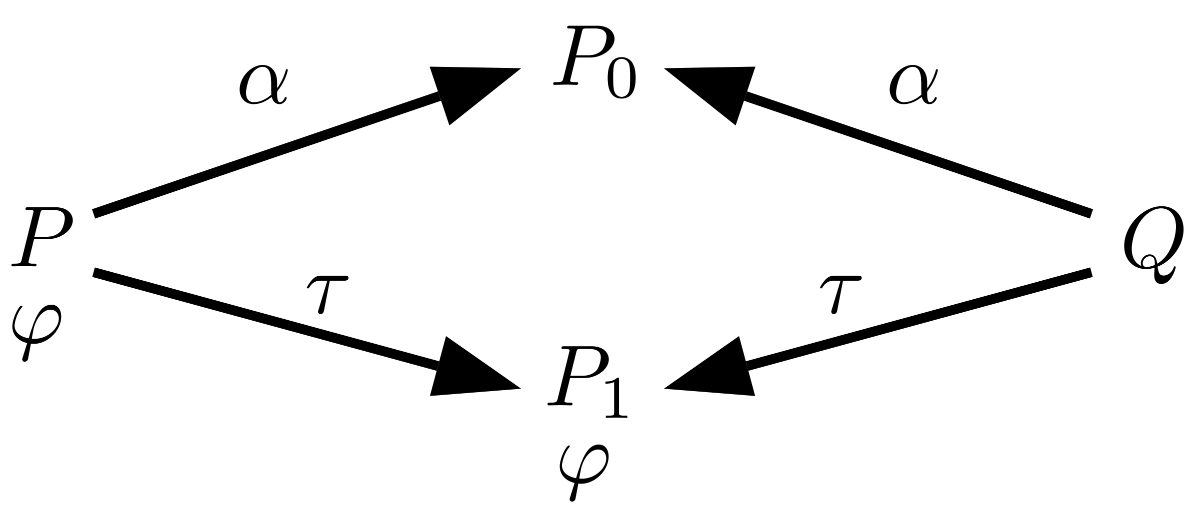
$P \,\dot{\approx}\, Q$

$\{(P,Q),(Q,Q)\}$ is a weak simulation and a WSI



$P \,\dot{\not\approx}\, Q$

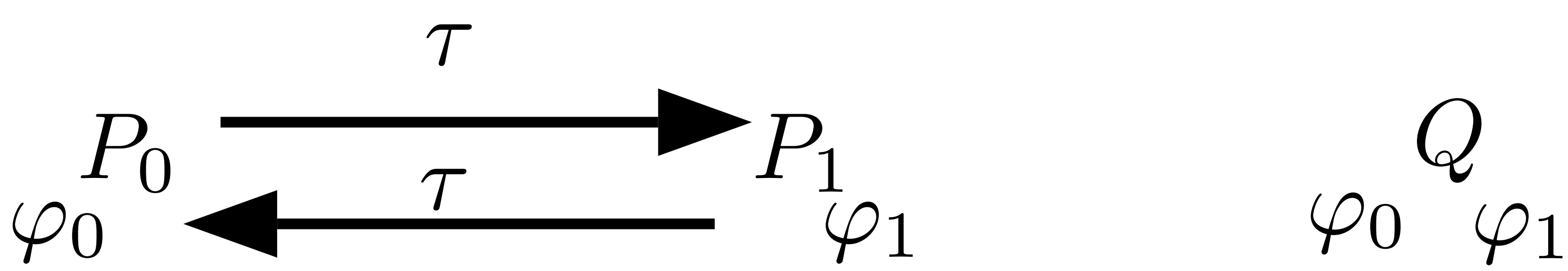
No relation is a WSI



$P \,\dot{\not\approx}\, Q$

No relation is both a weak simulation and a WSI

Exercise



Which of the three states are weakly bisimilar?

Note: $\varphi_0 \wedge \varphi_1$ is not a state predicate

All of them!

