Collaboration and Cooperation in Team Automata

Maurice H. ter Beek[1]

Tuesday 14 December 1999

⇒ joint work with Clarence A. Ellis[2],
Jetty Kleijn[1], and Grzegorz Rozenberg[1,2]

[1]LIACS, UL

[2]Department of Computer Science, University of Colorado

Motivation

Computer Supported Cooperative Work is:

How do groups communicate, *collaborate*, and *cooperate* when working together?

How can technology (groupware) assist?

CSCW needs:

Conceptual level:
define concepts and terminology consistently

Architectural level:
describe and compare groupware systems

Ellis introduced concept of *team automata* for modelling both levels informally in

"Team Automata for Groupware Systems"

(GROUP'97 ACM Conference on Supporting Group Work)

Motivation - continued

Ter Beek, Ellis, Kleijn, and Rozenberg formalized and elaborated this concept in

"Synchronizations in Team Automata for Groupware Systems"

(Submitted, 1999)

Model provides:

Conceptual level:
unambiguous formal framework for
defining basic groupware notions such as
communication, *collaboration*, and *cooperation*

Architectural level:
use of team automata as building blocks in design of groupware systems

Component Automaton

$$\mathcal{C} = (Q, (\Sigma_{inp}, \Sigma_{out}, \Sigma_{int}), \delta, I)$$

Q set of *states*

$\Sigma = \Sigma_{inp} \cup \Sigma_{out} \cup \Sigma_{int}$ *alphabet* (partition)

$\delta \subseteq Q \times \Sigma \times Q$ *transition relation* $\qquad q \xrightarrow{a} q'$

$I \subseteq Q$ set of *initial states*

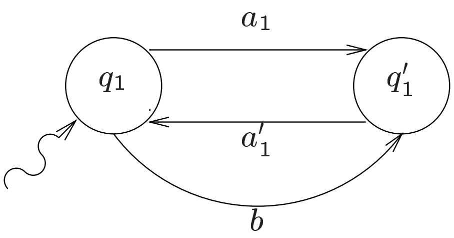
$\left.\begin{array}{l}\Sigma_{inp} \text{ input actions} \\ \Sigma_{out} \text{ output actions}\end{array}\right\}$ Σ_{ext} externally observable

Σ_{int} *internal actions* $\qquad$ cannot be observed

$\delta_a = \{(q, q') \mid (q, a, q') \in \delta\}$ set of a-*transitions*
(i.e. all arcs labelled with a)

Example Component Automata

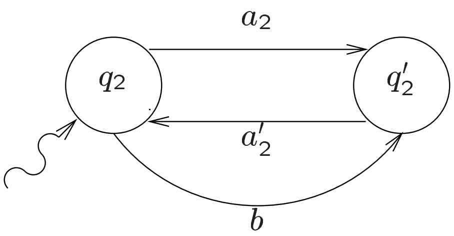
$\mathcal{C}_1$:



$a_1, a'_1 \in \Sigma_{1,int}$

$b \in \Sigma_{1,\boldsymbol{out}}$

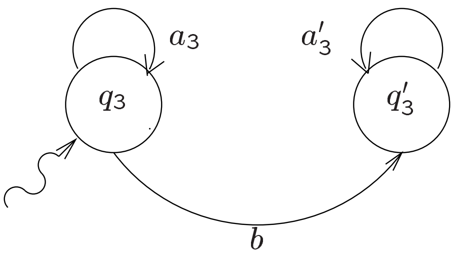
$\mathcal{C}_2$:



$a_2, a'_2 \in \Sigma_{2,int}$

$b \in \Sigma_{2,\boldsymbol{out}}$

$\mathcal{C}_3$:



