On a General Notion of Transformation for Multiagent Systems

Jochen Pfalzgraf, Thomas Soboll

{jpfalz, tsoboll}@cosy.sbg.ac.at

University of Salzburg

Department of Computer Sciences

A-5020 Salzburg, Austria

Outline of the talk

- Motivation
- Cat modeling of Multiagent Systems (MAS)
- Base Diagrams as Typed Categories
- The Category $\mathbb{MAS}$
- Application of The Double Pushout Approach
- Conclusion and Outlook

Motivation

- Wooldrige: "There is no such thing as a single agent system"

Motivation

- Wooldrige: "There is no such thing as a single agent system"
- Significant information of a MAS is in the cooperation and communication structure of a MAS

Motivation

- Wooldrige: "There is no such thing as a single agent system"
- Significant information of a MAS is in the cooperation and communication structure of a MAS
- This information is represented by a corresponding relations $\longrightarrow$ arrow diagram, we call it base diagram of the MAS.

Motivation

- Wooldrige: "There is no such thing as a single agent system"
- Significant information of a MAS is in the cooperation and communication structure of a MAS
- This information is represented by a corresponding relations $\longrightarrow$ arrow diagram, we call it base diagram of the MAS.
- A MAS is a dynamic system $\longrightarrow$ the relations in the system change $\longrightarrow$ the base diagram changes. Transformation System (DPO)

Motivation

- Wooldrige: "There is no such thing as a single agent system"
- Significant information of a MAS is in the cooperation and communication structure of a MAS
- This information is represented by a corresponding relations $\longrightarrow$ arrow diagram, we call it base diagram of the MAS.
- A MAS is a dynamic system $\longrightarrow$ the relations in the system change $\longrightarrow$ the base diagram changes. Transformation System (DPO)
- Goal: Develop a toolbox for MAS.

Motivation

- Wooldrige: "There is no such thing as a single agent system"
- Significant information of a MAS is in the cooperation and communication structure of a MAS
- This information is represented by a corresponding relations $\longrightarrow$ arrow diagram, we call it base diagram of the MAS.
- A MAS is a dynamic system $\longrightarrow$ the relations in the system change $\longrightarrow$ the base diagram changes. Transformation System (DPO)
- Goal: Develop a toolbox for MAS.
- No internal modeling of an agent but analysis of the structures in a MAS.

Motivation

- Wooldrige: "There is no such thing as a single agent system"
- Significant information of a MAS is in the cooperation and communication structure of a MAS
- This information is represented by a corresponding relations $\longrightarrow$ arrow diagram, we call it base diagram of the MAS.
- A MAS is a dynamic system $\longrightarrow$ the relations in the system change $\longrightarrow$ the base diagram changes. Transformation System (DPO)
- Goal: Develop a toolbox for MAS.
- No internal modeling of an agent but analysis of the structures in a MAS.

CAT Modelling of MAS

Category Theory : Unifying mathematical modeling language with many **constructive features**.

- The general communication and cooperation structure is represented in a so called base diagram of the MAS.

CAT Modelling of MAS

Category Theory : Unifying mathematical modeling language with many **constructive features**.

- The general communication and cooperation structure is represented in a so called base diagram of the MAS.
- Each base diagram forms a typed category.

CAT Modelling of MAS

Category Theory : Unifying mathematical modeling language with many **constructive features**.

- The general communication and cooperation structure is represented in a so called base diagram of the MAS.
- Each base diagram forms a typed category.
- Typed objects represent agents.

CAT Modelling of MAS

Category Theory : Unifying mathematical modeling language with many **constructive features**.

- The general communication and cooperation structure is represented in a so called base diagram of the MAS.
- Each base diagram forms a typed category.
- Typed objects represent agents.
- Object types represent properties of agents.

CAT Modelling of MAS

Category Theory : Unifying mathematical modeling language with many **constructive features**.

- The general communication and cooperation structure is represented in a so called base diagram of the MAS.
- Each base diagram forms a typed category.
- Typed objects represent agents.
- Object types represent properties of agents.
- Typed morphism represent different relations (communication in general) between the agents.

Base Diagrams as Categories

Let $R \subset X \times X$ denote a general relation. We associate with it the category denoted by PATH(X,R), PATH(X) or just **PATH**.

Base Diagrams as Categories

Let $R \subset X \times X$ denote a general relation. We associate with it the category denoted by PATH(X,R), PATH(X) or just **PATH**.

- Objects: the elements $x \in X$.
- Arrows, Morphisms: sequences (paths) of adjacent arrows.

Base Diagrams as Categories

Let $R \subset X \times X$ denote a general relation. We associate with it the category denoted by PATH(X,R), PATH(X) or just **PATH**.

- Objects: the elements $x \in X$.
- Arrows, Morphisms: sequences (paths) of adjacent arrows.

This naturally defines *composition of arrows*.

Base Diagrams as Categories

Let $R \subset X \times X$ denote a general relation. We associate with it the category denoted by PATH(X,R), PATH(X) or just **PATH**.

- Objects: the elements $x \in X$.
- Arrows, Morphisms: sequences (paths) of adjacent arrows.

This naturally defines *composition of arrows*.

There is a morphism (1) iff xRy.

(1) $$x \longrightarrow y$$

Base Diagrams as Categories

Let $R \subset X \times X$ denote a general relation. We associate with it the category denoted by PATH(X,R), PATH(X) or just **PATH**.

- Objects: the elements $x \in X$.
- Arrows, Morphisms: sequences (paths) of adjacent arrows.

This naturally defines *composition of arrows*.

There is a morphism (1) iff xRy. For arrows $x \to y$ and $y \to z$ there is no "direct arrow" in general (2) $x \to z$.

(2)

$$\begin{tikzcd} x \arrow[r] & y \arrow[dl] \\ z & \end{tikzcd}$$

Base Diagrams as Categories

Let $R \subset X \times X$ denote a general relation. We associate with it the category denoted by PATH(X,R), PATH(X) or just **PATH**.

- Objects: the elements $x \in X$.
- Arrows, Morphisms: sequences (paths) of adjacent arrows.

This naturally defines *composition of arrows*.

There is a morphism (1) iff xRy. For arrows $x \to y$ and $y \to z$ there is no "direct arrow" in general (2) $x \to z$. But a sequence (path) of consecutive arrows (3) $x \to y \to z$.

(3)

$$\begin{tikzcd} x \arrow[r] \arrow[d, "x\vec{y}z"'] & y \arrow[dl] \\ z & \end{tikzcd}$$

Base Diagrams as Categories

Let $R \subset X \times X$ denote a general relation. We associate with it the category denoted by PATH(X,R), PATH(X) or just **PATH**.

- Objects: the elements $x \in X$.
- Arrows, Morphisms: sequences (paths) of adjacent arrows.

