A Theory of System Behaviour in the presence of Node and Link Failure

CONCUR, August 2005.

Adrian Francalanza and Matthew Hennessy

{adrianf, matthewh}@sussex.ac.uk.

Starting Point: $D\pi$

$$P \mid (\nu\, n)Q \mid R$$

Starting Point: $D\pi$

$$l[[P]] \mid (\nu k)(k[[Q]] \mid l[[R]])$$

Starting Point: $D\pi$

Processes

$$P, Q \quad ::= \quad a!\langle v\rangle.P \quad | \; a?(x).P \quad | \; *P$$
$$| \quad \text{if } v = u \text{ then } P \text{ else } Q$$
$$| \quad \textbf{stop} \quad | \; P|Q \quad | \; (\nu a)P$$
$$| \quad \text{go } l.P \quad (\textit{migration})$$

Systems

$$N, M \quad ::= \quad l[\![P]\!] \quad (\textit{located process})$$
$$| \quad (\nu\, n)N \quad (\textit{network scoping})$$
$$| \quad N|M \quad (\textit{parallel systems})$$

Starting Point: $D\pi$

(r-comm)

$$l[\![a!\langle n\rangle.P]\!] \mid l[\![a?(x).Q]\!] \longrightarrow l[\![P]\!] \mid l[\![Q\{^{n}/_{x}\}]\!]$$

(r-go)

$$l[\![\text{go } k.P]\!] \longrightarrow k[\![P]\!]$$

Starting Point: $D\pi$

(r-comm)

$$l[\![a!\langle n\rangle.P]\!] \mid l[\![a?(x).Q]\!] \longrightarrow l[\![P]\!] \mid l[\![Q\{^{n}/_{x}\}]\!]$$

(r-go)

$$l[\![\text{go } k.P]\!] \longrightarrow k[\![P]\!]$$

Starting Point: $D\pi$

(r-comm)

$$l[\![a!\langle n\rangle.P]\!] \mid l[\![a?(x).Q]\!] \longrightarrow l[\![P]\!] \mid l[\![Q\{^{n}/_{x}\}]\!]$$

(r-go)

$$l[\![\text{go } k.P]\!] \longrightarrow k[\![P]\!]$$

Motivating Example (1)

$$\text{server} \Leftarrow (\nu\, \mathit{data}) \left(\begin{array}{l} l[\![\mathit{req}?(x_1, x_2).\mathit{data}!\langle x_1, x_2\rangle]\!] \\ \\ \mid l[\![\mathit{data}?(y_1, y_2).y_2!\langle \mathbf{lookup}(y_1)\rangle]\!] \end{array} \right)$$

$$\text{serverLoc} \Leftarrow (\nu\, \mathit{data}) \left(\begin{array}{l} l[\![\mathit{req}?(x_1, x_2).\text{go}\, k_1.\mathit{data}!\langle x_1, x_2\rangle]\!] \\ \\ \mid k_1[\![\mathit{data}?(y_1, y_2).\text{go}\, l.y_2!\langle \mathbf{lookup}(y_1)\rangle]\!] \end{array} \right)$$

Motivating Example (1)

$$\text{server} \Leftarrow (\nu\, data)\left(\begin{array}{l} l[\![req?(x_1, x_2).data!\langle x_1, x_2\rangle]\!] \\ \\ \mid l[\![data?(y_1, y_2).y_2!\langle \mathbf{lookup}(y_1)\rangle]\!] \end{array}\right)$$

$$\text{serverLoc} \Leftarrow (\nu\, data)\left(\begin{array}{l} l[\![req?(x_1, x_2).\text{go}\, k_1.data!\langle x_1, x_2\rangle]\!] \\ \\ \mid k_1[\![data?(y_1, y_2).\text{go}\, l.y_2!\langle \mathbf{lookup}(y_1)\rangle]\!] \end{array}\right)$$

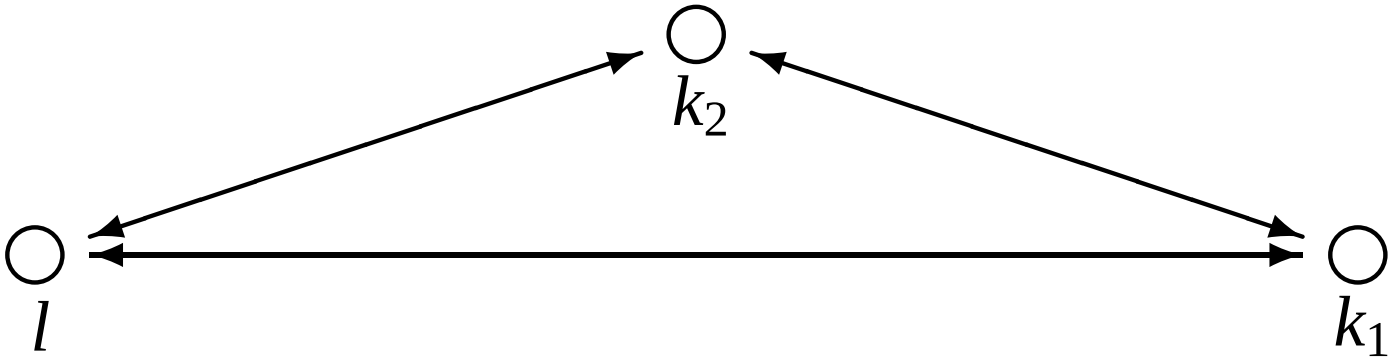
Motivating Example (1)

$$\mathsf{serverLnk} \Leftarrow (\nu\, \mathit{data})\left(\begin{array}{l} l\left[\!\left[\mathit{req}?(x_1, x_2).(\nu \mathit{sync})\left(\begin{array}{l} \text{go } k_1.\mathit{data}!\langle x_1, \mathit{sync}\rangle \\ \mid \text{go } k_2, k_1.\mathit{data}!\langle x_1, \mathit{sync}\rangle \\ \mid \mathit{sync}?(z).x_2!\langle z\rangle \end{array}\right)\right]\!\right] \\ \\ \mid k_1\left[\!\left[\mathit{data}?(y_1, y_2).\left(\begin{array}{l} \text{go } l.y_2!\langle \mathbf{lookup}(y_1)\rangle \\ \mid \text{go } k_2, l.y_2!\langle \mathbf{lookup}(y_1)\rangle \end{array}\right)\right]\!\right] \end{array}\right)$$

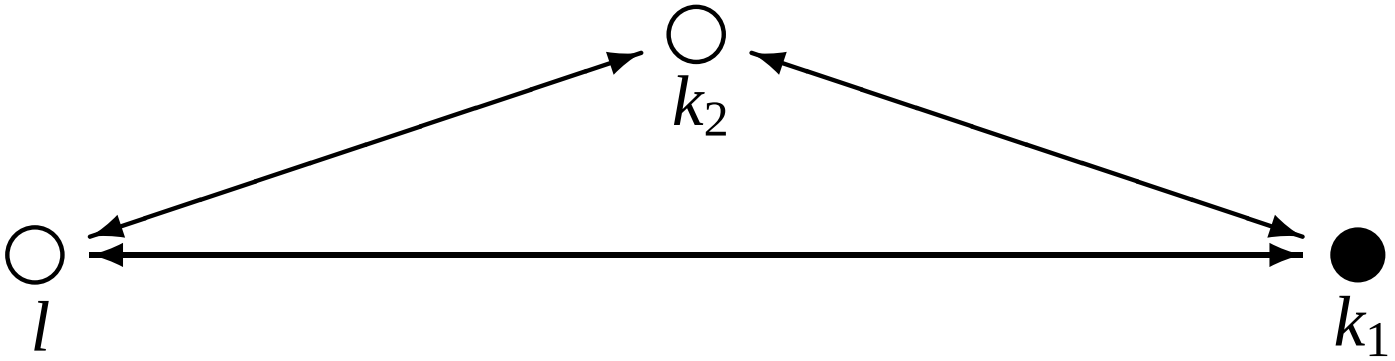
Motivating Example (1)

