Team Automata for Spatial Access Control

Maurice H. ter Beek[1]

Wednesday 10 May 2000

⇒ 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

Access Control

- security and information access control form vital component of any system

- groupware requires:

  - more refined access rights
    e.g. scroll file synchronously edited by group in real time

  - control of access meta-rights
    e.g. grant/revoke a team member the right to grant/revoke file access to a non-team member (i.e. delegation)

Spatial Access Control

SPACE metaphor (Bullock, 1998)

$\Downarrow$

access control governed by spaces (rooms)
in which subjects and objects reside

and

the ability of a subject to traverse space
in order to get closer to an object

SPACE Access Model

- basic tenet: a fundamental component of any collaborative environment is the environment itself (i.e. the space)

- hide explicit technical security mechanisms from end users through natural spatial makeup of the environment

- users can use their knowledge of environment to understand the implicit security policies

- avoids users tackling technical concepts such as so-called access matrices

- helps avoiding misunderstandings

Virtual Reality

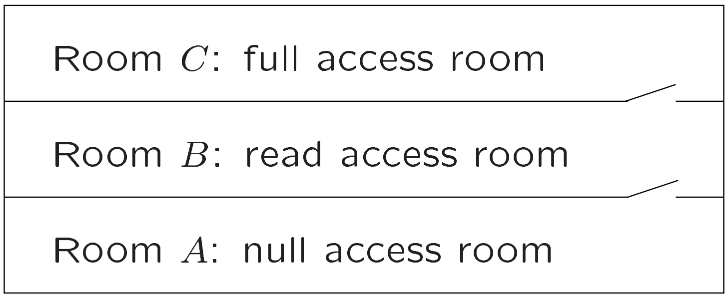
- user traverses from room to room
  by using keyboard keys or mouse

- *authorization* check at boundaries (doors)
  between spaces (rooms)

- if access is OK, then user can enter and
  use resources of newly entered room

- here abstract from *authentication*

Example



- satisfies end user friendly properties:
  simple, relatively natural and unobtrusive,
  understandable by non-computer people,
  and elegant

- modelling by team automata adds:
  precision, mathematical rigour,
  and analytic capabilities

Component Automaton

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

Q set of *states*

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

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

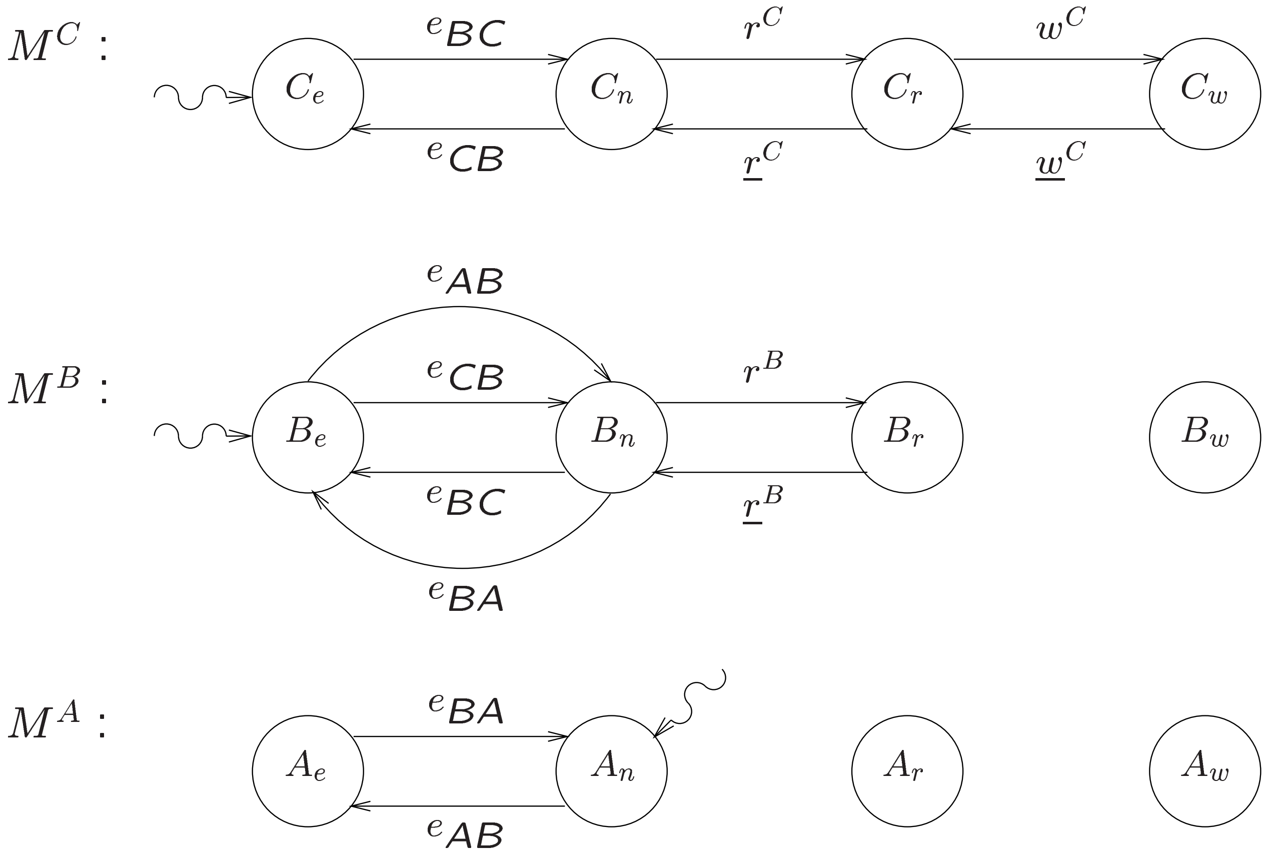
$I \subseteq Q$ set of *initial states*

$\left.\begin{array}{l}\Sigma_{inp} \text{ \textit{input actions}} \\ \Sigma_{out} \text{ \textit{output actions}}\end{array}\right\} \Sigma_{ext}$ 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)