$a_3, a'_3 \in \Sigma_{3,int}$

$b \in \Sigma_{3,\boldsymbol{inp}}$

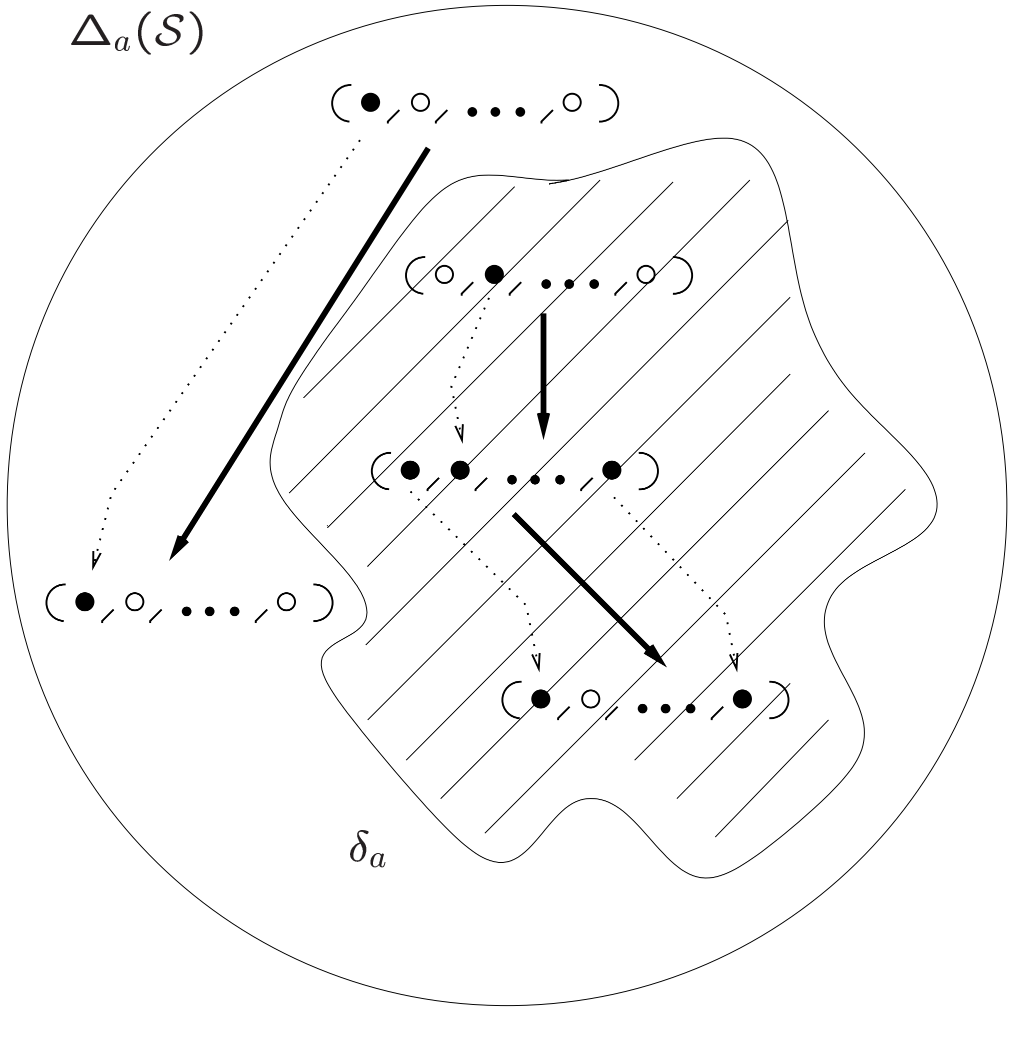
$\{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$ is a composable system

Composable System

A set of component automata $\mathcal{S} = \{\mathcal{C}_i \mid i \in \mathcal{I}\}$ is a *composable system* if, for all $i \in \mathcal{I}$,

$$\Sigma_{i,int} \cap \bigcup_{j \in \mathcal{I} \setminus \{i\}} \Sigma_j = \varnothing$$

Transition Space



Complete Transition Space

The *complete transition space* of $a \in \Sigma = \bigcup_{i \in \mathcal{I}} (\Sigma_{i,inp} \cup \Sigma_{i,out} \cup \Sigma_{i,int})$ in $\mathcal{S} = \{\mathcal{C}_i \mid i \in \mathcal{I}\}$ is

$$
\begin{aligned}
\Delta_a(\mathcal{S}) = \{(q, q') \in \textstyle\prod_{i \in \mathcal{I}} Q_i \times \prod_{i \in \mathcal{I}} Q_i \mid \\
\underline{\exists j \in \mathcal{I} :\ (\mathsf{proj}_j(q), a, \mathsf{proj}_j(q')) \in \delta_j} \ \wedge \\
\forall i \in \mathcal{I} :\ (\mathsf{proj}_i(q), a, \mathsf{proj}_i(q')) \in \delta_i \ \vee \\
\mathsf{proj}_i(q) = \mathsf{proj}_i(q')\}
\end{aligned}
$$

In every team transition at least one component acts <u>according</u> to its transition relation

All other components either join in this team transition or are idle

The choices of team transition relations δ_a, $a \in \Sigma$, define a specific team automaton

Team Automaton

$\mathcal{T} = (\prod_{i\in\mathcal{I}} Q_i, (\Sigma_{inp}, \Sigma_{out}, \Sigma_{int}), \delta, \prod_{i\in\mathcal{I}} I_i)$

is a *team automaton over* $\mathcal{S} = \{\mathcal{C}_i \mid i \in \mathcal{I}\}$ if

$$\left.\begin{array}{l} \Sigma_{int} = \bigcup_{i\in\mathcal{I}} \Sigma_{i,int} \\[1em] \Sigma_{out} = \bigcup_{i\in\mathcal{I}} \Sigma_{i,out} \\[1em] \Sigma_{inp} = (\bigcup_{i\in\mathcal{I}} \Sigma_{i,inp}) \setminus \Sigma_{out} \end{array}\right\} = \Sigma$$