$$\text{serverLnk} \Leftarrow (\nu\, data)\left(\begin{array}{l} l\left[\!\left[req?(x_1, x_2).(\nu sync)\left(\begin{array}{l} \text{go}\, k_1.data!\langle x_1, sync\rangle \\ \mid \text{go}\, k_2,\ k_1.data!\langle x_1, sync\rangle \\ \mid sync?(z).x_2!\langle z\rangle \end{array}\right)\right]\!\right] \\ \\ \mid k_1\left[\!\left[data?(y_1, y_2).\left(\begin{array}{l} \text{go}\, l.y_2!\langle \textbf{lookup}(y_1)\rangle \\ \mid \text{go}\, k_2,\ l.y_2!\langle \textbf{lookup}(y_1)\rangle \end{array}\right)\right]\!\right] \end{array}\right)$$

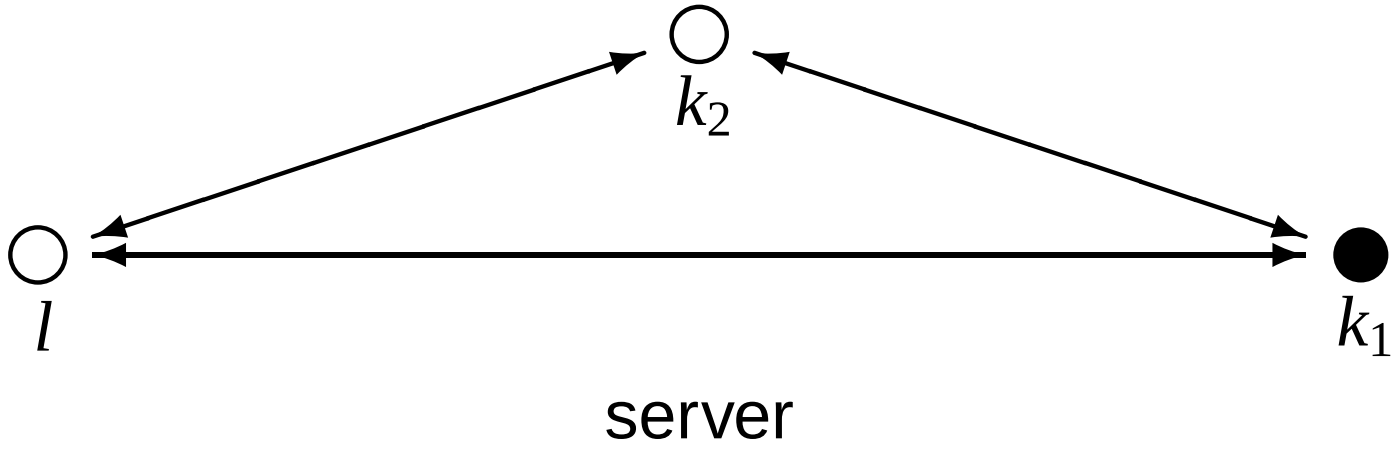
The Underlying Network



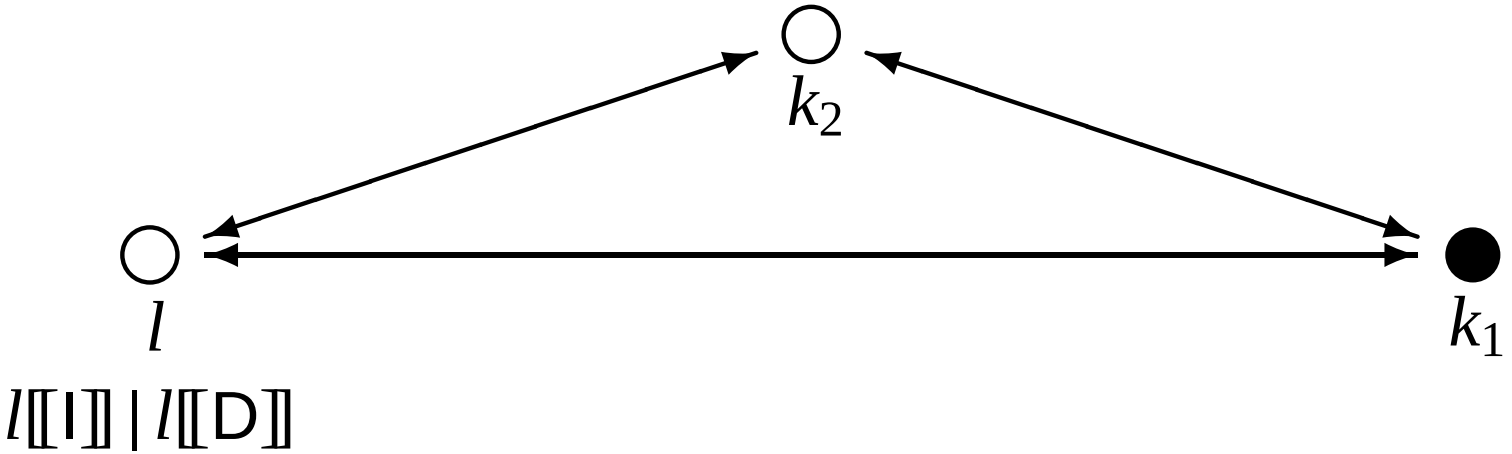