This naturally defines *composition of arrows*.

There is a morphism (1) iff xRy. For arrows $x \longrightarrow y$ and $y \longrightarrow z$ there is no "direct arrow" in general (2) $x \longrightarrow z$. But a sequence (path) of consecutive arrows (3) $x \longrightarrow y \longrightarrow z$.
The identity arrow for each object is an arrow of length 0.

Thus, **PATH** becomes a **category**.

Base Diagrams as Categories (2)

- An arbitrary binary relation R on X induces a corresponding **Arrow Diagram** $\mathbb{D}$.

Base Diagrams as Categories (2)

- An arbitrary binary relation R on X induces a corresponding **Arrow Diagram** $\mathbb{D}$.
- Every Arrow Diagram $\mathbb{D}$ can be interpreted as a category.

Base Diagrams as Categories (2)

- An arbitrary binary relation R on X induces a corresponding **Arrow Diagram** $\mathbb{D}$.
- Every Arrow Diagram $\mathbb{D}$ can be interpreted as a category.
- A Multiagent System has a set of relations. (e. g."dominant agents", "power criteria", "skills/qualification").

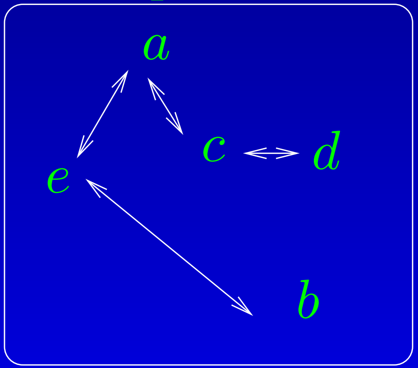
Base Diagrams as Categories (2)

- An arbitrary binary relation R on X induces a corresponding **Arrow Diagram** $\mathbb{D}$.
- Every Arrow Diagram $\mathbb{D}$ can be interpreted as a category.
- A Multiagent System has a set of relations. (e. g."dominant agents", "power criteria", "skills/qualification").
- The different relations in a MAS are represented by different types of arrows in the base diagram.

Base Diagrams as Categories (2)

- An arbitrary binary relation R on X induces a corresponding **Arrow Diagram** $\mathbb{D}$.
- Every Arrow Diagram $\mathbb{D}$ can be interpreted as a category.
- A Multiagent System has a set of relations. (e. g."dominant agents", "power criteria", "skills/qualification").
- The different relations in a MAS are represented by different types of arrows in the base diagram.

  Example base Diagram: MAS

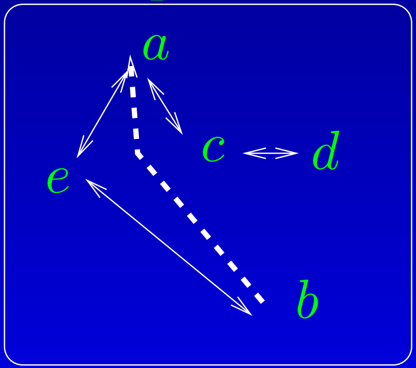
- 

  One relation (one arrow-type)

Base Diagrams as Categories (2)

- An arbitrary binary relation R on X induces a corresponding **Arrow Diagram** $\mathbb{D}$.
- Every Arrow Diagram $\mathbb{D}$ can be interpreted as a category.
- A Multiagent System has a set of relations. (e. g."dominant agents", "power criteria", "skills/qualification").
- The different relations in a MAS are represented by different types of arrows in the base diagram.

Example base Diagram: MAS

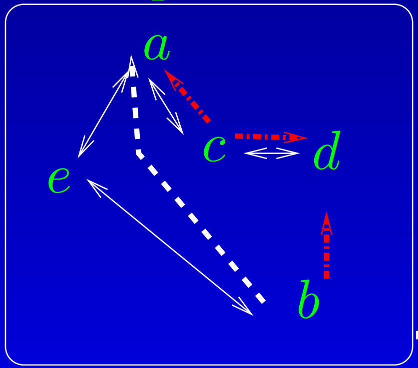
- 

Two relations (two arrow-types)

Base Diagrams as Categories (2)

- An arbitrary binary relation R on X induces a corresponding **Arrow Diagram** $\mathbb{D}$.
- Every Arrow Diagram $\mathbb{D}$ can be interpreted as a category.
- A Multiagent System has a set of relations. (e. g."dominant agents", "power criteria", "skills/qualification").
- The different relations in a MAS are represented by different types of arrows in the base diagram.

  Example base Diagram: MAS

- Three relations (three arrow-types)

Typed Categories

A typed category $\mathbf{T}$ consists of:

- objects
- arrow types
- object types
- a map assigning to each object a set of object types
- a set of arrows for each tripple (arrowtype,domain,codomain)
- the composition is defined typewise.

We define:

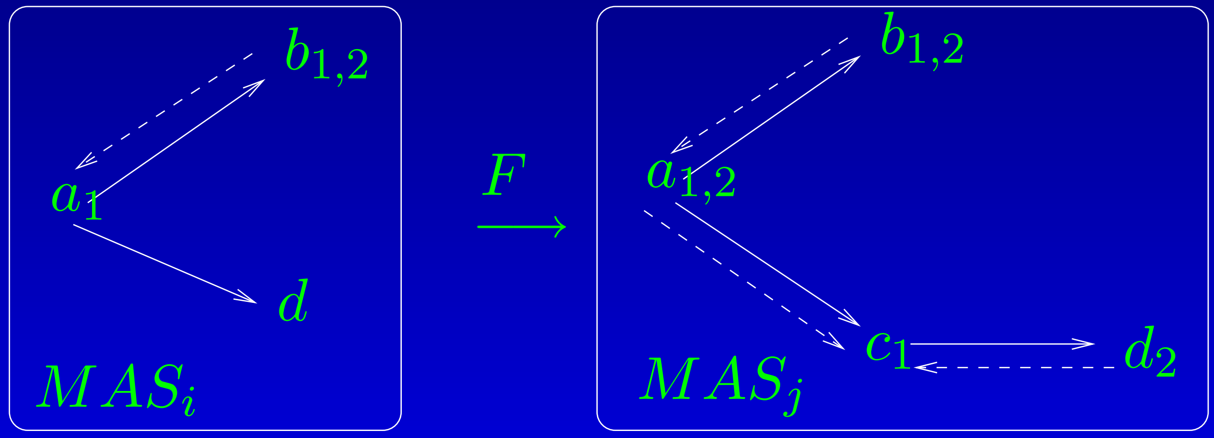
Typed Subcategories and Typed Functors. Definition

The Category $\mathbb{MAS}$

Category of all small (set of objects) categories Cat. Functors act as morphisms between categories.