Component Automata M^C, M^B, M^A: Rooms C, B and A



r^C, $\underline{r}^C$, w^C, $\underline{w}^C$, r^B, and $\underline{r}^B$ **internal** actions

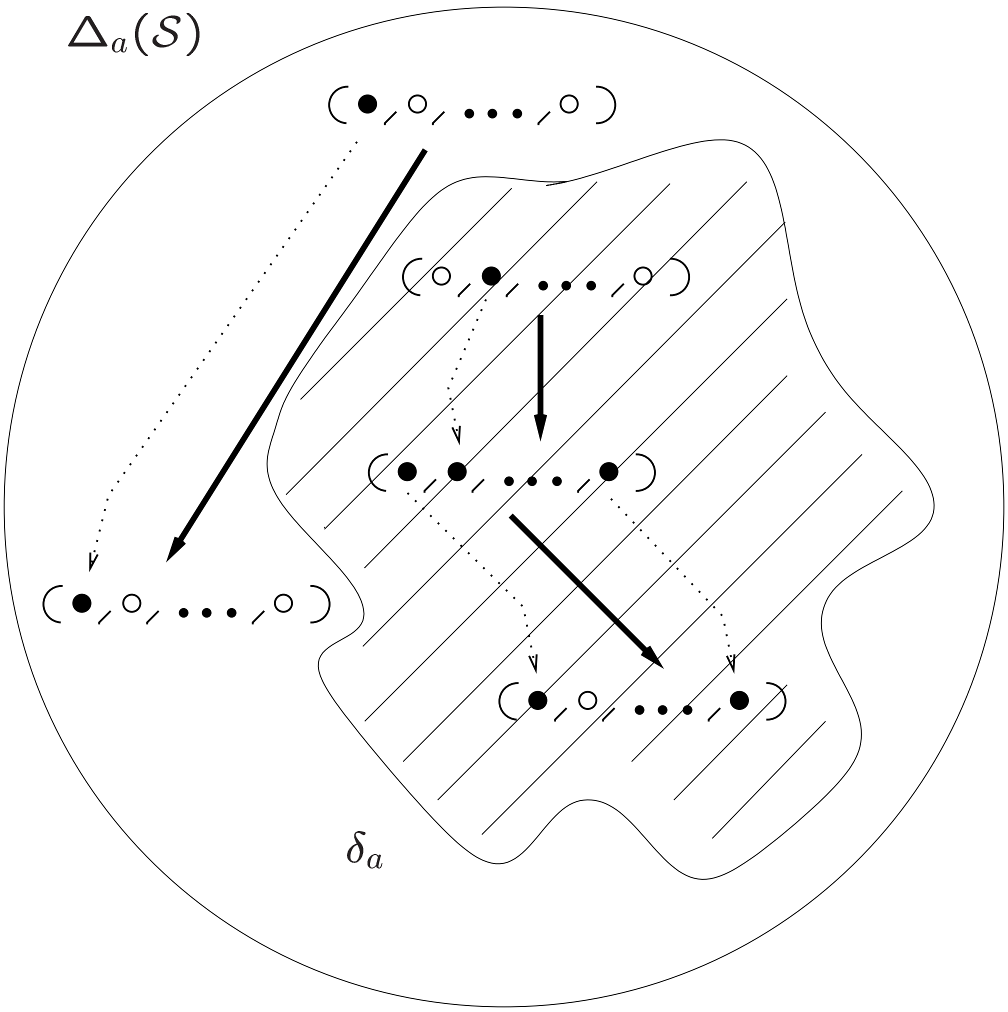
e_{BC}, e_{CB}, e_{AB}, and e_{BA} **external** actions

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

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

$$\Delta_a(\mathcal{S}) = \{(q, q') \in \textstyle\prod_{i \in \mathcal{I}} Q_i \times \prod_{i \in \mathcal{I}} Q_i \mid$$

$$\underline{\exists j \in \mathcal{I} : \ (\mathsf{proj}_j(q), a, \mathsf{proj}_j(q')) \in \delta_j} \ \wedge$$

$$\forall i \in \mathcal{I} : \ (\mathsf{proj}_i(q), a, \mathsf{proj}_i(q')) \in \delta_i \ \vee$$
$$\mathsf{proj}_i(q) = \mathsf{proj}_i(q')\}$$

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

$\Rightarrow$ 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 $\forall a \in \Sigma$