$\delta \subseteq \prod_{i\in\mathcal{I}} Q_i \times \Sigma \times \prod_{i\in\mathcal{I}} Q_i$ such that $\forall a \in \Sigma$

$$\delta_a \subseteq \Delta_a(\mathcal{S})$$

and moreover $\delta_a = \Delta_a(\mathcal{S})$ if $a \in \Sigma_{int}$

Lemma: Every team automaton is a component automaton

Example Team Automaton over $\{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$

$$\mathcal{T} = \Big(\prod_{i \in \{1,2,3\}} Q_i, (\Sigma_{inp}, \Sigma_{out}, \Sigma_{int}), \delta, \{(q_1, q_2, q_3)\} \Big)$$

$\Sigma_{inp} = \varnothing$ (!)

$\boldsymbol{\Sigma_{out} = \{b\}}$

$\Sigma_{int} = \{a_1, a'_1, a_2, a'_2, a_3, a'_3\}$

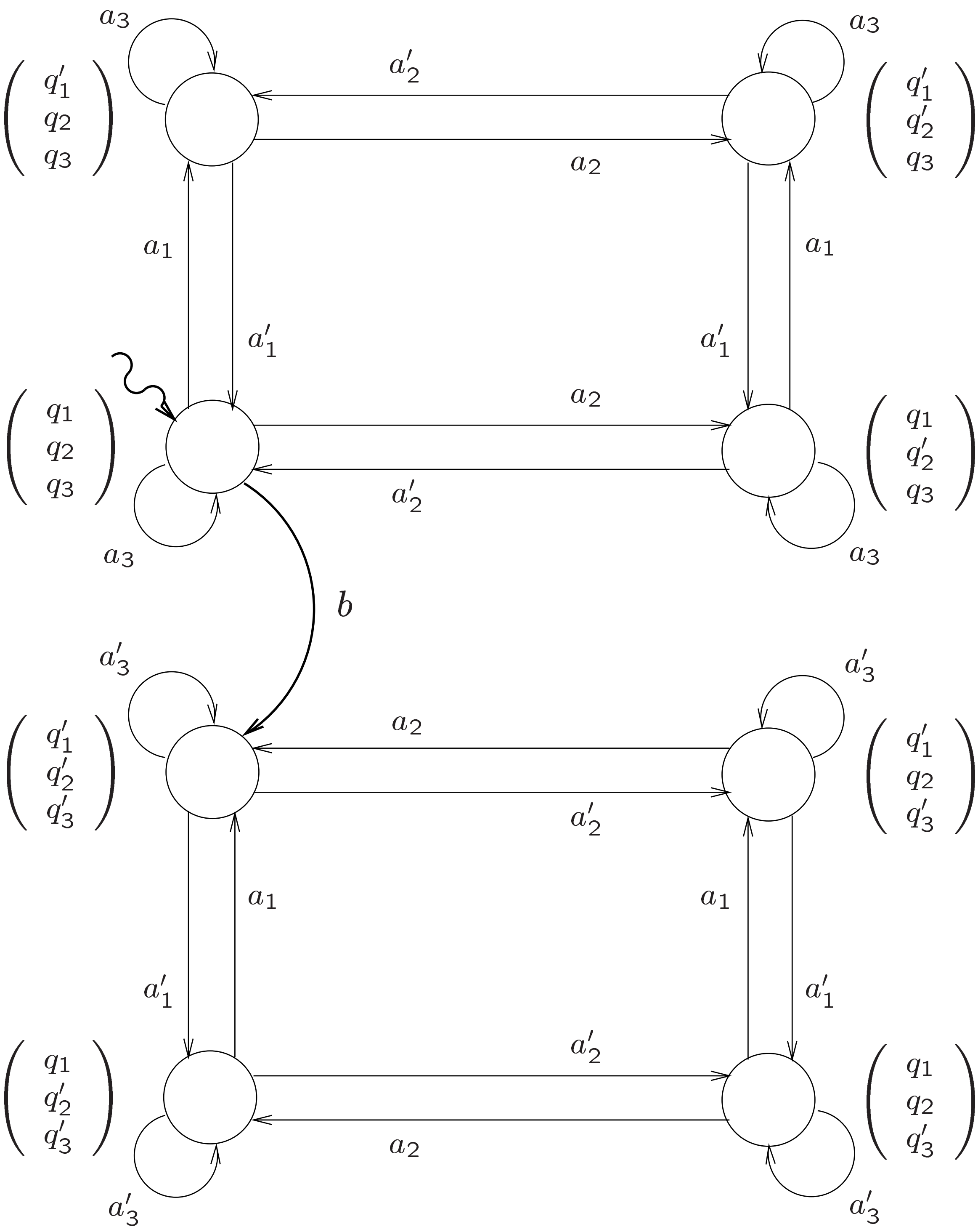
δ is defined by

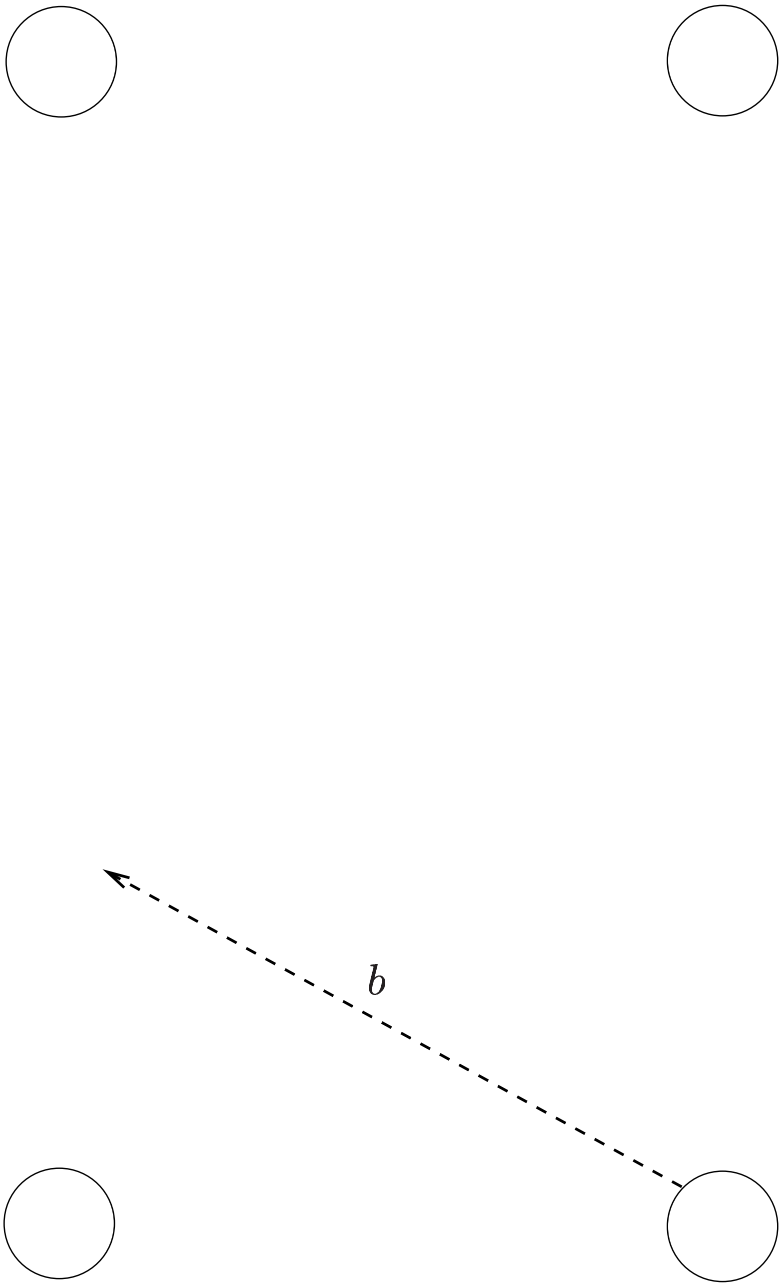
$\delta_a = \Delta_a(\{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\})$ for all $a \in \Sigma_{int}$