We can build the Category $\mathbb{MAS}$

- Objects: Base Diagrams of MAS (typed categories)
- Morphisms: Covariat typed functors



arrows for dominance relation

arrows for communication relation

$\mathbb{MAS}$ Morphism

A $\mathbb{MAS}$ morphism $F : MAS_i \rightarrow MAS_j$ is a quadruple $F = (F_{AT}, F_A, F_O, F_{\mathfrak{P}})$ of maps:

$$
\begin{tikzcd}
ArrTypes(MAS_i) \arrow[r, "F_{AT}"] \arrow[dr, phantom, "(1)"] & ArrTypes(MAS_j) \\
Arr(MAS_i) \arrow[u, "\pi_{MAS_i}"] \arrow[r, "F_A"] \arrow[d, shift right, "codom_{MAS_i}"'] \arrow[d, shift left, "dom_{MAS_i}"] \arrow[dr, phantom, "(2)"] & Arr(MAS_j) \arrow[u, "\pi_{MAS_j}"'] \arrow[d, shift right, "codom_{MAS_j}"'] \arrow[d, shift left, "dom_{MAS_j}"] \\
Obj(MAS_i) \arrow[r, "F_O"] \arrow[d, "\tau_{MAS_i}"] \arrow[dr, phantom, "(3)"] & Obj(MAS_j) \arrow[d, "\tau_{MAS_j}"] \\
\mathfrak{P}(ObjTypes(MAS_i)) \arrow[r, "F_{\mathfrak{P}}"] & \mathfrak{P}(ObjTypes(MAS_i))
\end{tikzcd}
$$

In this category pushouts exist.

DPO Approach

In $\mathbb{MAS}$ a $\mathbb{MAS}$ -production$p = (p^l, p^r)$ is defined as a pair of $\mathbb{MAS}$ morphisms with common domain.

$$L \xleftarrow{p^l} MAS^I \xrightarrow{p^r} R$$

DPO Approach

In $\mathbb{MAS}$ a $\mathbb{MAS}$ -production$p = (p^l, p^r)$ is defined as a pair of $\mathbb{MAS}$ morphisms with common domain. Given a $\mathbb{MAS}$ -production p, a $\mathbb{MAS}$ object MAS^l and a $\mathbb{MAS}$ morphism $m : codom(p^l) \rightarrow MAS^l$, called match, defines a

$$\begin{tikzcd} L \arrow[d, "m"'] & MAS^I \arrow[l, "p^l"'] \arrow[r, "p^r"] & R \\ MAS^l & & \end{tikzcd}$$

DPO Approach

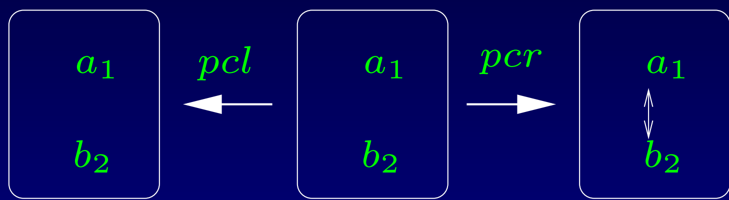
In $\mathbb{MAS}$ a $\mathbb{MAS}$ -production$p = (p^l, p^r)$ is defined as a pair of $\mathbb{MAS}$ morphisms with common domain. Given a $\mathbb{MAS}$ -production p, a $\mathbb{MAS}$ object MAS^l and a $\mathbb{MAS}$ morphism $m : codom(p^l) \rightarrow MAS^l$, called match, defines a direct transformation step. $MAS^l \Rightarrow MAS^r$.

$$\begin{tikzcd}
L \arrow[d, "m"'] & MAS^I \arrow[l, "p^l"'] \arrow[r, "p^r"] \arrow[d, "g"] & R \arrow[d, "g^r"'] \\
MAS^l & MAS^C \arrow[l, "p^{-l}"'] \arrow[r, "p^{-r}"] & MAS^r
\end{tikzcd}$$

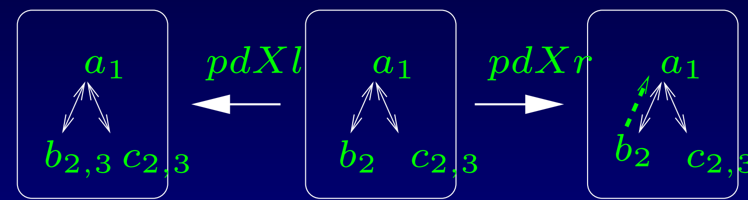
This diagram illustrates a Double Pushout, for more details we refer to the book "Fundamentals of Algebraic Graph Transformation".

Example (1)

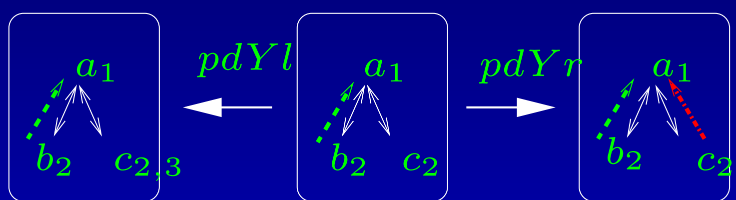
Based on four relations that change while the MAS performs its task five productions are defined that model the application conditions of actions and the actions.