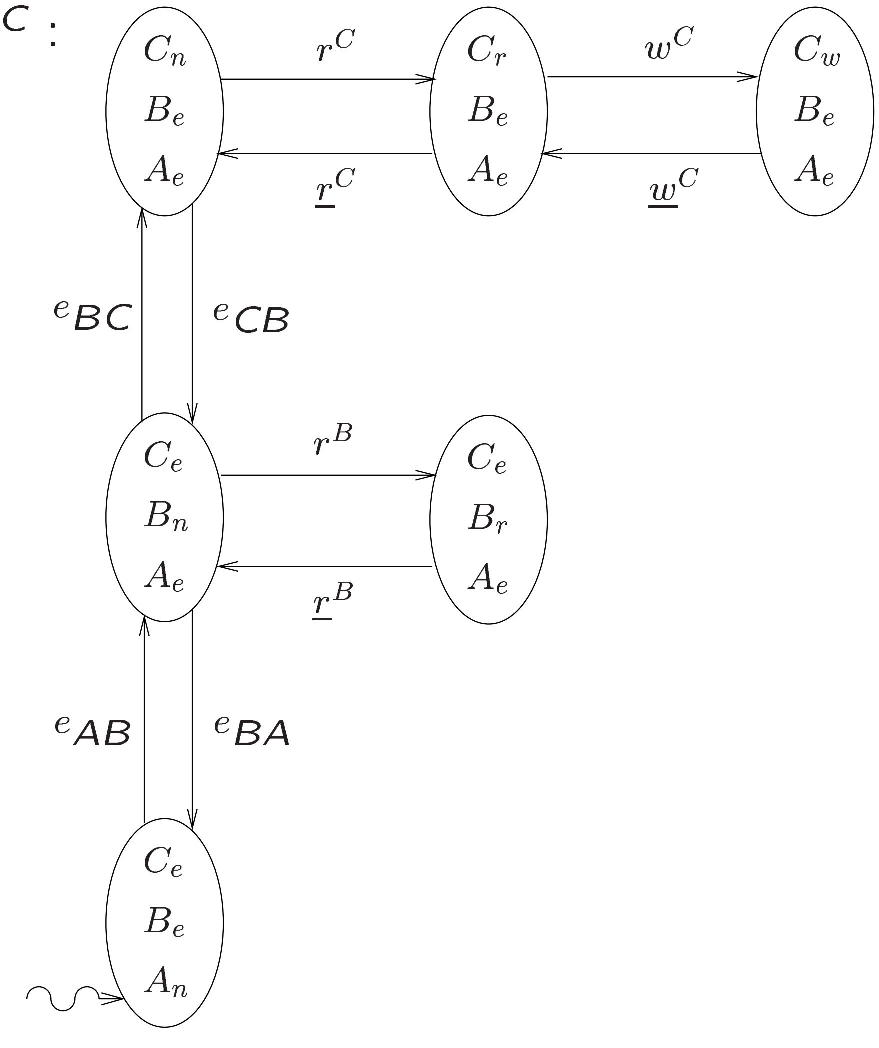
$$\delta_a \subseteq \Delta_a(\mathcal{S})$$

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

Lemma: every team automaton is a component automaton

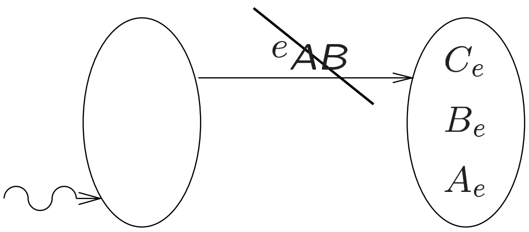
Example Team Automaton T^{ABC} over $\{M^C, M^B, M^A\}$

T^{ABC} :



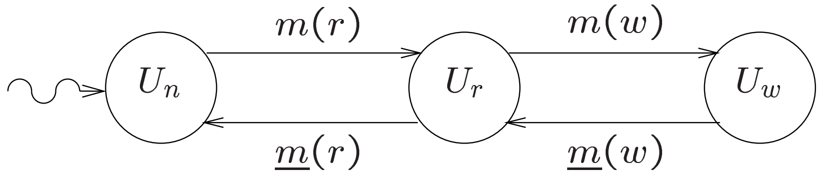
peer-to-peer vs. *master-slave* synchronizations

T^{ABC} :



Authorization Added to T^{ABC}

M^U :