$\delta_b = \{((q_1, q_2, q_3), (q'_1, q'_2, q'_3))\}$

$$\mathcal{T}' : \quad \delta'_b = \{((q_1, q_2, q_3), (q'_1, q'_2, q'_3)), \\ ((q_1, q_2, q'_3), (q'_1, q'_2, q'_3))\}$$

Example Team Automaton $\mathcal{T}$




b

Subteam

The subteam of $\mathcal{T}$ determined by $J \subseteq \mathcal{I}$ is

$$SUB_J(\mathcal{T}) = (\prod_{j \in J} Q_j, (\Sigma_{J,inp}, \Sigma_{J,out}, \Sigma_{J,int}), \delta_J, \prod_{j \in J} I_j)$$

$$\left.\begin{array}{l} \Sigma_{J,int} = \bigcup_{j \in J} \Sigma_{j,int} \\ \\ \Sigma_{J,out} = \bigcup_{j \in J} \Sigma_{j,out} \\ \\ \Sigma_{J,inp} = (\bigcup_{j \in J} \Sigma_{j,inp}) \setminus \Sigma_{J,out} \end{array}\right\} = \Sigma_J$$

δ_J is defined by, for all $a \in \Sigma_J$,

$$(\delta_J)_a = \{(\mathrm{proj}_J(q), \mathrm{proj}_J(q')) \mid (q, q') \in \delta_a)\} \cap \Delta_a(\{\mathcal{C}_j \mid j \in J\})$$