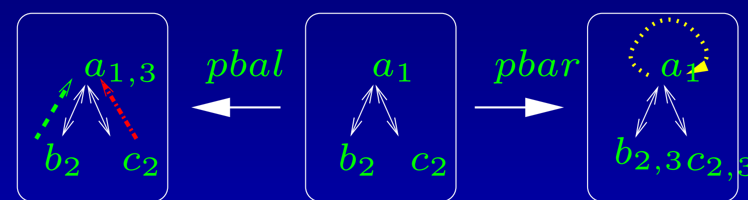
procuction $pc = (pcl, pcr)$



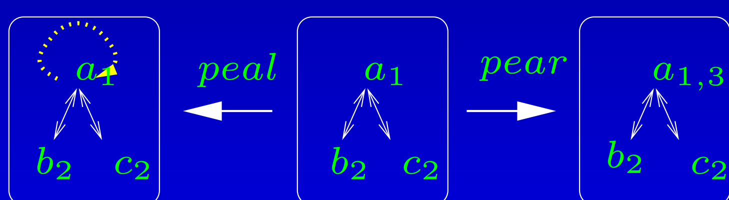
production $pdX = (pdXl, pdXr)$



production $pdY = (pdYl, pdYr)$



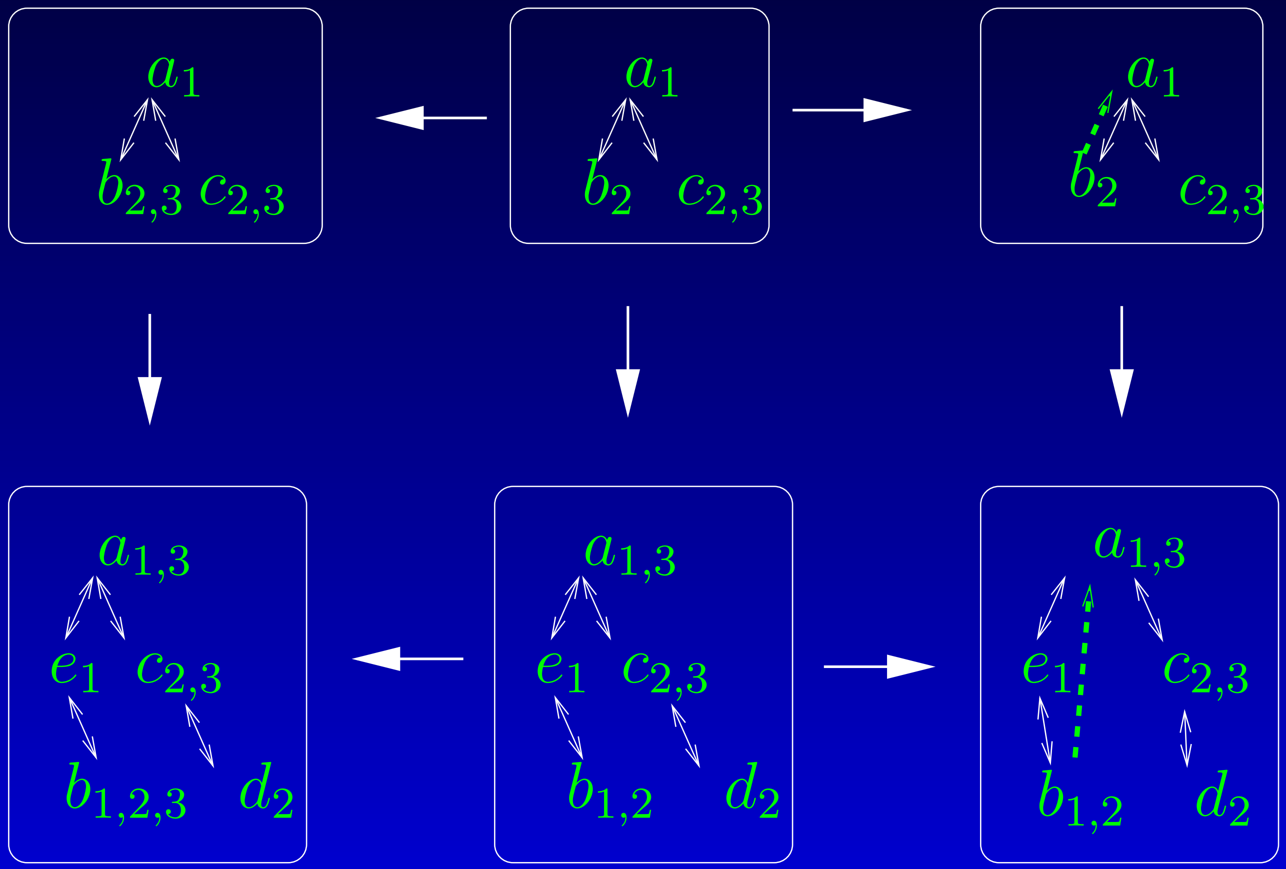
production $pba = (pbal, pbar)$



production $pea = (peal, pear)$

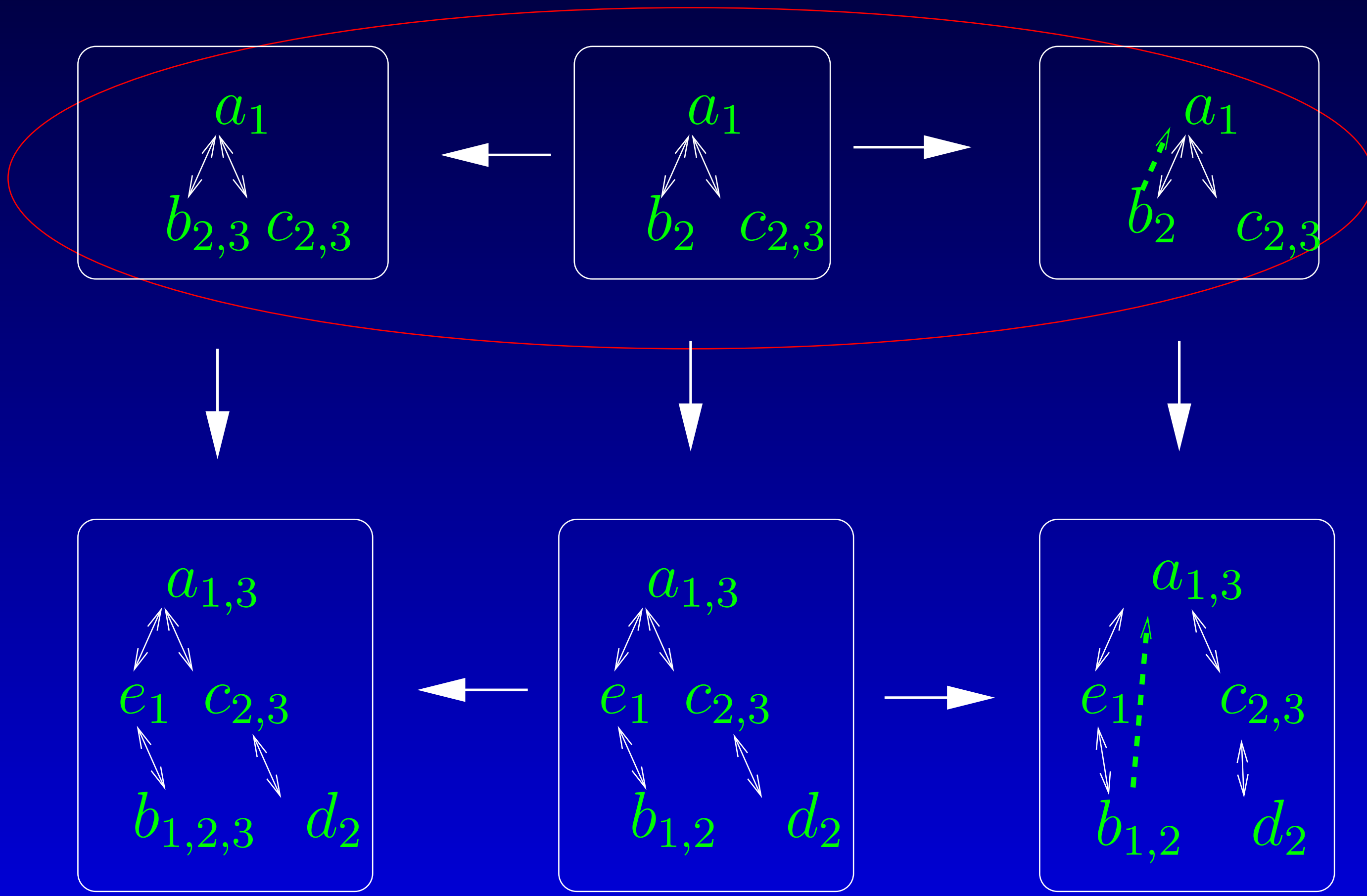
Example (2)

Application of production pdX to a given $\mathbb{MAS}$ object.