The Underlying Network



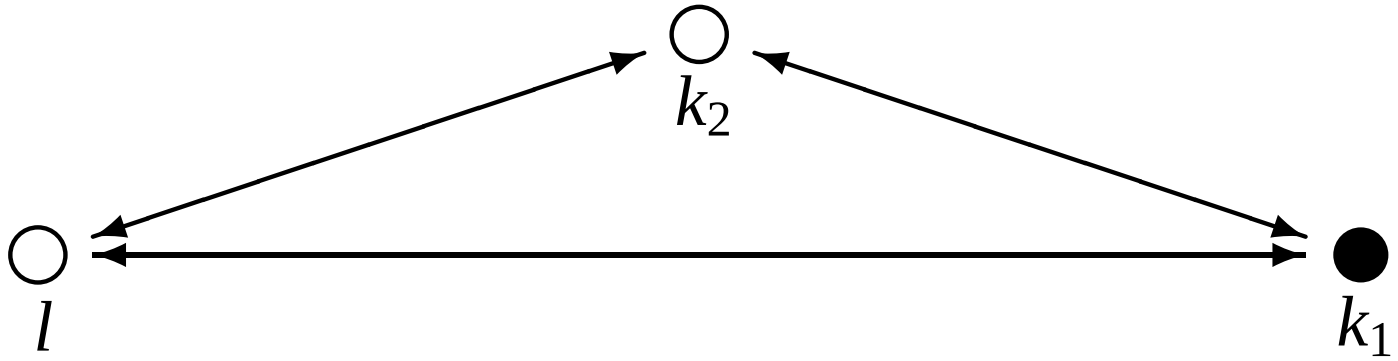
The Underlying Network



The Underlying Network

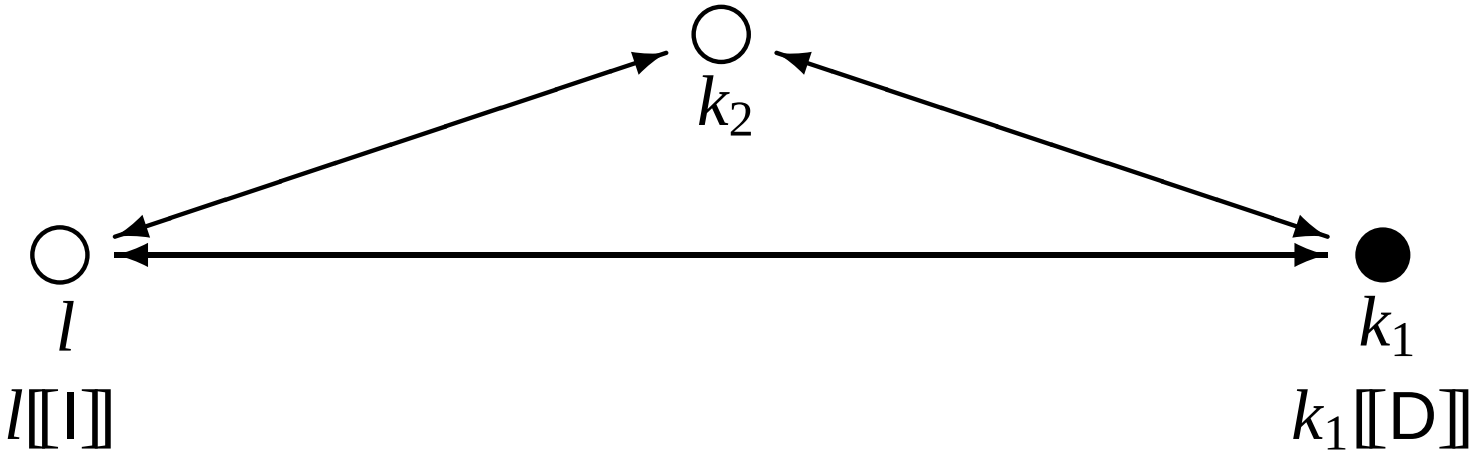


The Underlying Network

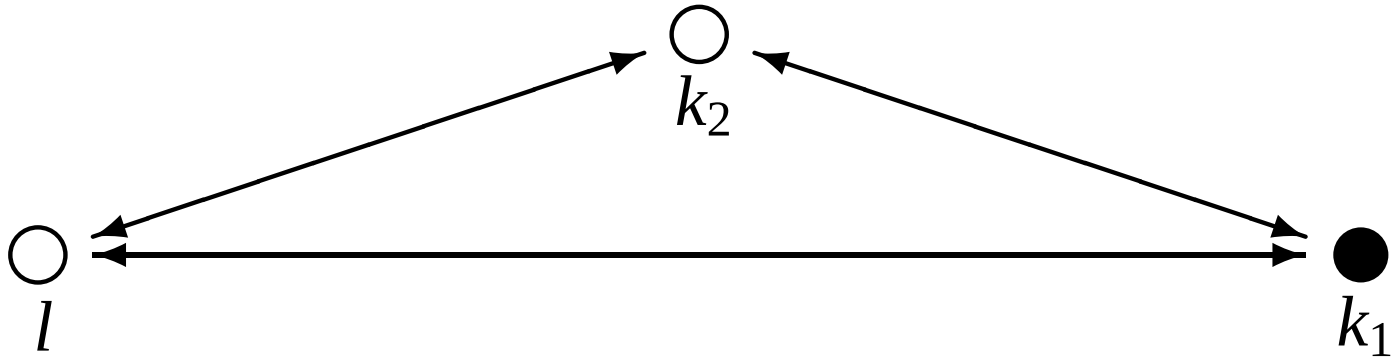


serverLoc

The Underlying Network

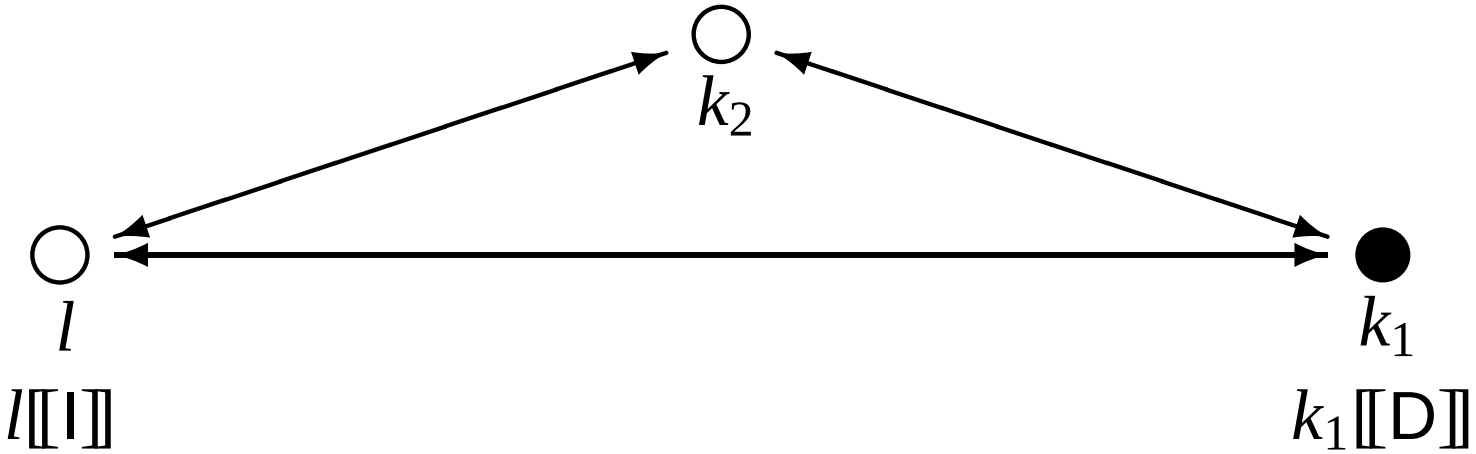


The Underlying Network

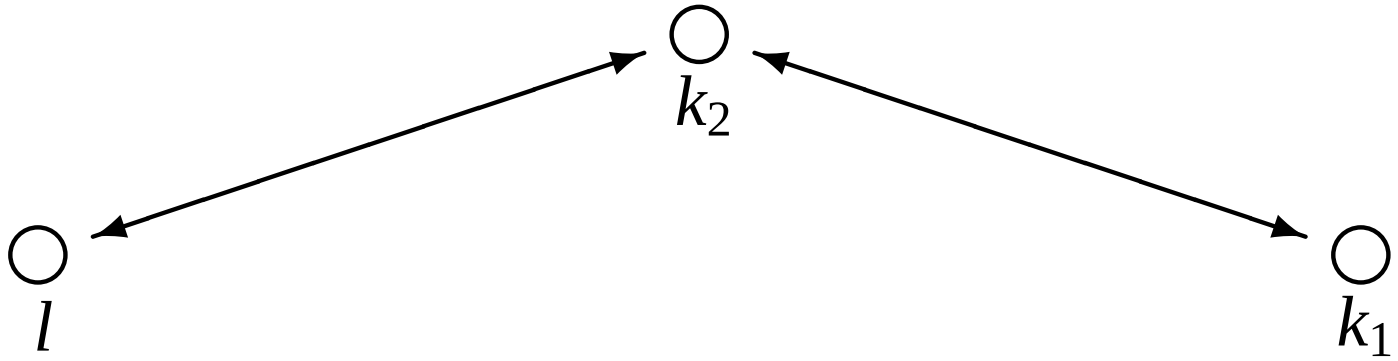


serverLnk

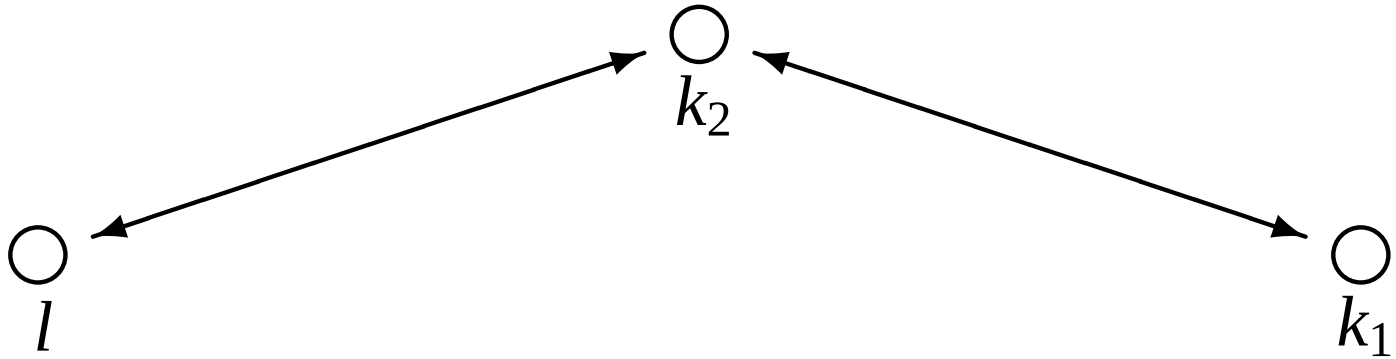
The Underlying Network



The Underlying Network

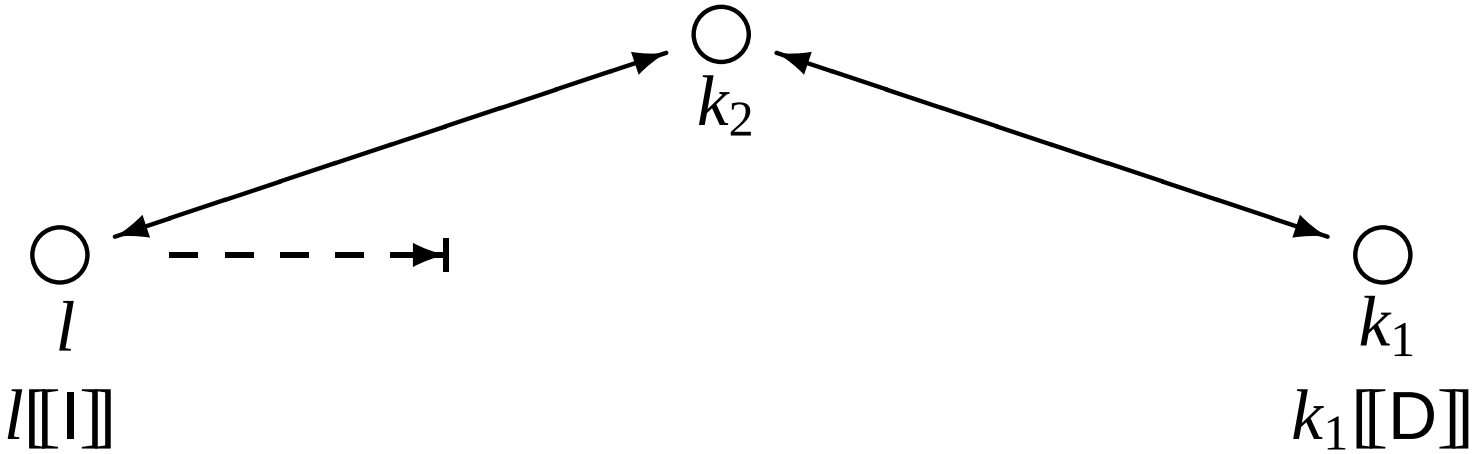


The Underlying Network

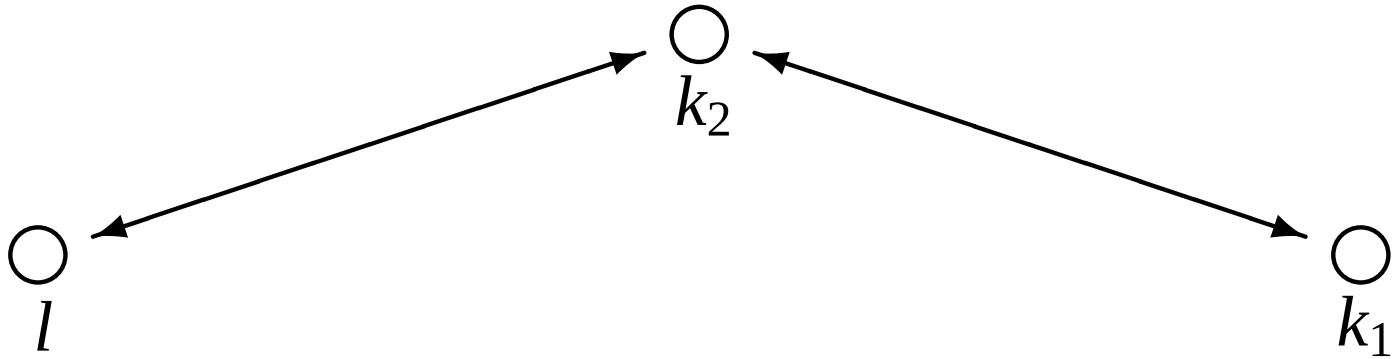


serverLoc

The Underlying Network