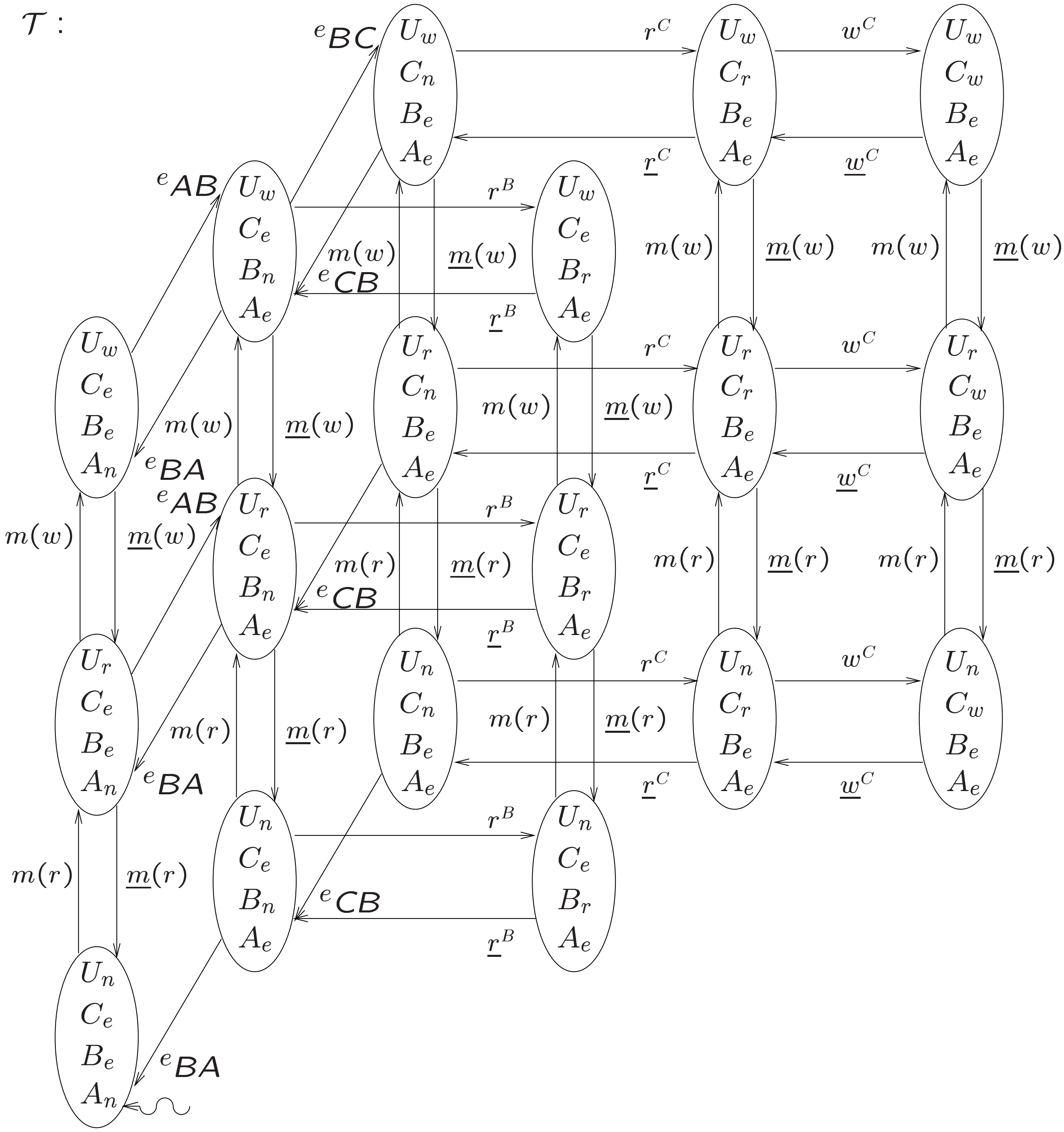
$$\left.\begin{array}{ll} m(r) & \text{read access granted} \\ \underline{m}(r) & \text{read access revoked} \\ m(w) & \text{write access granted} \\ \underline{m}(w) & \text{write access revoked} \end{array}\right\} \begin{array}{l} \textbf{input} \\ \text{actions} \end{array}$$

$\{T^{ABC}, M^U\}$ composable system

$\Rightarrow$ compose team automaton over $\{T^{ABC}, M^U\}$
by choosing a transition relation!

Example Team Automaton $\mathcal{T}$ over $\{T^{ABC}, M^U\}$

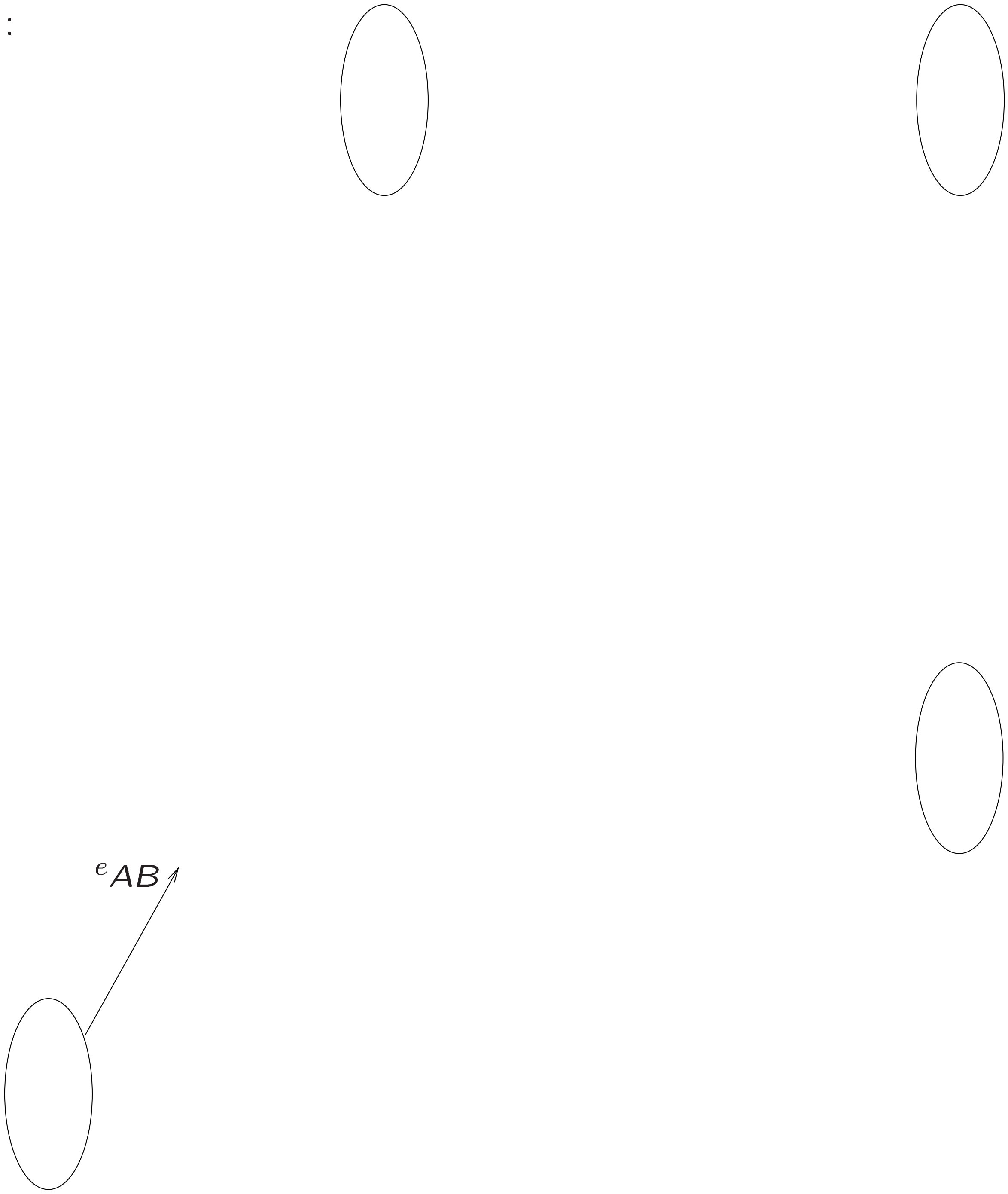
$\mathcal{T}$:



