Foundations of a Measurement Set Data Model for Physical Experiments

La Londe Les Maures - **17 May 2012**

F. Viallefond.

Laboratoire d'Étude du Rayonnement et de la Matière en Astrophysique

Outline

- Context
- Physical quantities and measurements
- The underlying topology of the Measurement Set
- The topology of the binary relations
- Conclusions

Context: *origins, evolutions, new requirements*

- Mesurement Set (MS) *Kemball and Wieringa, 2000*

  1. premises (∼1998): J. Noordam T. Cornwell, ...

  2. specification (2000): excellent foundational structure
     - *only approximatively relational*

  3. AIPS++ implementation: nothing but a tabular representation
     - *not relational (⇒ semantic not captured)*
     - *not represented to persist in archives (⇒ can not be a standard)*

  4. effectively used only for offline for data reduction
     - *it requires a filler (⇒ lost of efficiency, ...)*
     - *used: ALMA, EVLA (LOFAR) ..(a lot of practices).*
     - *multi-beam etc...: missing specifications and practices*

Context: *origins, evolutions, new requirements*

- ALMA Science Data Model (ASDM), *Viallefond and Lucas 2006*

  1. MS adapted to support on-line data acquisition (*ref. ALMA and EVLA*)
     - *concept of configuration added*
  2. set-based implementation (rule of uniqueness)
     - *but lack polymorphic representations has corrupted semantics*
  3. physical quantities and measurements:
     - *no model for these concepts (ASDM $\notin$ AIPS++/CASA)*
  4. split between the bulk data and meta-data
     - *decoupling $\Rightarrow$ high flexibility (local constraints, optimizations, ...)*
  5. informal representation of the datasets as persistent objects for archives
     - *data model does not exist as a persistent type*

Context: *origins, evolutions, new requirements*

- New generation of instruments in radioastronomy

  1. multi-beam in focal planes
     - "cameras" in single-dish *e.g.* IRAM-30m
     - "phased arrays" in interferometry
       (*e.g.* ASKAP (prep-SKA), WSRT Apertif (under construction))
  2. multi-beam in aperture plane: phased arrays
     - LOFAR (commissioning)
     - EMBRACE prototype (SKA design study)

Data model: *effective goals and facts, purpose*

Effective goals

1. to describe any kind of physical quantities and measurements
2. integrate the measurements with their experimental context
3. test the model in concrete cases (radioastronomy, ...)

Facts

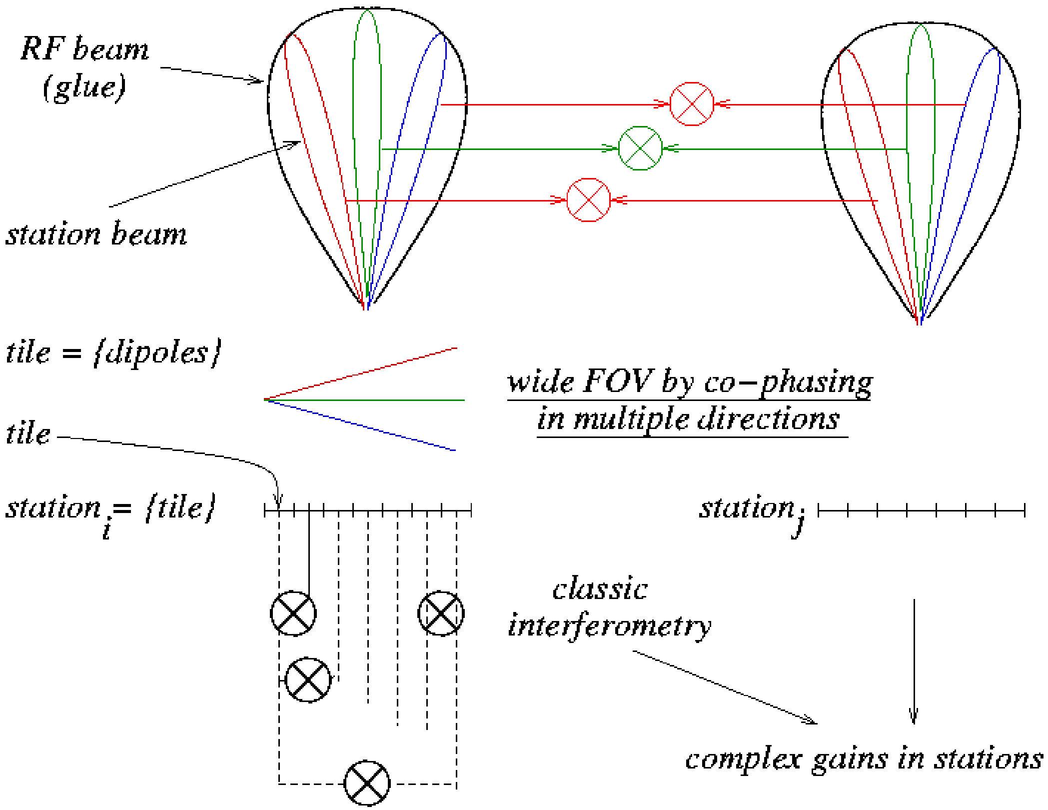
1. tightly connected to mathematics
2. connected to on-going research in maths