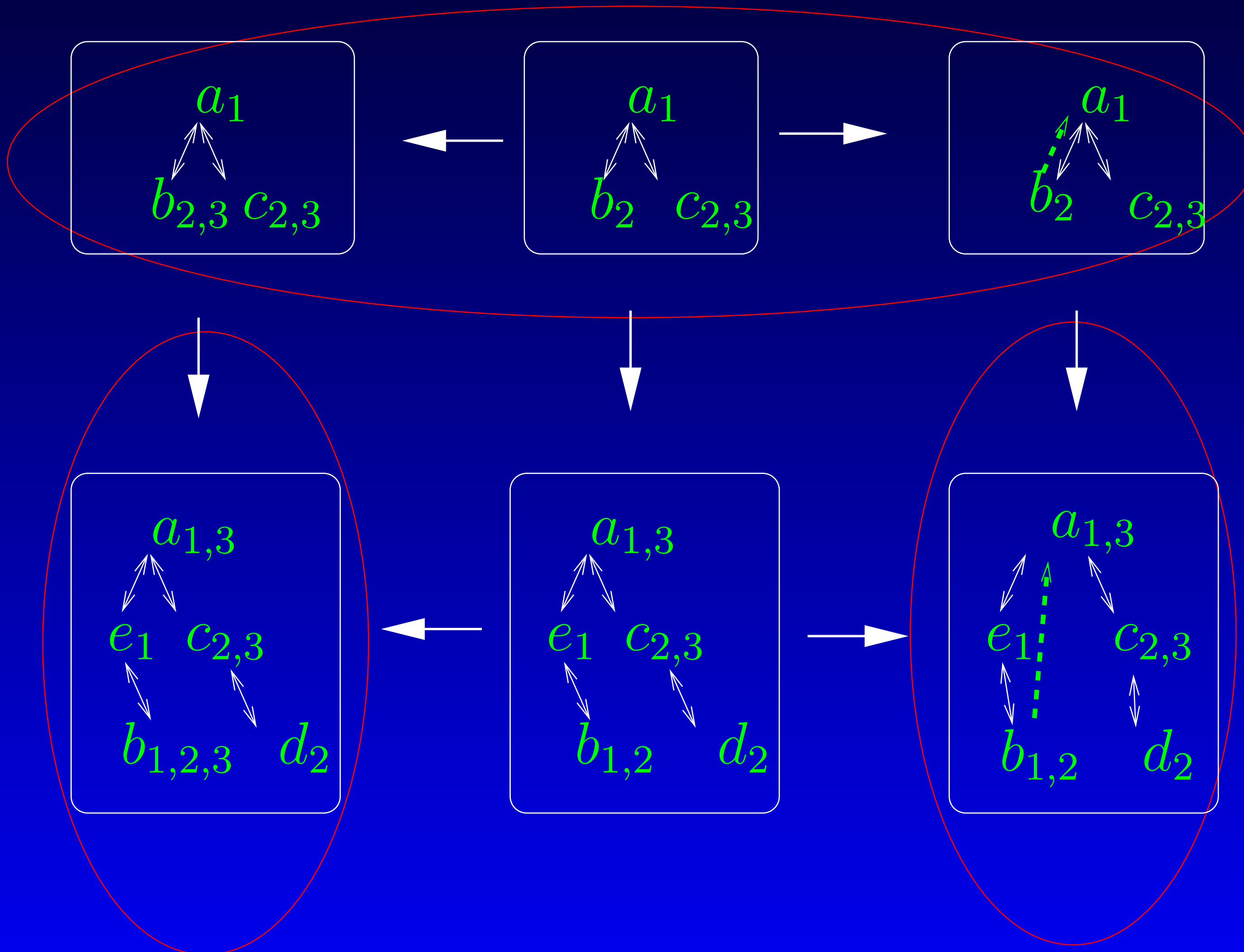
Example (2)

Application of production pdX to a given $\mathbb{MAS}$ object.



Example (2)

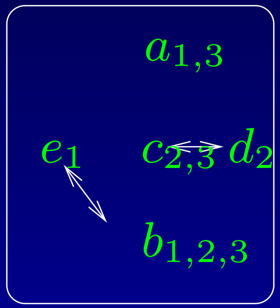
Application of production pdX to a given $\mathbb{MAS}$ object.



Example (3)

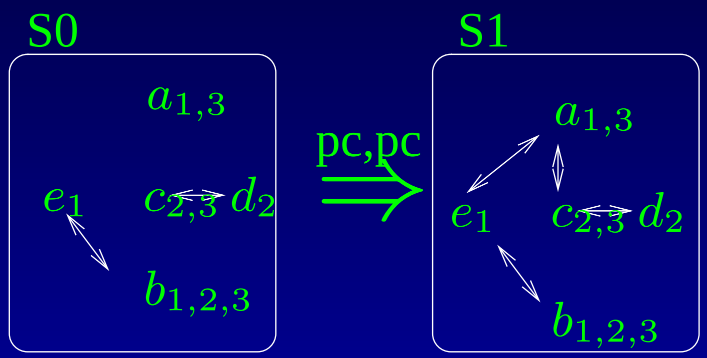
System run: initial $\mathbb{MAS}$ object S0. The productions are applied together with suitable matches.

S0



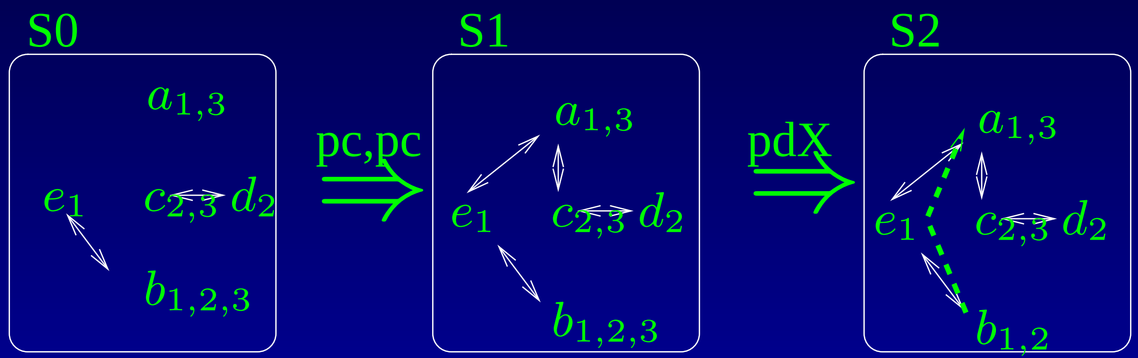
Example (3)

System run: initial $\mathbb{MAS}$ object S0. The productions are applied together with suitable matches.