Lemma: $SUB_J(\mathcal{T})$ is a team automaton over $\{\mathcal{C}_j \mid j \in J\}$

Key Notions So Far

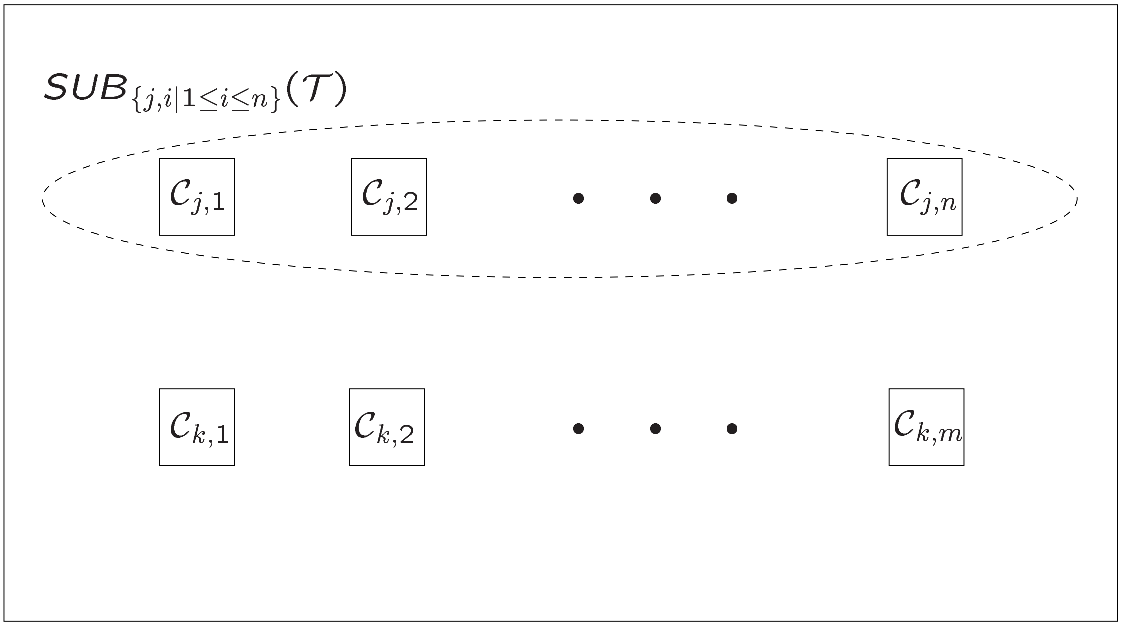
Component automaton

Composable system (of component automata)

Team automaton (over a composable system)

Subteam (of a team automaton)

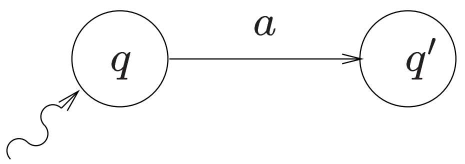
$\mathcal{T}$ over $\mathcal{S} = \{\mathcal{C}_{j,i}, \mathcal{C}_{k,l} \mid 1 \leq i \leq n,\ 1 \leq l \leq m\}$



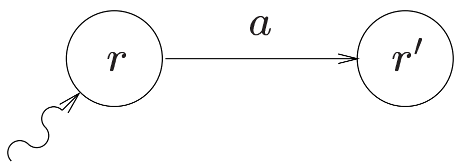
Synchronizations

Action a is *ai* (*action-indispensable*) in team automaton $\mathcal{T}$ if all components which have a as an action participate in every a-transition of $\mathcal{T}$

$\mathcal{C}_1$:



$\mathcal{C}_2$:

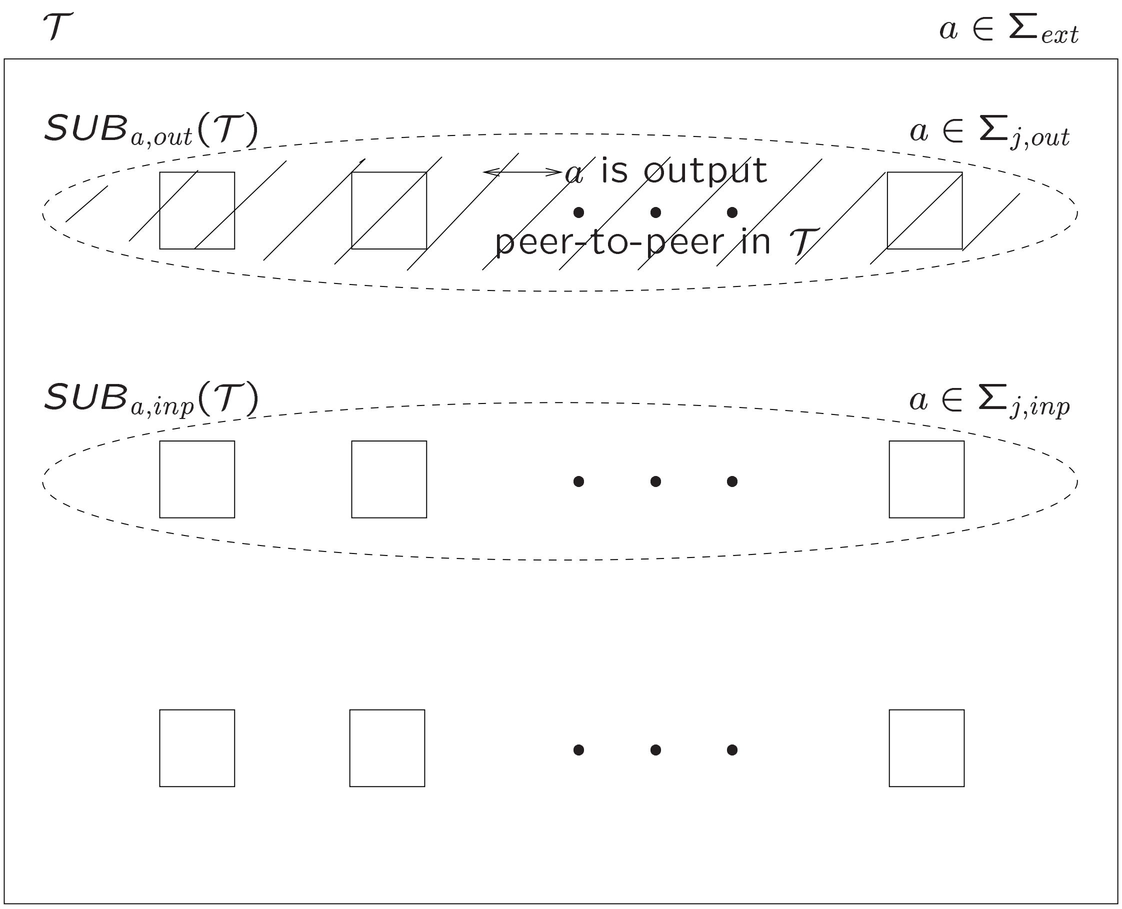


in $\mathcal{T}$:	is action a *ai* ?
$\delta = \{((q,r),a,(q',r'))\}$	yes
$\delta = \{((q,r),a,(q',r))\}$	no
$\delta = \{((q,r'),a,(q',r'))\}$	no
$\delta = \{((q,r),a,(q',r)),((q,r),a,(q',r'))\}$	no

Collaboration

between peers (input or output)

Action a is *output peer-to-peer* in team automaton $\mathcal{T}$ if a is *ai* in $SUB_{a,out}(\mathcal{T})$



Input peer-to-peer analogous

Cooperation

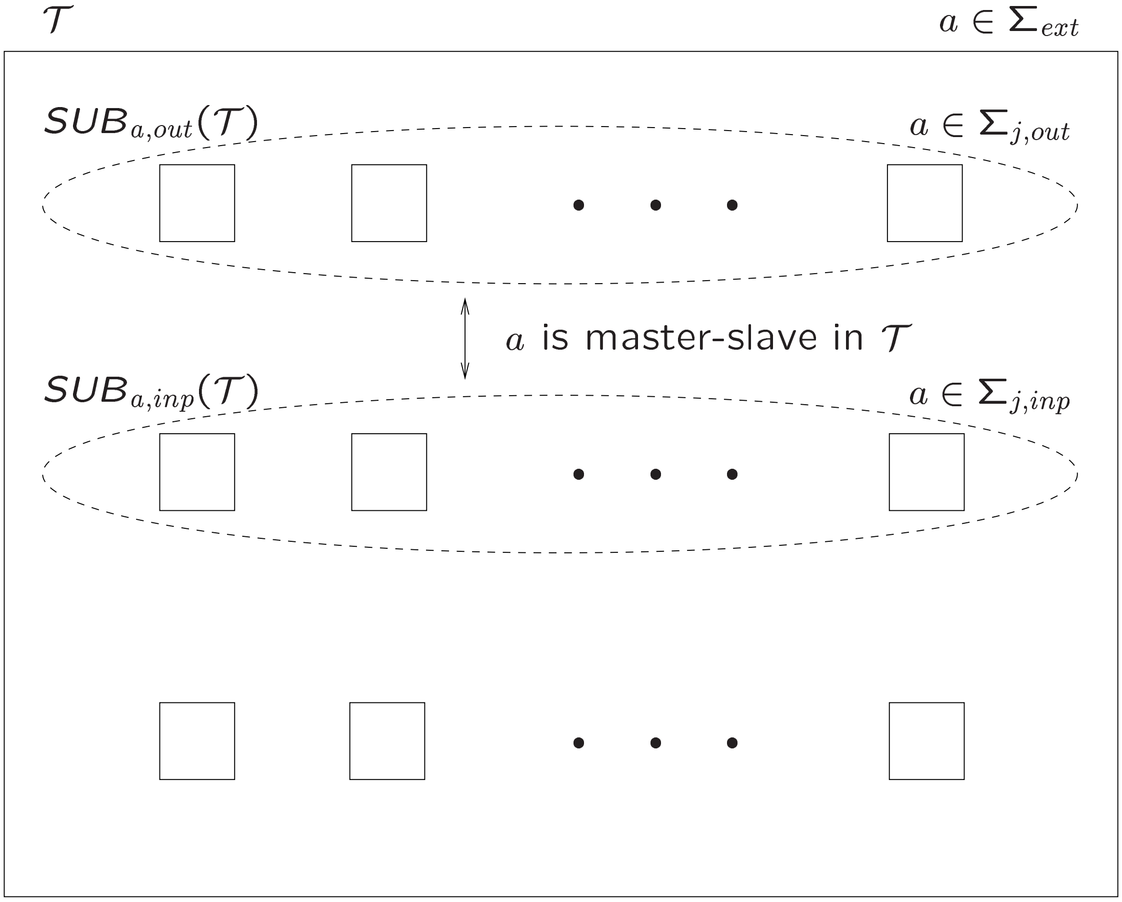
between masters (output) and slaves (input)