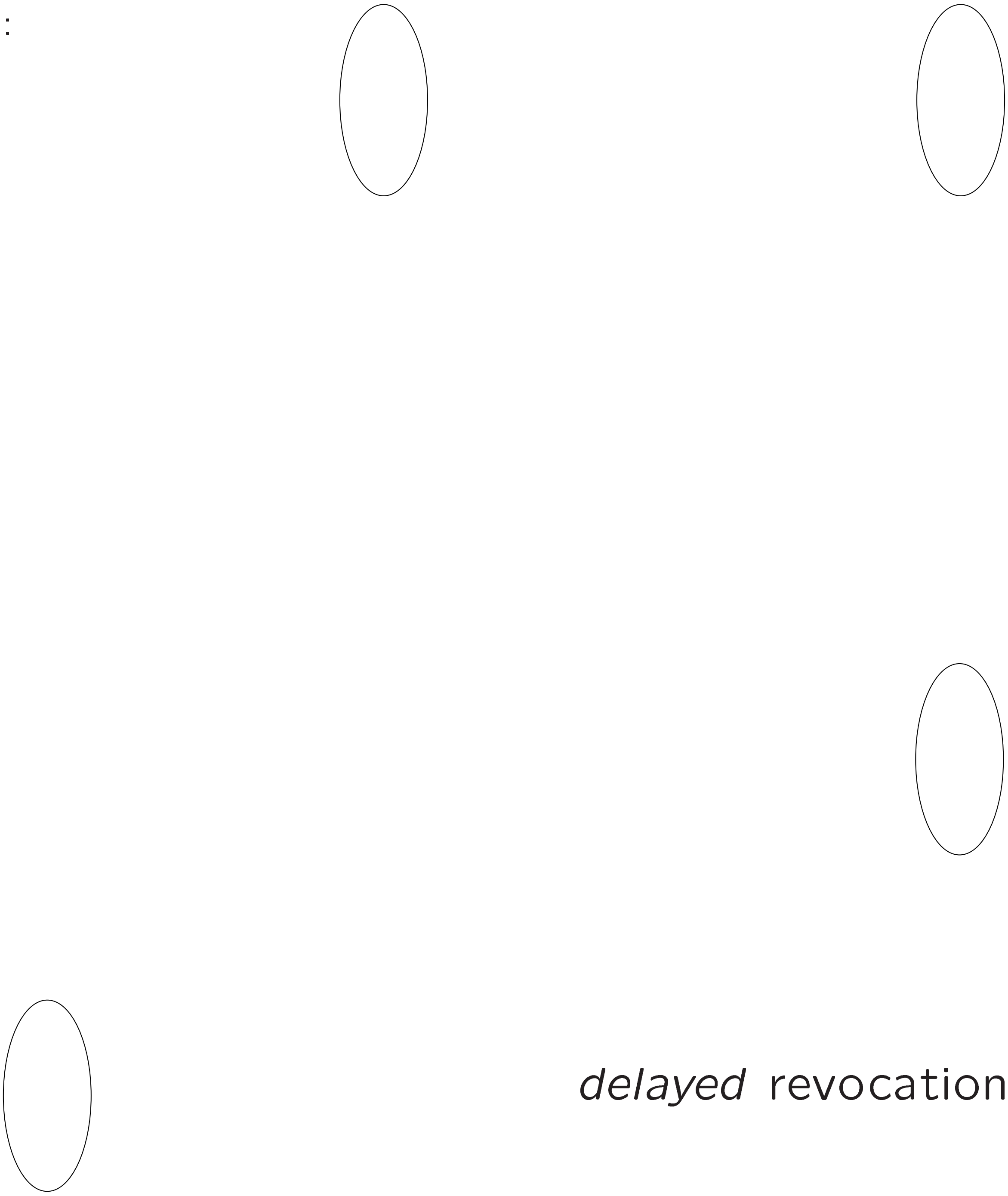
$\mathcal{T}$:

authorization

$\mathcal{T}$:

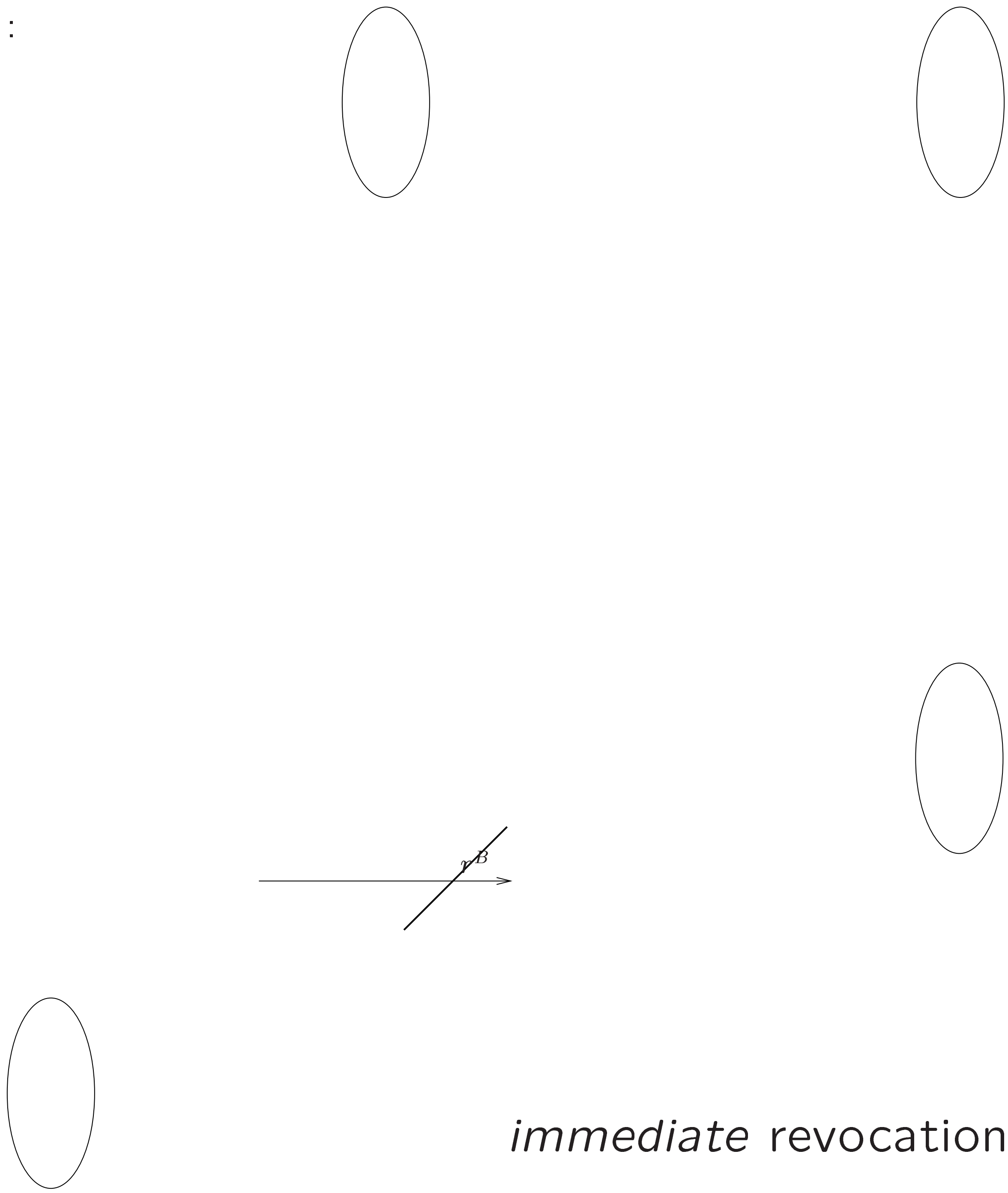


$\mathcal{T}$:



delayed revocation

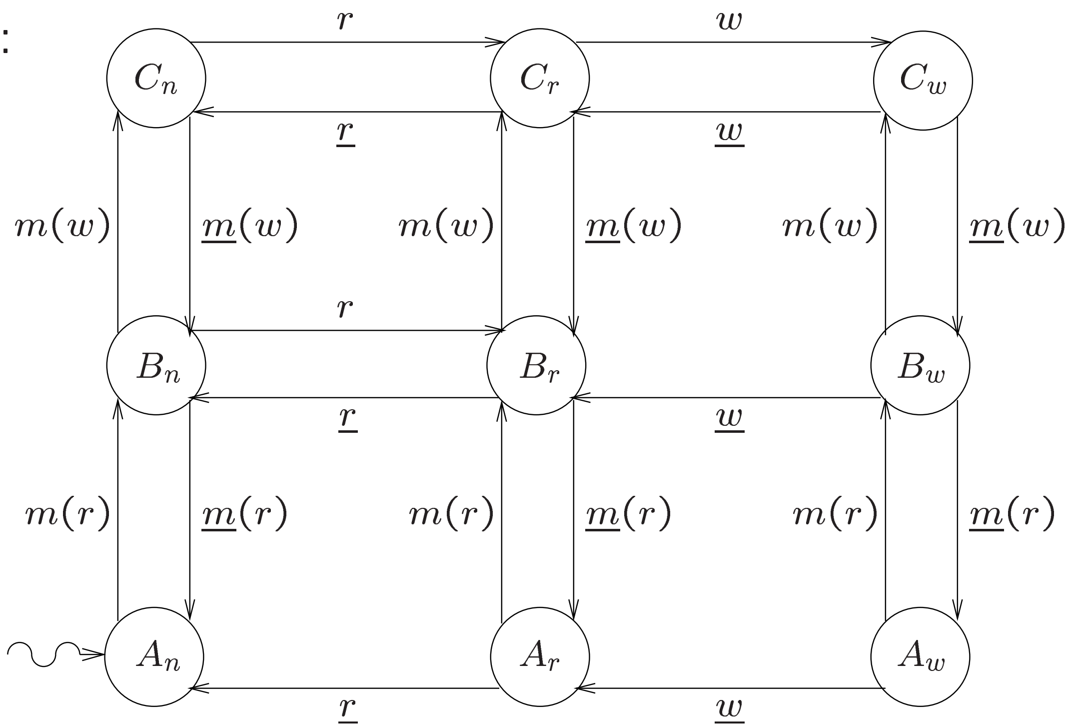
$\mathcal{T}$:



immediate revocation

Automaton M^0: The Access Building

M^0 :



r, $\underline{r}$, w, and $\underline{w}$ **internal** actions

$m(r)$, $\underline{m}(r)$, $m(w)$, and $\underline{m}(w)$ **input** actions

irregular states

Meta Access Control

- Rights to grant and revoke can themselves be granted and revoked
  e.g. journal refereeing process:

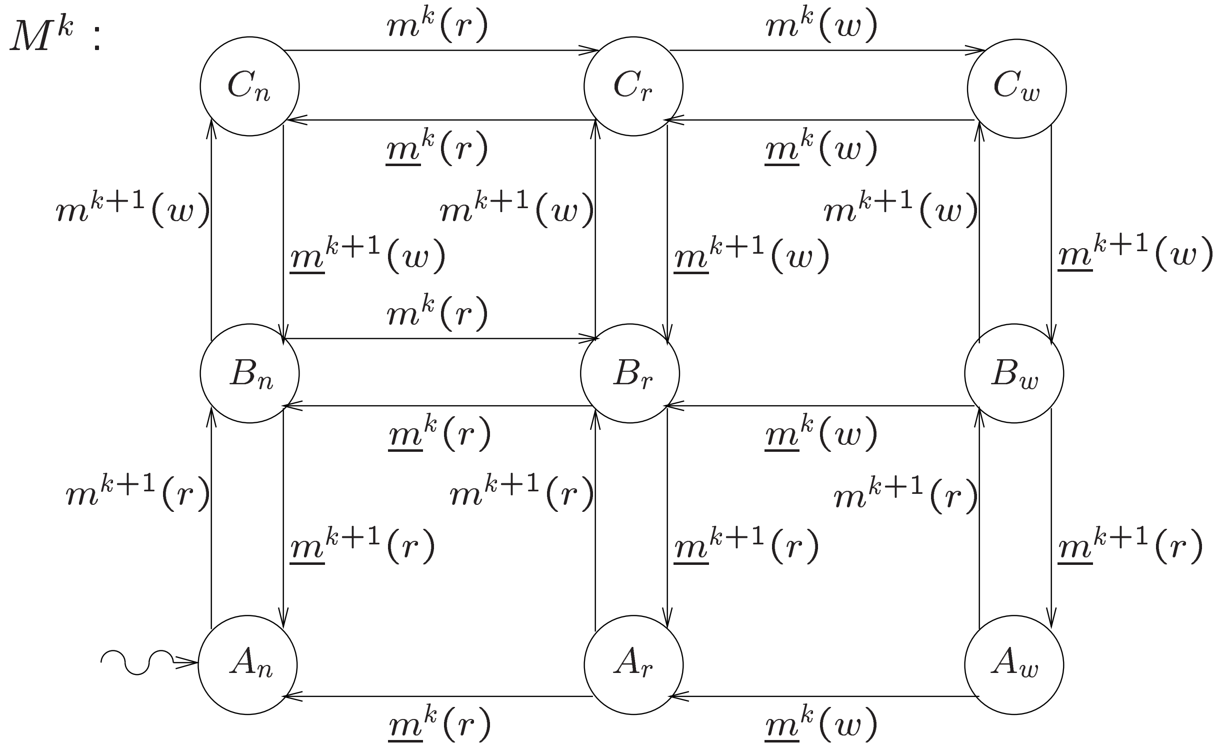
  - author of paper may delegate publication responsibilities to co-author