Aperture phased array use-case

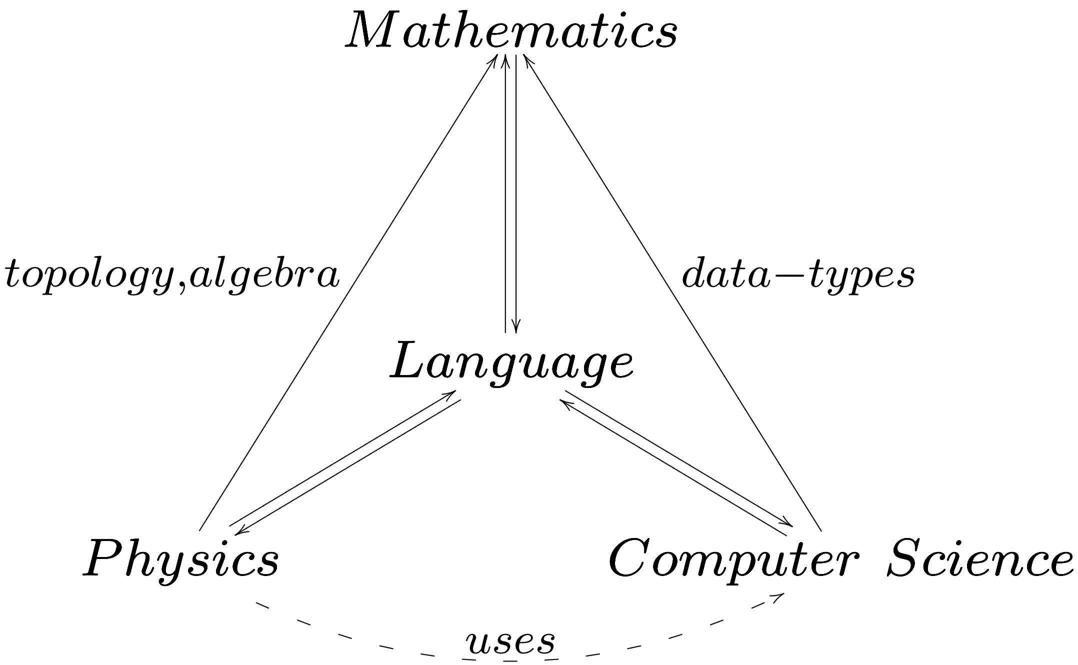
Storage capacity to record N beamlets (GBytes or Tbytes)

beamlets	data flow	10 min	1 hour	5 hours	10 hours
62	93 MB/s	54.2	325.5	1.6	3.2
124	185 MB/s	108.5	651	3.2	6.4
186	278 MB/s	162.8	976.4	4.8	9.6
248	370 MB/s	217	1.3	6.4	12.8