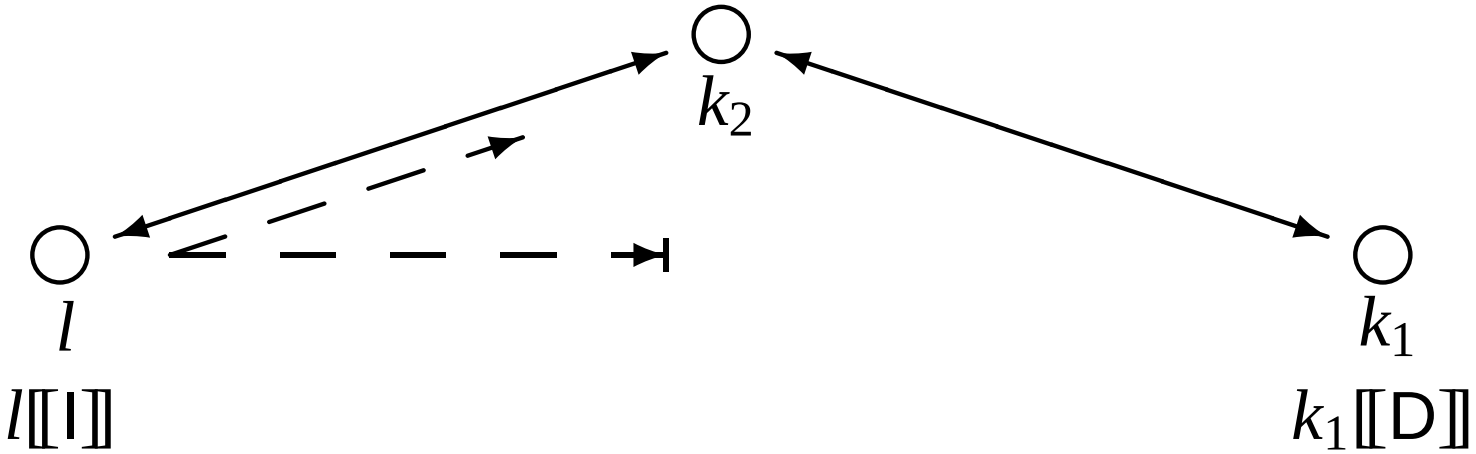


The Underlying Network



serverLnk

The Underlying Network



$D\pi F$: $D\pi$ with failure

$$l[[P]] \mid (\nu k)(k[[Q]] \mid l[[R]])$$

$D\pi F$: $D\pi$ with failure

$$\Delta \triangleright l[[P]] \mid (\nu k : \mathtt{T})(k[[Q]] \mid l[[R]])$$

DπF: Dπ with failure

Types

$$\mathtt{T},\mathtt{U} \;::=\; \mathtt{loc}[\mathtt{A}, \{l_1, .., l_n\}] \;\mid \mathtt{ch}$$

$$\mathtt{A} \;::=\; \mathtt{a} \mid \mathtt{d}$$

Processes

$$P, Q \;::=\; \ldots$$

$\mid$	$(\nu\, n : \mathtt{T})P$	(*typed scoping*)
$\mid$	kill	(*kill location*)
$\mid$	break l	(*break link*)
$\mid$	ping $l.P$ else Q	(*ping*)