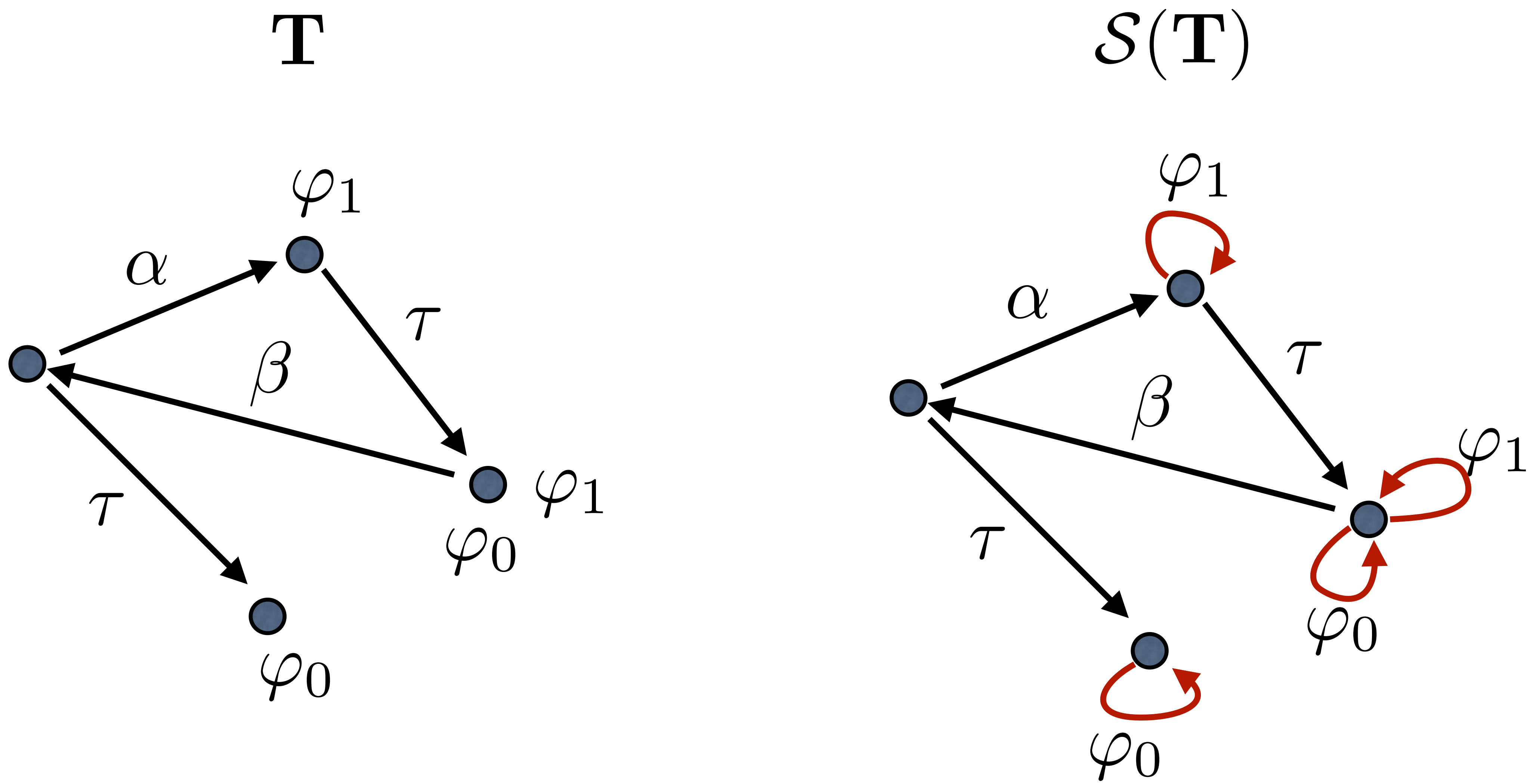
Let *U* be the universal relation on all three states

U is a weak simulation

U is a weak static implication

Eliminating state predicates

Eliminating state predicates



Transformation on transition systems

Replace state predicates by self-loop transitions

Result

THEOREM (State predicate elimination)

$$P \mathrel{\dot{\approx}_{\mathbf{T}}} Q \text{ iff } P \mathrel{\dot{\approx}_{\mathcal{S}(\mathbf{T})}} Q$$

For a corresponding transformation on formulas, replacing predicates by actions

$$P \models_{\mathcal{S}(\mathbf{T})} A \quad \text{iff} \quad P \models_{\mathbf{T}} \mathcal{S}^{-1}(A)$$

Conclusion

- Generic HML
- Suitable for embedding other logics in
  - Guaranteed soundness!
- A sublogic characterises weak bisimulation
- A uniform extension/encoding for
  - early bisimulation, early congruence, late bisimulation, late congruence, *open bisimulation*, hyperbisimulation