Aperture phased array use-case



Data model: *effective goals, facts, purpose*



Data model: *effective goals, facts, purpose*

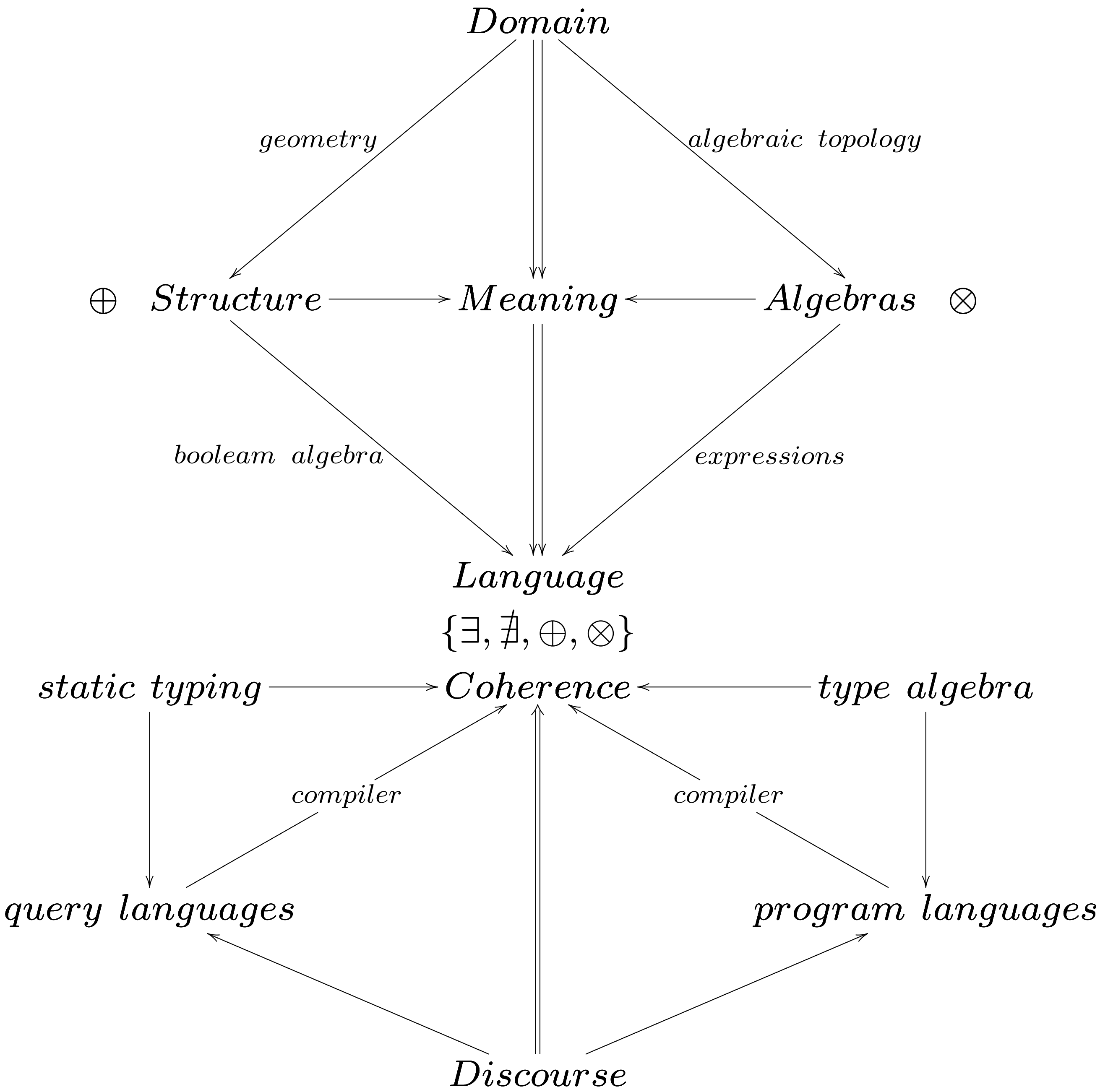
1. The purpose of a data-model in theoretical context is to transform in some form this verbal meaning into objects such that their semantics become part of an information system.