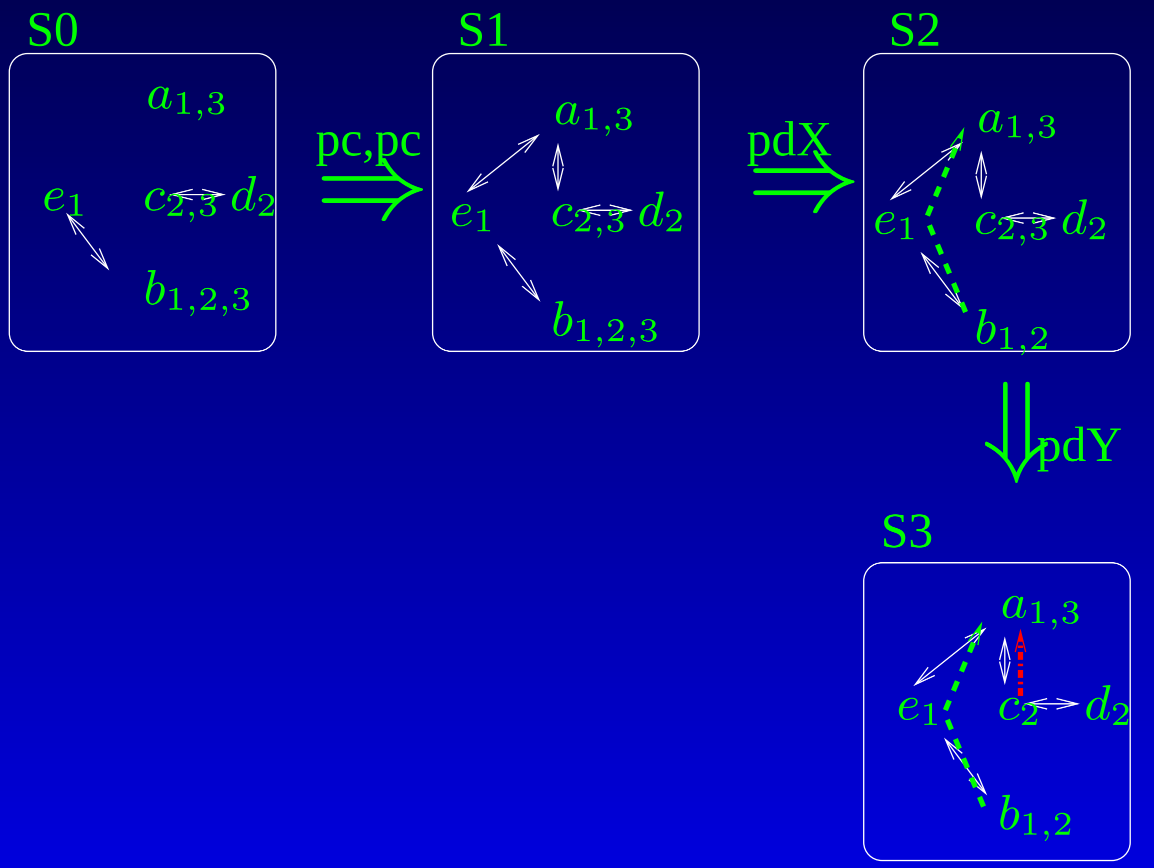
Example (3)

System run: initial $\mathbb{MAS}$ object S0. The productions are applied together with suitable matches.



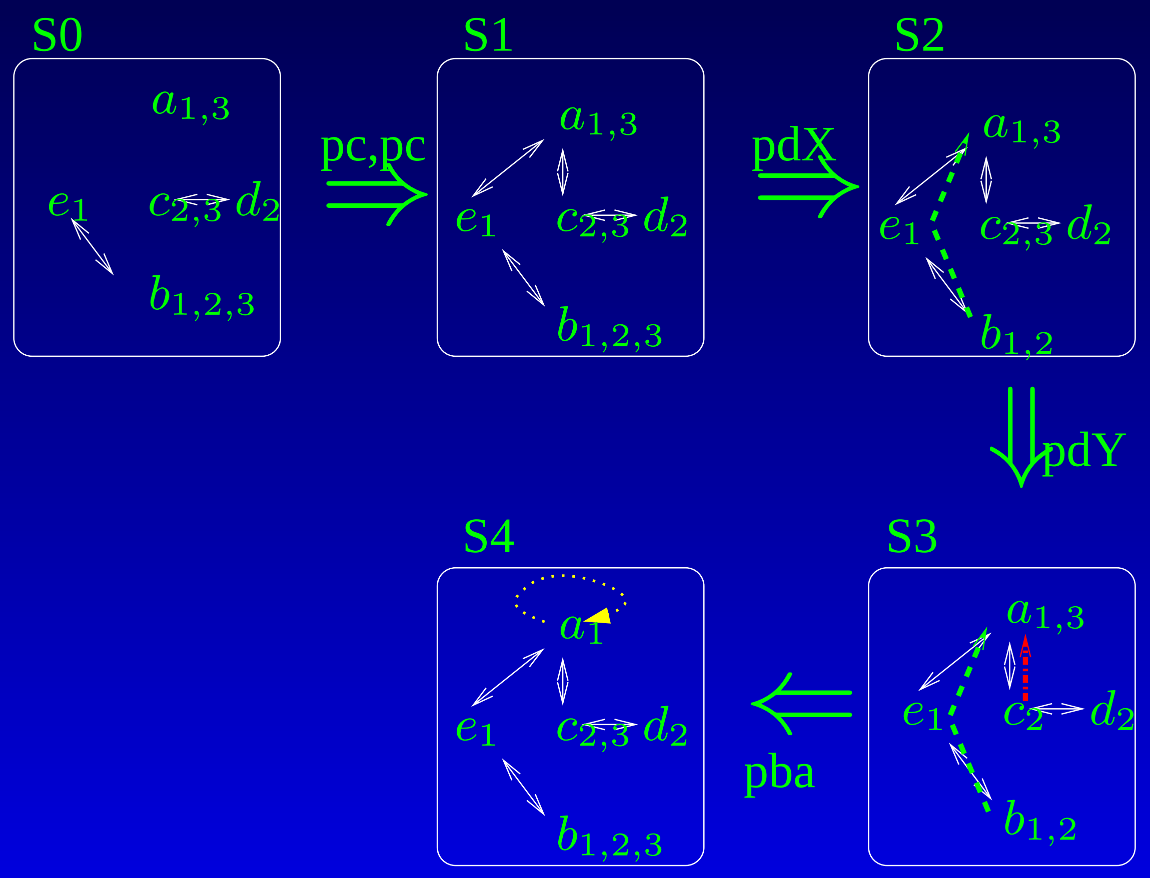
Example (3)

System run: initial $\mathbb{MAS}$ object S0. The productions are applied together with suitable matches.