  - co-author may select a journal and grant $m^2(r)$ rights to editor-in-chief

  - editor-in-chief may grant $m(r)$ rights to assistent editors

  - assistent editors may grant and revoke read access to reviewers

Automaton M^k: Meta Access at Layer k



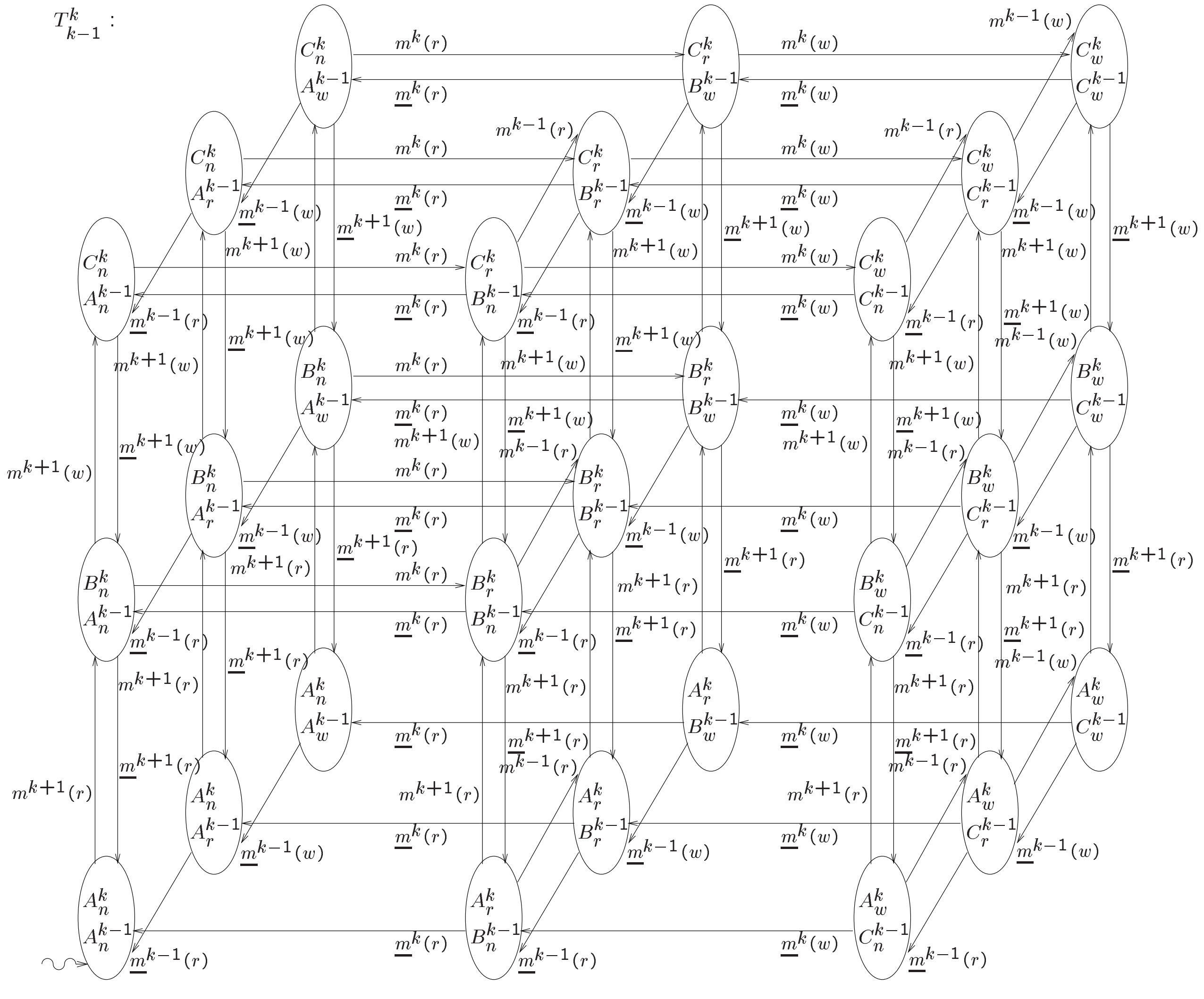
$m^k(r)$, $\underline{m}^k(r)$, $m^k(w)$, and $\underline{m}^k(w)$ **output** actions

$m^{k+1}(r)$, $\underline{m}^{k+1}(r)$, $m^{k+1}(w)$, and $\underline{m}^{k+1}(w)$ **input**

$\{M^0, M^1, \ldots, M^n\}$ composable system ($n \geq k$)

$\Rightarrow$ compose *iterated* team over $\{M^0, M^1, \ldots, M^n\}$
by choosing a transition relation!

Example Team Automaton T^{k}_{k-1} over $\{M^{k-1}, M^{k}\}$



T^k_{k-1} :

$m^k(r)$, $\underline{m}^k(r)$, $m^k(w)$, and $\underline{m}^k(w)$ **master-slave** synchronizations with master M^k and slave M^{k-1}

T^k_{k-1} :

shallow vs. *deep* revocation

Why Accept This at CSCW 2000?

- spatial access control accepted metaphor:

  - demystifies access control concepts for end user

- team automata framework accepted model:

  - flexible (role of actions, choice of transitions)
  - scalable (modular construction, iterated teams)
  - extendible (time, priorities)

$\Rightarrow$ metaphor and model together make a powerful team!

- forces user to make explicit design choices &
- provides mathematically precise analytic tools for proving crucial design properties, before implementation!