2. Software side:

    - effectiveness (static typing, high performance calculi)

    - expressiveness (formal language)

    - robustness (type-safe)


Domain
geometry
algebraic topology
⊕
Structure
Meaning
Algebras
⊗
booleam algebra
expressions
Language
{∃, ∄, ⊕, ⊗}
static typing
Coherence
type algebra
compiler
compiler
query languages
program languages
Discourse

Physical quantities: *goals*, *facts*, *static view*, *non-static view*

Our language express a physical quantity by a simple structure, a pair:

$$q_\varphi = q_v u_\varphi \quad e.g. \quad v = 12.3\ \mathrm{km.s}^{-1}$$

The units are important but not foundamental:

$$v = 12.3\ \mathrm{km.s}^{-1} = 12300\ \mathrm{m.s}^{-1}$$

The units and dimensionality are not sufficient to give the semantic:

Speed	$\mathrm{m.s}^{-1}$	L^1T^{-1}
EnergyDensity	$\mathrm{J.m}^{-3}$	$L^{-1}M^1T^{-2}$
RadiantEnergyDensity	$\mathrm{J.m}^{-3}$	$L^{-1}M^1T^{-2}$
Pressure	$\mathrm{Pa{=}N.m}^{-2}$	$L^{-1}M^1T^{-2}$
Radiance	$\mathrm{W.m}^{-2}.\mathrm{sr}^{-1}$	M^1T^{-3}
ApertureEfficiency	%	
SidebandRejection	dB	

Goal: be able to represent and use any kind of quantity.

Physical quantities: *goals, facts, static view, non-static view*

Facts: physical quantities

are the name of equations

may have dimensionnal units *e.g.* a speed ($m.s^{-1}$)

may be dimensionless *e.g.* an aperture efficiency (%)

may be partially dimensionless *e.g.* a radiance ($W.m^{-2}.sr^{-1}$)

Method:

A/ elaboration of a topology:

First axis: the 7 components of the SI system (NC)

Second axis: an axis of degenerescence (SC)

Physical quantities: *goals, facts, static view, non-static view*

Define two categories whose objects are monoids:

QT (Quantity Type): a typename & arrow pointing to its topological space
$\Longrightarrow$ Kleisli category
Ex.: typename = Speed $\Longrightarrow$ QT<Speed>

PQ (Physical Quantity): a product of categories,

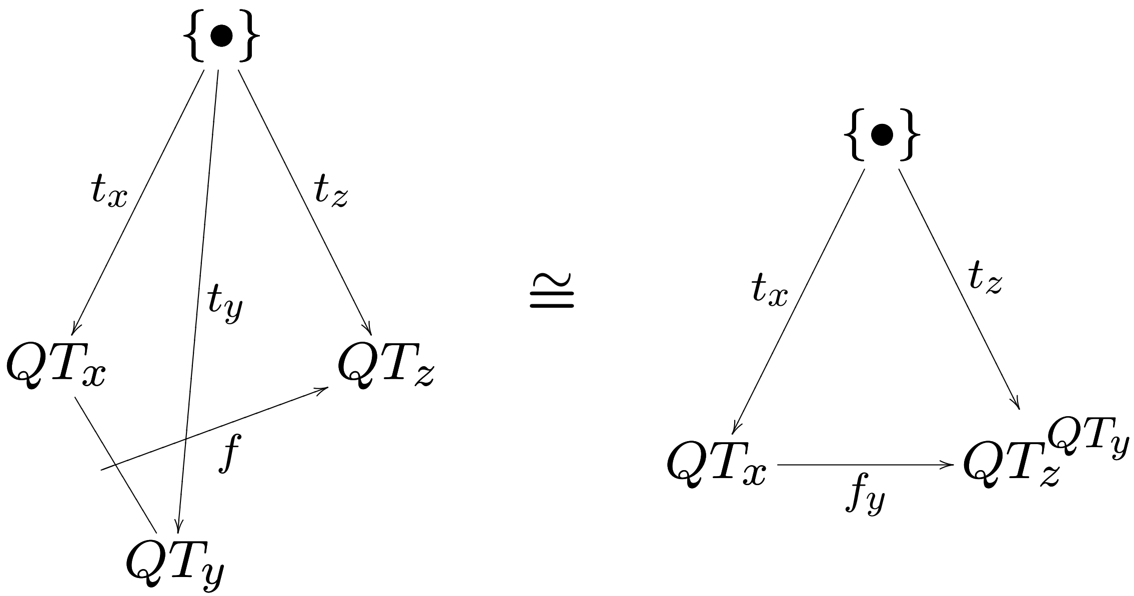
$$\mathbf{PQ} = \mathbf{QV} \times_{units} \mathbf{QT}$$