DπF: Dπ with failure

Systems

$$N, M \quad ::= \quad \ldots \quad | (\nu\, n : \mathtt{T}) N$$

We work at the level of **Configurations**

$$\Delta \triangleright M$$

where Δ is a network representation

$D\pi F$: $D\pi$ with failure

(r-comm)

$$\frac{\Delta \vdash l : \textbf{alive}}{\Delta \triangleright l[\![a!\langle n\rangle.P]\!] \mid l[\![a?(x).Q]\!] \longrightarrow \Delta \triangleright l[\![P]\!] \mid l[\![Q\{^{n}/_{x}\}]\!]}$$

(r-go)

$$\frac{\Delta \vdash l : \textbf{alive} \qquad \Delta \vdash k : \textbf{alive} \qquad \Delta \vdash l \leftrightarrow k}{\Delta \triangleright l[\![\text{go } k.P]\!] \longrightarrow \Delta \triangleright k[\![P]\!]}$$

$D\pi F$: $D\pi$ with failure

(r-kill)

$$\frac{\Delta \vdash l : \mathbf{alive}}{\Delta \triangleright l[\![\mathsf{kill}]\!] \longrightarrow \Delta - l \triangleright l[\![\mathbf{stop}]\!]}$$

(r-brk)

$$\frac{\Delta \vdash l : \mathbf{alive} \qquad \Delta \vdash l \leftrightarrow k}{\Delta \triangleright l[\![\mathsf{break}\ k]\!] \longrightarrow \Delta - l \leftrightarrow k \triangleright l[\![\mathbf{stop}]\!]}$$

$D\pi F$: $D\pi$ with failure

(r-ping)

$$\frac{\Delta \vdash l : \mathbf{alive} \qquad \Delta \vdash k : \mathbf{alive} \qquad \Delta \vdash l \leftrightarrow k}{\Delta \triangleright l[\![\text{ping } k.P \text{ else } Q]\!] \longrightarrow \Delta \triangleright l[\![P]\!]}$$

(r-nping1)

$$\frac{\Delta \vdash l : \mathbf{alive} \qquad \Delta \vdash k : \mathbf{dead}}{\Delta \triangleright l[\![\text{ping } k.P \text{ else } Q]\!] \longrightarrow \Delta \triangleright l[\![Q]\!]}$$

(r-nping2)

$$\frac{\Delta \vdash l : \mathbf{alive} \qquad \Delta \vdash l \nleftrightarrow k}{\Delta \triangleright l[\![\text{ping } k.P \text{ else } Q]\!] \longrightarrow \Delta \triangleright l[\![Q]\!]}$$

Motivating Example (2)

$$\mathsf{M} \Leftarrow (\nu\, k)(l[\![a!\langle k\rangle]\!] \mid k[\![b!\langle\rangle]\!])$$

$$\mathsf{N} \Leftarrow (\nu\, k)(l[\![a!\langle k\rangle]\!] \mid k[\![c!\langle\rangle]\!])$$

Motivating Example (2)

$$\mathsf{M} \Leftarrow (\nu\, k)(l[\![a!\langle k\rangle]\!] \mid k[\![b!\langle\rangle]\!])$$

$$\mathsf{N} \Leftarrow (\nu\, k)(l[\![a!\langle k\rangle]\!] \mid k[\![c!\langle\rangle]\!])$$

$$\mathsf{M} \not\cong \mathsf{N}$$

Motivating Example (2)

$$\mathsf{M} \Leftarrow (\nu\, k)(l[\![a!\langle k\rangle]\!] \mid k[\![b!\langle\rangle]\!])$$

$$\mathsf{N} \Leftarrow (\nu\, k)(l[\![a!\langle k\rangle]\!] \mid k[\![c!\langle\rangle]\!])$$

$$\mathsf{M} \not\cong \mathsf{N}$$

$$\mathsf{O} \Leftarrow l[\![a?(x).\mathsf{go}\ x.b?().\mathsf{go}\ l.\mathsf{ok}!\langle\rangle]\!]$$

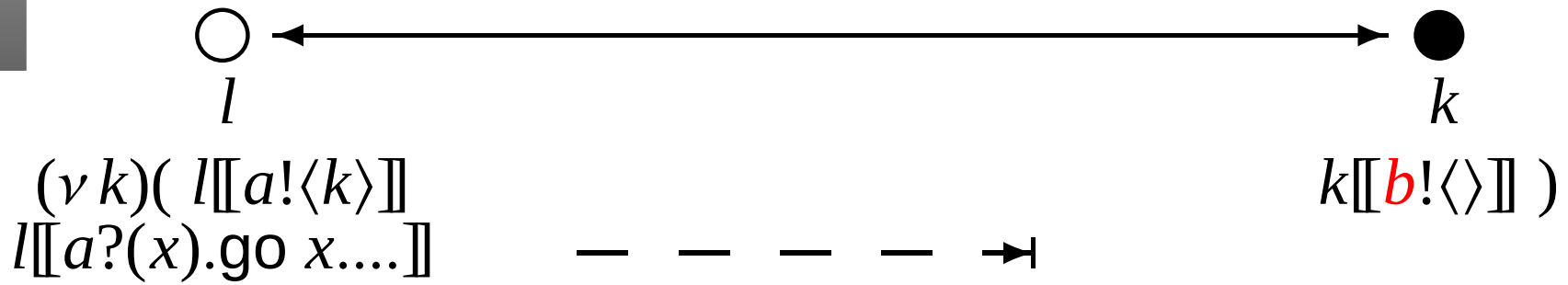
$$\mathsf{M}|\mathsf{O} \Downarrow_{\mathsf{ok}@l}$$

$$\mathsf{N}|\mathsf{O} \not\Downarrow_{\mathsf{ok}@l}$$

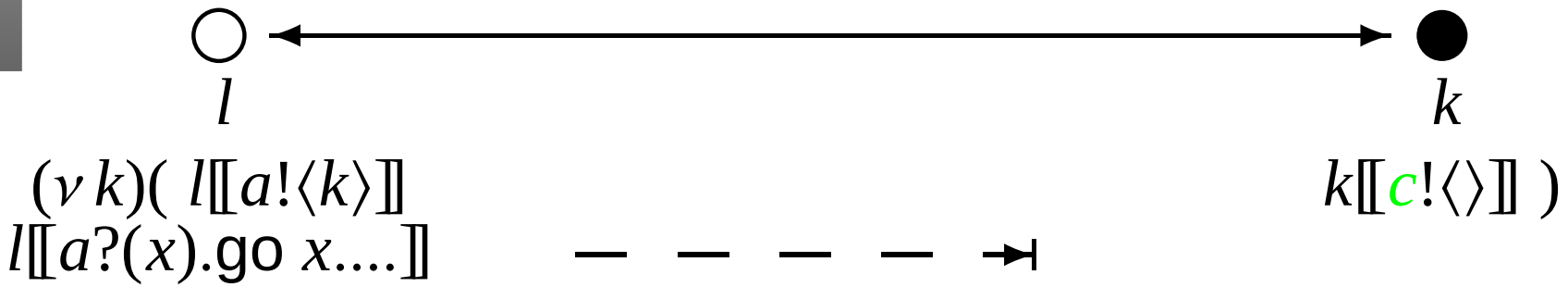
Motivating Example (2)