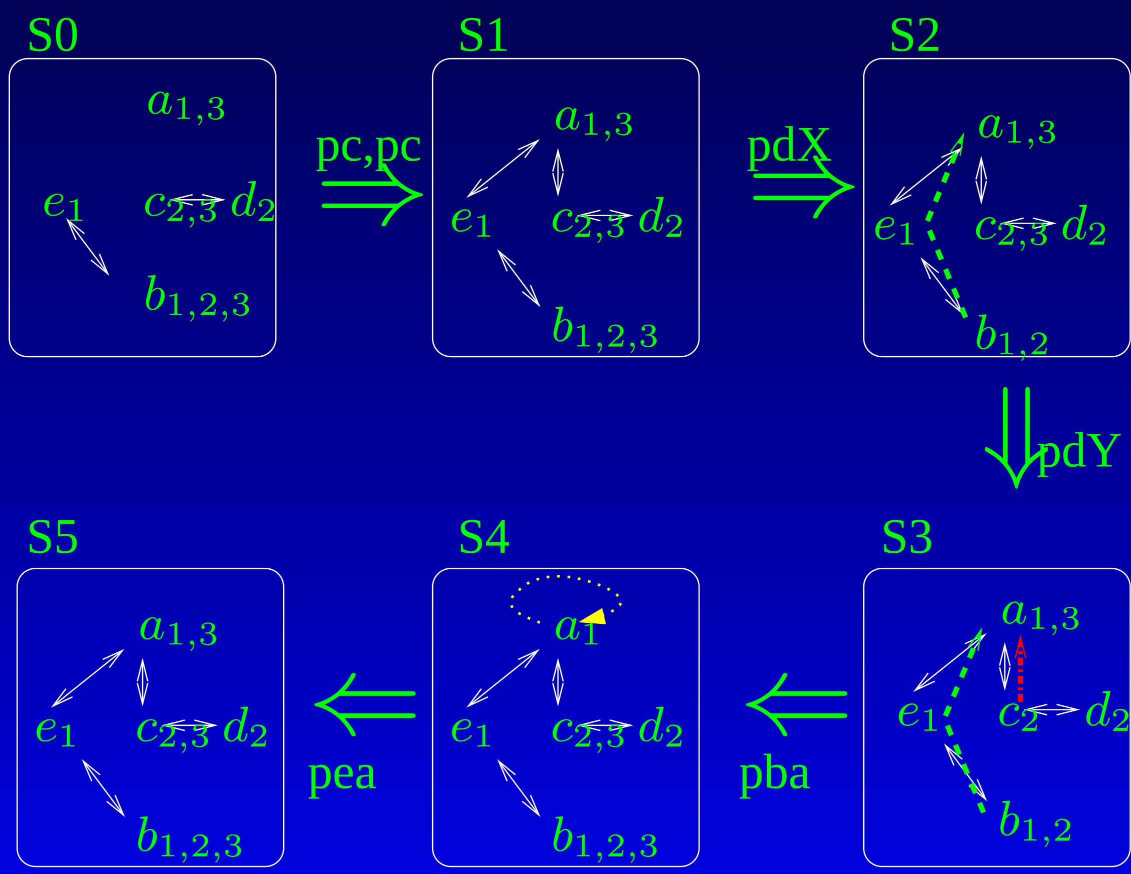
Example (3)

System run: initial $\mathbb{MAS}$ object S0. The productions are applied together with suitable matches.



Example (3)

System run: initial $\mathbb{MAS}$ object S0. The productions are applied together with suitable matches.



Conclusion and Outlook

- The concept of $\mathbb{MAS}$ transformations is a natural way to describe changes in the base diagram of Multiagent Systems.
- It is independent of the implementation of the agents.
- We analyze MAS on the basis of their cooperation and communication structures.
- Actions and their application conditions in a MAS are described by productions in $\mathbb{MAS}$.

Conclusion and Outlook (2)

- ’Relational Fibering’ to model local global interactions in the relational structure of a MAS.
- A first application of this approach is to compute subcategories of a MAS on demand, by taking the collection of the fibers over a defined set of agents as a starting point.
- Limits and colimits of Multiagent Systems for different morphism types (universal communicator).
- Action Planning.

Category

A category $\mathbf{C}$ consists of a class of objects denoted by $A, B, C, ... \in Obj(\mathbf{C})$. For each pair of objects A, B there is a set of morphisms, $Mor(A, B)$, also denoted by $\mathbf{C}(A, B)$ (the "arrows" between A and B). $\mathbf{C}(A_1, B_1)$ and $\mathbf{C}(A_2, B_2)$ are disjoint unless $A_1 = A_2$ and $B_1 = B_2$. (Note that $Mor(A, B)$ can be empty). There is a composition operation on morphisms: if $f : A \longrightarrow B$ and $g : B \longrightarrow C$ are morphisms, then there is a morphism $g \circ f : A \longrightarrow C$, the composition of f and g. In a category the following axioms have to hold.

- The composition of morphisms is associative, that is for morphisms $f : A \longrightarrow B, g : B \longrightarrow C$ and $h : C \longrightarrow D$ it holds: $h \circ (g \circ f) = (h \circ g) \circ f$.
- For every object $A \in Obj(\mathbf{C})$ there is the identity morphism id_A with the properties $f \circ id_A = f$ and $id_B \circ f = f$ for all $f : A \longrightarrow B$.

There are two operations assigning to each $\mathbf{C}$ -arrow f a $\mathbf{C}$ -object $dom(f)$ and a $\mathbf{C}$ -object $codom(f)$. If $A = dom(f)$ and $B = codom(f)$ we display this as $f : A \longrightarrow B$ or $A \xrightarrow{f} B$.