They are monoids on the addition because

$$\text{QT<Speed>} = \text{QT<Speed>} \oplus \text{QT<Speed>}$$
$$\text{PQ<Speed>} = \text{PQ<Speed>} + \text{PQ<Speed>}$$

Physical quantities: *goals, facts, static view, non-static view*

Define the algebraic topology



QT<Speed> = QT<Length> ⊗ QT<InvTime>

They are the morphisms in **QT**.

Physical quantities: *goals, facts, static view, non-static view*

Define the product:

$$z_0 = \ x_0 + y_0 \qquad \forall x_0 \wedge \forall y_0 \text{ and, } \forall\ i \geq \ 1 \text{ the conditions:}$$

$$z_i = \begin{cases} x_i + y_i & if \quad x_i + y_i \neq 0 \\ x_i + y_i - x_{i-1} - y_{i-1} & if \quad x_i + y_i \neq 0 \ \wedge \ x_{i-1} + y_{i-1} = x_i + y_i \\ x_i + y_i + x_{i-1} & if \quad x_i + y_i = 0 \ \wedge \ x_{i-1} + y_{i-1} \neq x_i + y_i \end{cases}$$

Physical quantities: *goals, facts, static view, non-static view*

Properties of this product:

Let $X(x_0, x_1, ..., x_n)$, $Y(y_0, y_1, .., y_n)$ and $Z(z_0, z_1, ..., z_n)$. Then:

commutativity: $X \odot Y = Y \odot X$ iff $x_{i-1} + y_{i-1} = x_i + y_i$

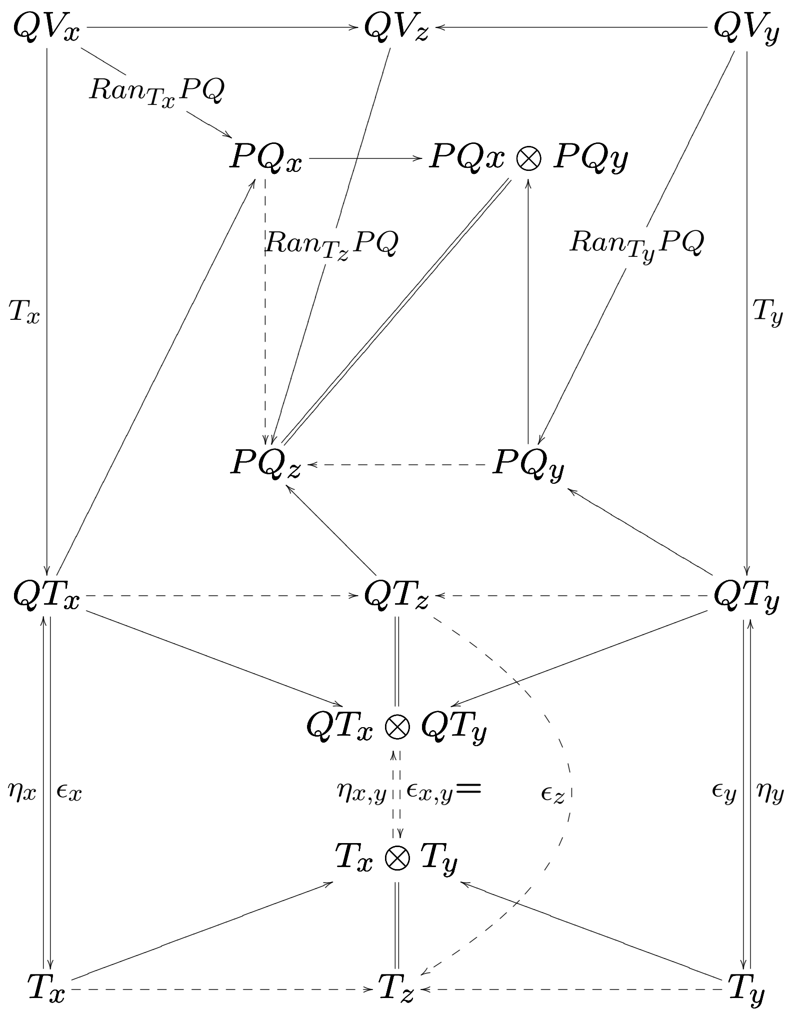
neutral element: let $Id_\odot = Id(0, 0, ...)$; $Id_\odot \odot X = X \odot Id_\odot = X$

associativity: $Z \odot (Y \odot X) = (Z \odot Y) \odot X$ iff $\begin{cases} \forall i \neq n_{sc}\ (x_i + y_i \neq 0, y_i + z_i \neq 0) \text{ or} \\ \forall\ (x_i = y_i = z_i) \end{cases}$

Therefore the associativity is not strict.

The quantity types partially dimensionless do not satisfy the coherence conditions of a monoidal category.