Motivating Example (2)

Motivating Example (2)

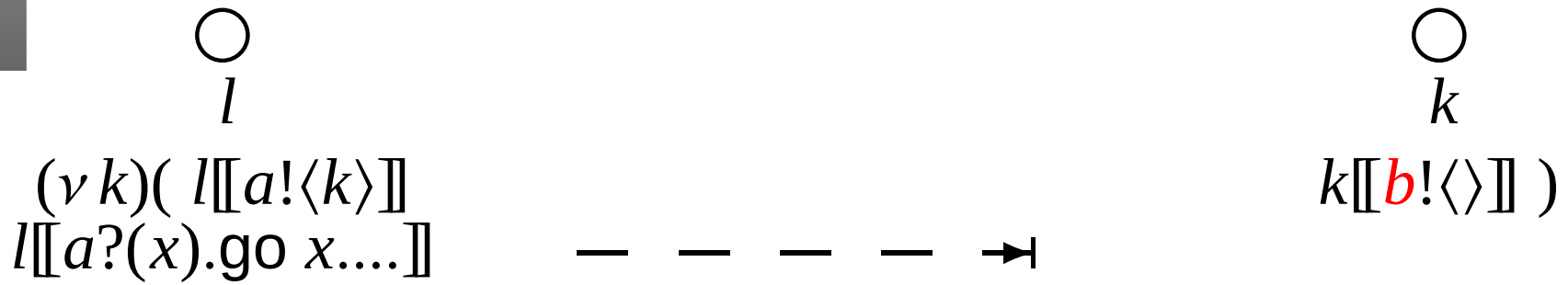


Motivating Example (2)



Motivating Example (2)

Motivating Example (2)



Motivating Example (2)

Motivating Example (2)

For $\Delta \vdash l$: **alive**

$$\mathsf{M} \Leftarrow (\nu\, k : \mathsf{T})(l[\![a!\langle k\rangle]\!] \mid k[\![b!\langle\rangle]\!])$$

$$\mathsf{N} \Leftarrow (\nu\, k : \mathsf{T})(l[\![a!\langle k\rangle]\!] \mid k[\![c!\langle\rangle]\!])$$

$$\mathsf{O} \Leftarrow l[\![a?(x).\mathrm{go}\ x.b?().\mathrm{go}\ l.\mathrm{ok}!\langle\rangle]\!]$$

$$\mathsf{M}|\mathsf{O}\Downarrow\!\!\!\!\!\!/\ _{\mathrm{ok}@l}$$
$$\mathsf{N}|\mathsf{O}\Downarrow\!\!\!\!\!\!/\ _{\mathrm{ok}@l}$$

$\Delta \triangleright \mathsf{M} \cong \Delta \triangleright \mathsf{N}$ for certain types T

Technical Development

- We define a reduction barbed congruence, $\cong$ for systems running over the same network, Δ.

Technical Development

- We define a reduction barbed congruence, $\cong$ for systems running over the same network, Δ.
- We define an alternative network representation, Σ, that encodes, in addition to the state of the network, the observer partial view. (Δ is a simple case of Σ)

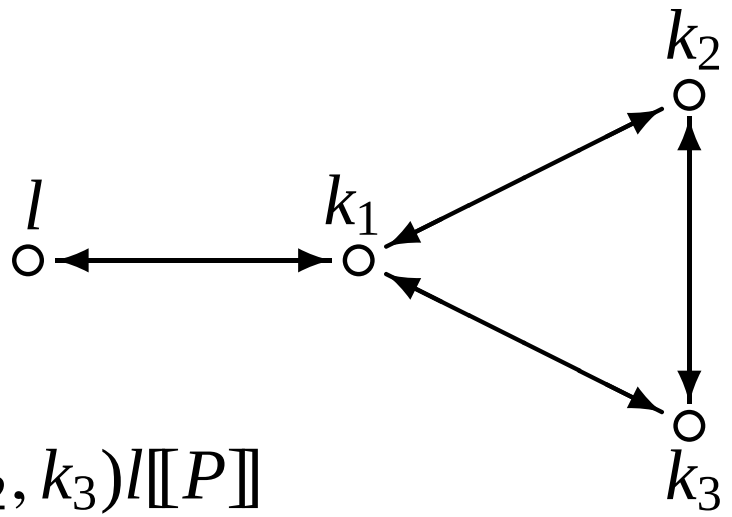
Technical Development

- We define a reduction barbed congruence, $\cong$ for systems running over the same network, Δ.
- We define an alternative network representation, Σ, that encodes, in addition to the state of the network, the observer partial view. (Δ is a simple case of Σ)
- We define a labelled transition system for configurations subject to Σ: labels record changes in observable part of Σ.

Technical Development

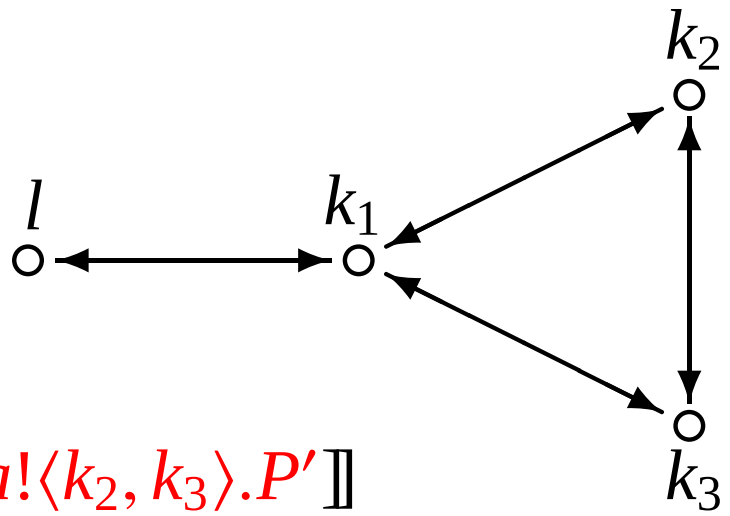
- We define a reduction barbed congruence, $\cong$ for systems running over the same network, Δ.
- We define an alternative network representation, Σ, that encodes, in addition to the state of the network, the observer partial view. (Δ is a simple case of Σ)
- We define a labelled transition system for configurations subject to Σ: labels record changes in observable part of Σ.
- We define a bisimulation $\approx$ and prove that it coincides with $\cong$.