Action a is *master-slave* in team automaton $\mathcal{T}$ if

$SUB_{a,out}$ participates in every a-transition of $\mathcal{T}$
(i.e. slaves never proceed on their own)

and

$SUB_{a,inp}$ participates in every a-transition of $\mathcal{T}$
(i.e. slaves have to obey the master)



Example Team Automaton $\mathcal{T}$

b output and input peer-to-peer in $\mathcal{T}$

b master-slave in $\mathcal{T}$ since

<u>only</u>
b-transition of $\mathcal{T}$:

$$\begin{tikzcd}[column sep=large]
(\underline{q_1, q_2}, q_3) \arrow[r, "b"] \arrow[r, bend right=60, "b"'] & (\underline{q_1', q_2'}, q_3')
\end{tikzcd}$$

$SUB_{\{1,2\}}(\mathcal{T})$

and

<u>only</u>
b-transition of $\mathcal{T}$:

$$\begin{tikzcd}[column sep=large]
(q_1, q_2, \underline{q_3}) \arrow[r, "b"] \arrow[r, bend right=60, "b"'] & (q_1', q_2', \underline{q_3'})
\end{tikzcd}$$

$b \in \Sigma_{3,inp}$

$SUB_{\{3\}}(\mathcal{T})$

Example Team Automaton $\mathcal{T}'$

b output and input peer-to-peer in $\mathcal{T}'$

b **not** master-slave in $\mathcal{T}'$ since

also in
$$\begin{tikzcd} (\underline{q_1, q_2}, q_3') \arrow[r, "b"] \arrow[r, bend right=60, "b"'] & (\underline{q_1', q_2'}, q_3') \end{tikzcd}$$

$SUB_{\{1,2\}}(\mathcal{T}')$

but
$$\begin{tikzcd} (q_1, q_2, \underline{q_3'}) \arrow[r, "b"] \arrow[r, bend right=60, "\cancel{b}"'] & (q_1', q_2', \underline{q_3'}) \end{tikzcd}$$

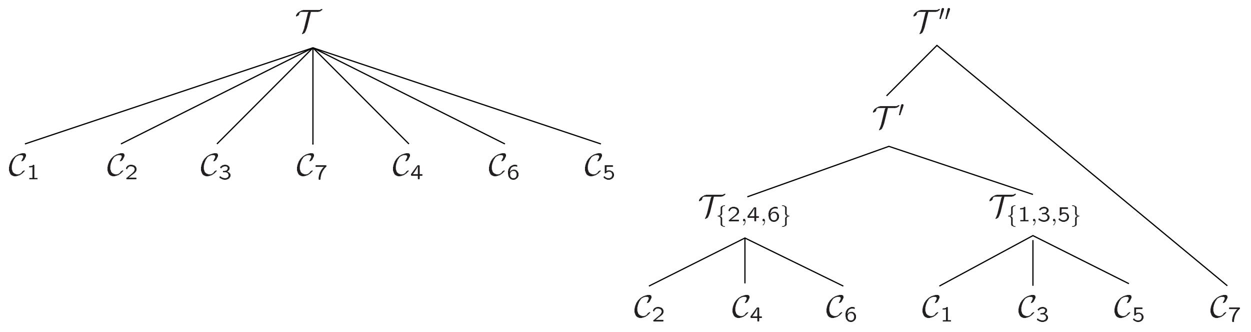
$b \in \Sigma_{3,inp}$

$\cancel{SUB_{\{3\}}(\mathcal{T}')}$

Research Topics

- Teams over Teams
- The Problem with Loops
- Vector Team Automata
- Behaviour of Team Automata
- Maximal Team Automata
- Team Automata vs. Petri Nets
- Team Automata vs. I/O Automata
- Languages of Team Automata