Typed Categories

A typed category $\mathbf{T}$ consists of:

- a collection of objects $Obj(\mathbf{T})$
- a set of arrow types denoted by $ArrTypes(\mathbf{T})$
- a set of object types denoted by $ObjTypes(\mathbf{T})$
- a map $\tau_{\mathbf{T}} : Obj(\mathbf{T}) \to \mathfrak{P}(ObjTypes(\mathbf{T}))$ assigning to each object in $Obj(\mathbf{T})$ a set of object types where $\mathfrak{P}(ObjTypes(\mathbf{T}))$
- and for each triple (A, B, t) with $A, B \in Obj(\mathbf{T})$ and $t \in ArrTypes(\mathbf{T})$ a set of $\mathbf{T}$-morphisms $Mor_t(A, B)$ (We call $f \in Mor_t(A, B)$ a typed morphism from A to B and write $f : A \to_t B$)
- the composition is defined typewise.

back

Typed Subcategory

For $t \in ArrTypes(\mathbf{T})$ the category $\mathbf{S}_t$ is called a typed subcategory of $\mathbf{T}$ if the following holds:

- Every $\mathbf{S}_t$ object is a $\mathbf{T}$-object.
- For A $\in Obj(\mathbf{S}_t)$ the set of object types in $\mathbf{S}_t$, $\tau_{\mathbf{S}_t}(A)$ equals the set of object types $\tau_{\mathbf{T}}(A)$ in $\mathbf{T}$.
- For A,B $\in Obj(\mathbf{S}_t)$ the set of morphisms from A to B in $\mathbf{S}_t$ is a subset of the set of $\mathbf{T}$-morphisms of type t denoted by $Mor_t(A, B)$.

$\mathbf{S}_t$ is called a full typed subcategory of $\mathbf{T}$ if for all $\mathbf{S}_t$ objects A, B it holds: the set of morphisms in $\mathbf{S}_t$ from A to B equals the set of $\mathbf{T}$-morphisms of type t, denoted by $Mor_t(A, B)$. back

Typed Functor

Let MAS_i and MAS_j be two typed Categories. A typed functor $F : MAS_i \to MAS_j$ assigns to every object $A \in Obj(MAS_i)$ an object $F(A) \in Obj(MAS_j)$, to every arrow type $t \in ArrTypes(MAS_i)$ an arrow type $F(t) \in ArrTypes(MAS_j)$, to every object type $o \in ObjTypes(MAS_i)$ an object type $F(o) \in ObjTypes(MAS_j)$ and to every typed morphism $f : A \to_t B$ of type t a morphism $F(f) : F(A) \to_{F(t)} F(B)$ such that for morphisms $f : A \to_t B$, $g : B \to_t C$, id_A and $A \in Obj(MAS_i)$ it holds:

- $F(g \circ f) = F(g) \circ F(f)$
- $F(id_A) = id_{F(A)}$
- $F(\tau_{MAS_i}(A)) \subseteq \tau_{MAS_j}(F(A))$

back

Definition Pushout

The diagram (1) is called a pushout (or fibred coproduct) square if it commutes (i.e. $g' \circ f = f' \circ g$) and for any commuting square (i.e. $g'' \circ f = f'' \circ g$) of the form (2) there exists a unique morphism $k : D \to D'$ such that the diagram (3) commutes (i.e. $g'' = k \circ g'$ and $f'' = k \circ f'$).

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] \arrow[dr, phantom, "(1)"] & B \arrow[d, "g'"] \\ C \arrow[r, "f'"'] & D \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] \arrow[dr, phantom, "(2)"] & B \arrow[d, "g''"] \\ C \arrow[r, "f''"'] & D' \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] \arrow[dr, phantom, "(3)"] & B \arrow[d, "g'"] \arrow[ddr, bend left, "g''"] & \\ C \arrow[r, "f'"'] \arrow[drr, bend right, "f''"'] & D \arrow[dr, "k"] & \\ & & D' \end{tikzcd}$$

As an example of a pushout situation we consider two morphisms in the category **SET** $f : A \to B$ and $g : A \to C$, a pushout in **SET** is obtained by forming the disjoint union $B \coprod C$ and then identifying $f(x)$ with $g(x)$ for all $x \in A$.

back

Pushout Construction

$$
\begin{tikzcd}
 & ArrTypes(I) \arrow[rr, "G_{AT}"] \arrow[dl, "F_{AT}"'] & & ArrTypes(K) \arrow[dl, "F'_{AT}"'] \\
ArrTypes(J) \arrow[rr, "G'_{AT}"] & & ArrTypes(PO) & \\
 & Arr(I) \arrow[uu, "\pi_I"'] \arrow[rr, "G_A"] \arrow[dl, "F_A"'] \arrow[dd, shift right, "dom_I"'] \arrow[dd, shift left, "codom_I"] & & Arr(K) \arrow[uu, "\pi_K"'] \arrow[dl, "F'_A"'] \arrow[dd, shift right, "dom_K"'] \arrow[dd, shift left, "codom_K"] \\
Arr(J) \arrow[uu, "\pi_J"'] \arrow[rr, "G'_A"] \arrow[dd, shift right, "dom_J"'] \arrow[dd, shift left, "codom_J"] & & Arr(PO) \arrow[uu, dotted, "\pi_{PO}"'] \arrow[dd, dotted, shift right, "dom_{PO}"'] \arrow[dd, dotted, shift left, "codom_{PO}"] & \\
 & Obj(I) \arrow[rr, "G_O"] \arrow[dl, "F_O"'] \arrow[dd, "\tau_I"'] & & Obj(K) \arrow[dl, "F'_O"'] \arrow[dd, "\tau_K"'] \\
Obj(J) \arrow[rr, "G'_O"] \arrow[dd, "\tau_J"'] & & Obj(PO) \arrow[dd, dotted, "\tau_{PO}"'] & \\
 & \mathfrak{P}(ObjTypes(I)) \arrow[rr, "G_{\mathfrak{P}}"] \arrow[dl, "F_{\mathfrak{P}}"'] & & \mathfrak{P}(ObjTypes(K)) \arrow[dl, "F'_{\mathfrak{P}}"'] \\
\mathfrak{P}(ObjTypes(J)) \arrow[rr, "G'_{\mathfrak{P}}"] & & \mathfrak{P}(ObjTypes(PO)) &
\end{tikzcd}
$$

Componentwise Pushout Visualization back

MAS

There is no general definition for Agent or Multiagent System (MAS).

Typical characteristics:

MAS

There is no general definition for Agent or Multiagent System (MAS).

Typical characteristics:

- ”... agents are computational systems that inhabit some **complex dynamic environment**, **sense** and **act autonomously** in this environment, and by doing so realize a set of **goals** or tasks for which they are designed.”

MAS

There is no general definition for Agent or Multiagent System (MAS).

Typical characteristics:

- ”... agents are computational systems that inhabit some **complex dynamic environment**, **sense** and **act autonomously** in this environment, and by doing so realize a set of **goals** or tasks for which they are designed.”
- ”An agent is a computer system that is **situated** in some **environment** and, that is capable of **autonomous action** in this environment to meet ist design objectives”