Application: Network Discovery



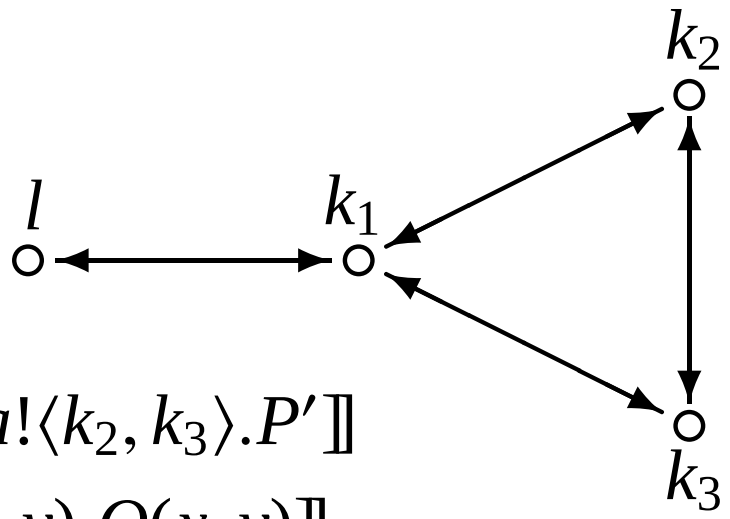
$$(\nu k_1, k_2, k_3) l[\![P]\!]$$

Application: Network Discovery



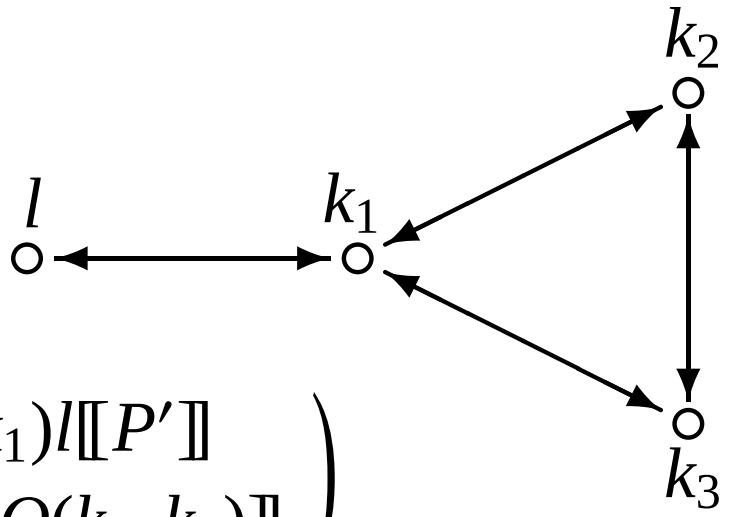
$$(\nu k_1, k_2, k_3) l[\![a!\langle k_2, k_3\rangle.P']\!]$$

Application: Network Discovery



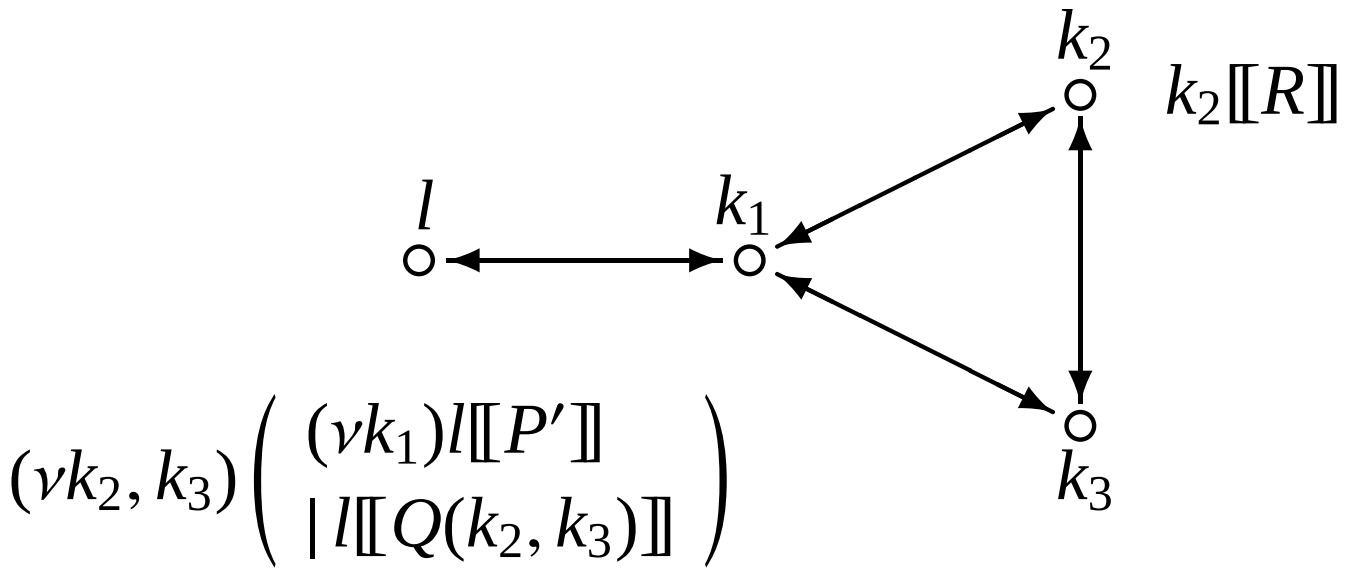
$$(\nu k_1, k_2, k_3) l[\![a!\langle k_2, k_3\rangle.P']\!]$$
$$\mid l[\![a?(x,y).Q(x,y)]\!]$$

Application: Network Discovery

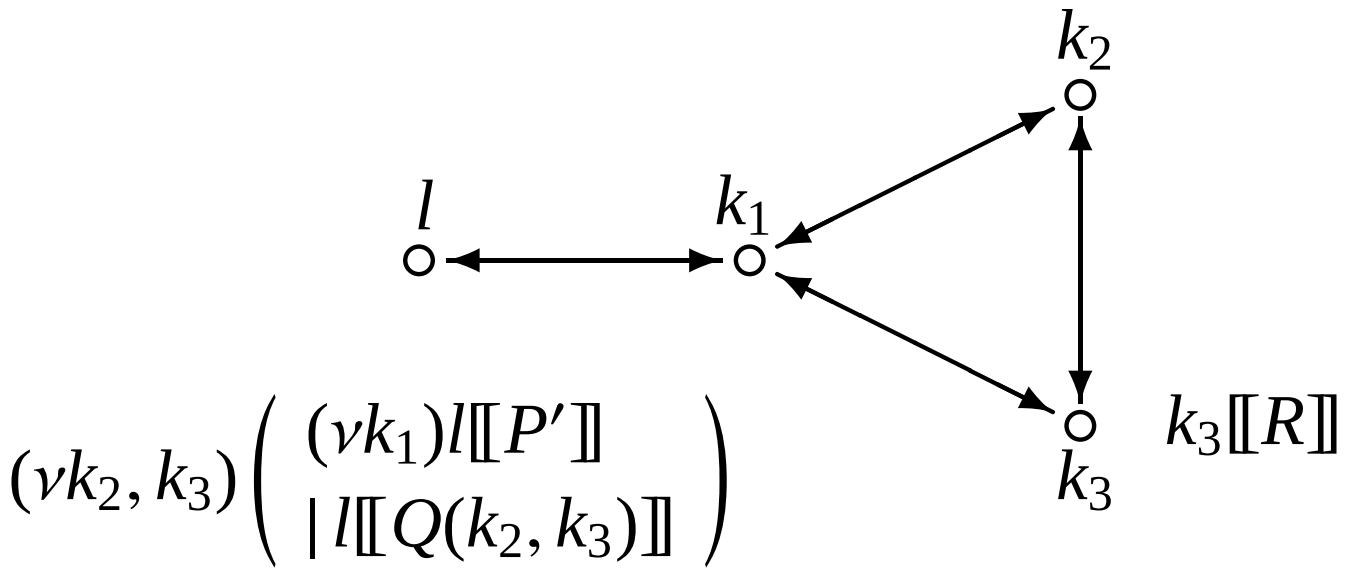


$$(\nu k_2, k_3)\left(\begin{array}{l}(\nu k_1) l[\![P']\!] \\ \mid l[\![Q(k_2, k_3)]\!]\end{array}\right)$$