Teams over Teams



$\mathcal{T}$ team over $\{\mathcal{C}_i \mid i \in \{1, 2, \ldots, 7\}\}$
$\mathcal{T}''$ iterated team over $\{\mathcal{C}_i \mid i \in \{1, 2, \ldots, 7\}\}$

<u>Theorem</u>: after "reordering" elements of $\mathcal{T}''$:

state space $\mathcal{T}''$ = state space $\mathcal{T}$

alphabets $\mathcal{T}''$ = alphabets $\mathcal{T}$ (incl. partition)

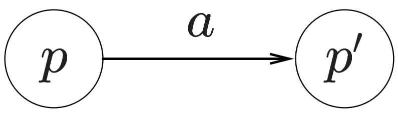
initial state space $\mathcal{T}''$ = initial state space $\mathcal{T}$

(potentially the same transition relation)

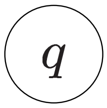
In fact, also $\mathcal{T}''$ can be considered as a team over $\{\mathcal{C}_i \mid i \in \{1, 2, \ldots, 7\}\}$

The Problem with Loops

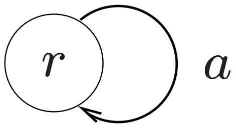
$\mathcal{T}$: $(p, q, r) \xrightarrow{a} (p', q, r)$

$\mathcal{C}_1$: 

$\mathcal{C}_1$ executed a

$\mathcal{C}_2$: 

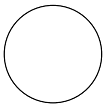
$\mathcal{C}_2$ cannot execute a

$\mathcal{C}_3$: 

did $\mathcal{C}_3$ execute a
or was $\mathcal{C}_3$ idle?

Vector Team Automata

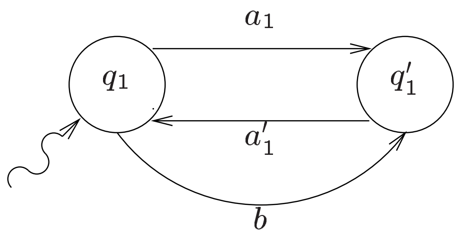
$$\begin{pmatrix} a \\ \lambda \\ \end{pmatrix}$$



<u>$\mathcal{C}_3$ executed a</u> !

Behaviour of Automata

$\mathcal{C}$:



$a_1, a'_1 \in \Sigma_{1,int}$

$b \in \Sigma_{1,out}$

$w = q_1 b q'_1 a'_1 q_1 b q'_1 \in \mathbf{C}_{\mathcal{C}_1}$ (w *computation* of $\mathcal{C}_1$)

input record of w is empty
output record of w is bb
internal record of w is a'_1

$bb \in \mathbf{B}_{\mathcal{C}_1}^{\Sigma_{out}}$ (bb in *output behaviour* of $\mathcal{C}_1$)
$\{\lambda\} = \mathbf{B}_{\mathcal{C}_1}^{\Sigma_{inp}}$ (no *input behaviour* of $\mathcal{C}_1$)

<u>Lemma</u>: (1) $\forall i \in \mathcal{I} : w \in \mathbf{C}_{\mathcal{T}} \Rightarrow \pi_{\mathcal{C}_i}(w) \in \mathbf{C}_{\mathcal{C}_i}$

(2) if a is *ai* $\forall a$ in alphabet Θ, then
$\forall i \in \mathcal{I} : w \in \mathbf{B}_{\mathcal{T}}^{\Theta} \Rightarrow \mathrm{pres}_{\Sigma_i}(w) \in \mathbf{B}_{\mathcal{C}_i}^{\Theta}$

Maximal Team Automata

$\mathcal{T}$ is <u>the</u> *ai-maximal team automaton* over $\mathcal{S} = \{\mathcal{C}_i \mid i \in \mathcal{I}\}$ if, for all $a \in \Sigma$,

$$\delta_a = \{(q, q') \in \Delta_a(\mathcal{S}) \mid \forall i \in \mathcal{I} : \\ a \in \Sigma_i \Rightarrow (\mathrm{proj}_i(q), \mathrm{proj}_i(q')) \in \delta_{i,a}\}$$

Thus δ_a is the unique maximal transition relation in $\Delta_a(\mathcal{S})$ by which a is *ai*

Hence every action of the *ai*-maximal team automaton (over $\mathcal{S}$) is *ai*

<u>Lemma</u>: if $\mathcal{T}$ is *ai*-maximal team automaton and w sequence of actions of $\mathcal{T}$, then $(\forall i \in \mathcal{I} : \mathrm{pres}_{\Sigma_i}(w) \in \mathbf{B}^{\Sigma}_{\mathcal{C}_i}) \Rightarrow w \in \mathbf{B}^{\Sigma}_{\mathcal{T}}$