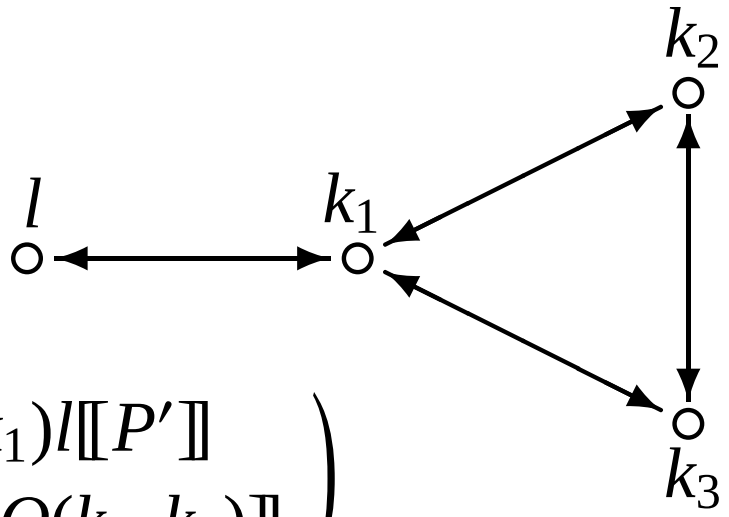
Application: Network Discovery



Application: Network Discovery

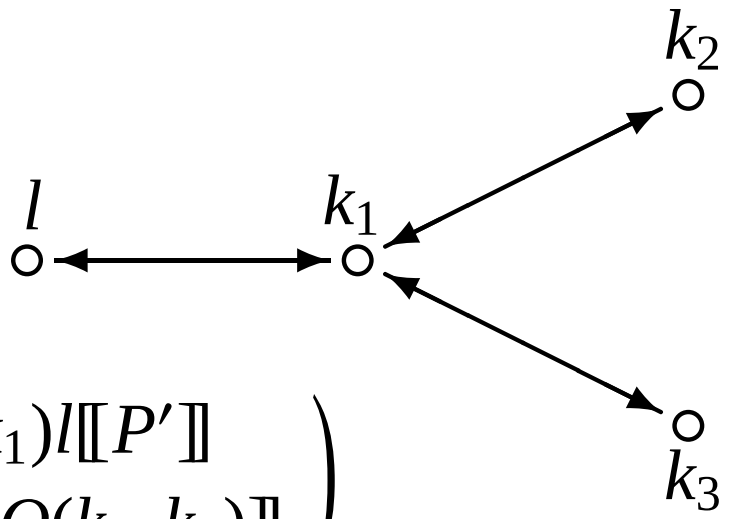


Application: Network Discovery



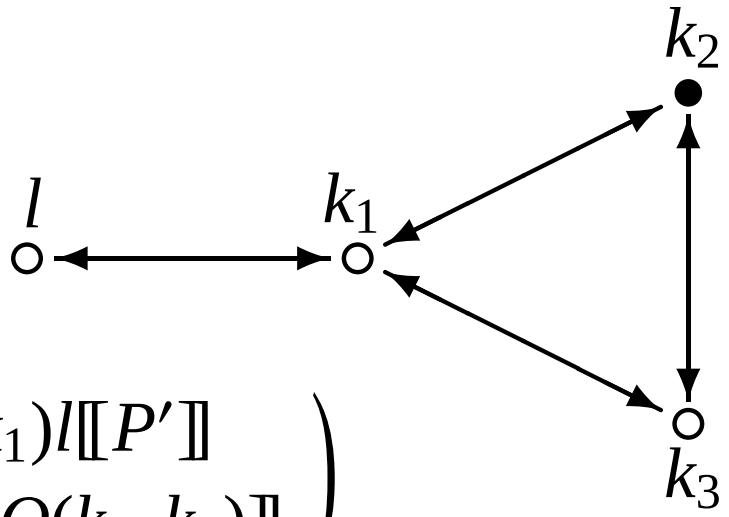
$$(\nu k_2, k_3)\left(\begin{array}{l}(\nu k_1) l[\![P']\!] \\ \mid l[\![Q(k_2, k_3)]\!]\end{array}\right)$$

Application: Network Discovery



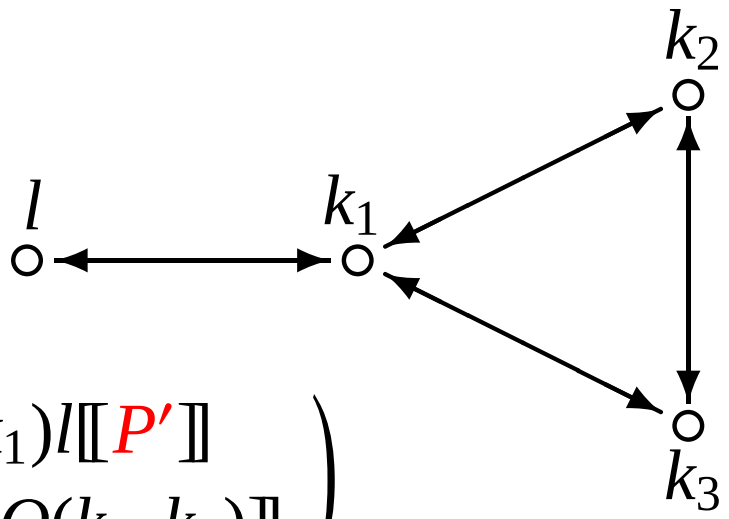
$$(\nu k_2, k_3)\left(\begin{array}{l}(\nu k_1) l[\![P']\!] \\ \mid l[\![Q(k_2, k_3)]\!]\end{array}\right)$$

Application: Network Discovery



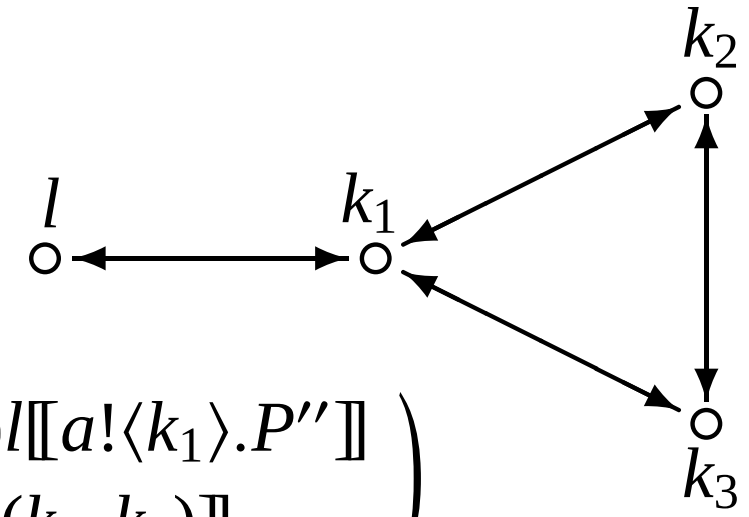
$$(\nu k_2, k_3)\left(\begin{array}{l}(\nu k_1) l[\![P']\!] \\ \mid l[\![Q(k_2, k_3)]\!]\end{array}\right)$$

Application: Network Discovery



$$(\nu k_2, k_3)\left(\begin{array}{l}(\nu k_1) l[\![P']\!] \\ \mid l[\![Q(k_2, k_3)]\!]\end{array}\right)$$

Application: Network Discovery



$$(\nu k_2, k_3)\left(\begin{array}{l}(\nu k_1) l[\![a!\langle k_1\rangle.P'']\!] \\ \mid l[\![Q(k_2, k_3)]\!]\end{array}\right)$$

References

- **Location and Link Failure in a Distributed π-calculus** ,
  Adrian Francalanza, Matthew Hennessy, University of Sussex Technical Report, 2005:01, January 2005.
- **Failure and Fault Tolerance in a Distributed π-calculus** ,
  Adrian Francalanza, University of Sussex D.Phil. Thesis,*(submitted May 2005).*