Physical quantities: *goals, facts, static view, non-static view*



value calculus at run-time

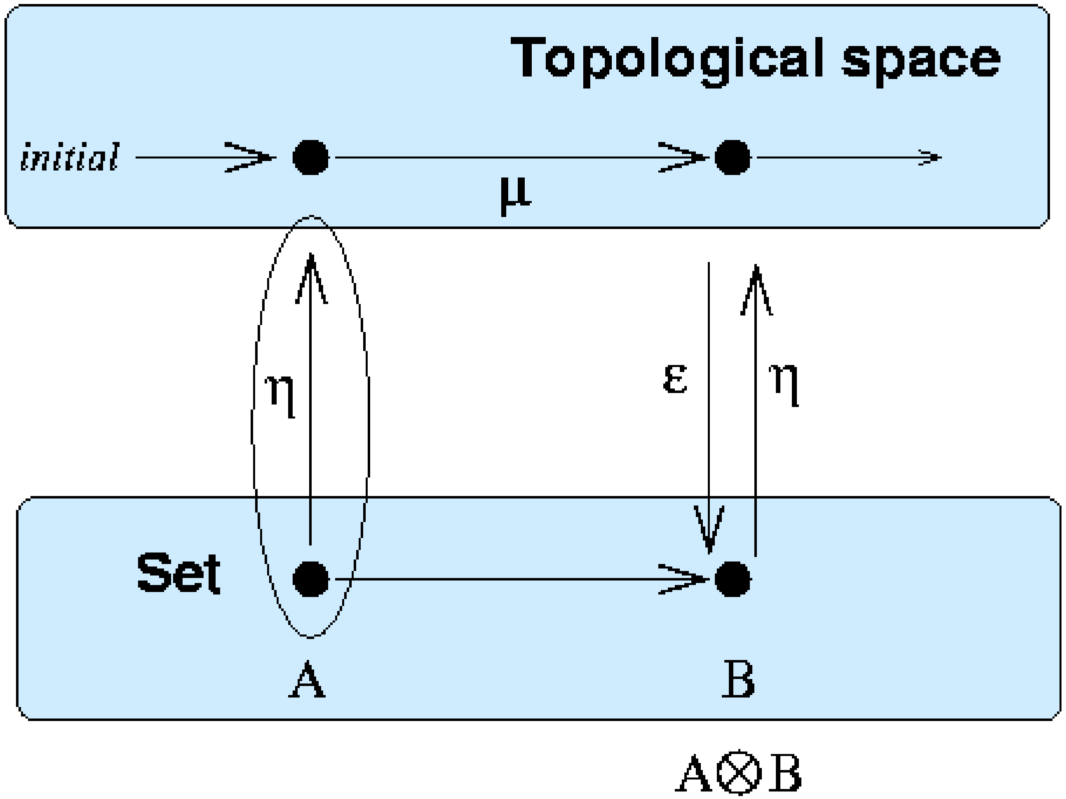
expressive equation in code

type calculus at compile-time

validation at compile time

language in physics

Physical quantities: *goals, facts, static view, non-static view*



Physical quantities: *algebra*, *homotopy equivalences*

Let

```
PQ<Length> len(100,km);
PQ<Time> time(3600);
```

The expression

```
PQ<Speed> v = len/time;
```

compiles and

```
cout<<"v="<<v.str("km/h")<<endl;
```

gives "v=100km/h" at run-time.

On the other hand

```
PQ<Acceleration> g=len/time;
```

would not compile but

```
PQ<Acceleration> g=len/time/time;
```

would.

Physical quantities: *algebra, homotopy equivalences*

In case of homotopy, to pass from one fiber to an other looks like this:

```
PQ<Pressure> p(0.5,atm);
PQ<EnergyDensity> u(Epi<Pressure>(p));
```

On the other hand

```
PQ<RadiantEnergyDensity> ru(Epi<EnergyDensity>(p));
```

would not compile because RadiantEnergyDensity and EnergyDensity are not an epi-phenomenon.
Being only an equivalence the coherent expression is:

```
PQ<RadiantEnergyDensity> ru(Equi<EnergyDensity>(p));
```

Physical quantities: summary

- PQ is a functor category, a singleton. It is a pure abstraction.
- PQ is the set all the physical expressions
- PQ is an endomorphism
- PQ is a monad PQ(PQ()) = PQ(); $1_{PQ} \times$ PQ = PQ $\Longrightarrow \exists\lambda$ calculus
- PQ_T is a monoid, a constructible functor with polymorphic representation *monomorphism:* $Ran_T PQ$ *and its dual,* $Lan_T PQ$*, for polymorphism.*
- PQ_T is a cartesian closed category whose objects are physical quantity states and the morphisms tensor products.
- PQ is monadic (T-algebra) $\Longrightarrow$ type-safe
- PQ has inductive cones

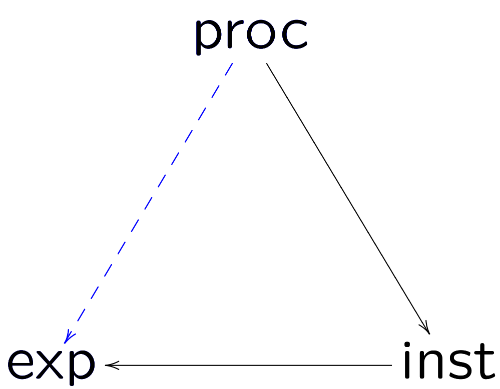
MS topology: <u>state machine</u>

- modeling a field of knowledge
  - any physical experiment

- the field turned into abstraction

- application in context: radio-astronomy
  - aperture synthesis interferometry in radio-astronomy
  - the instrument

State machine: a controlled system

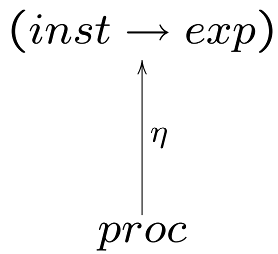


A commutative triangle:

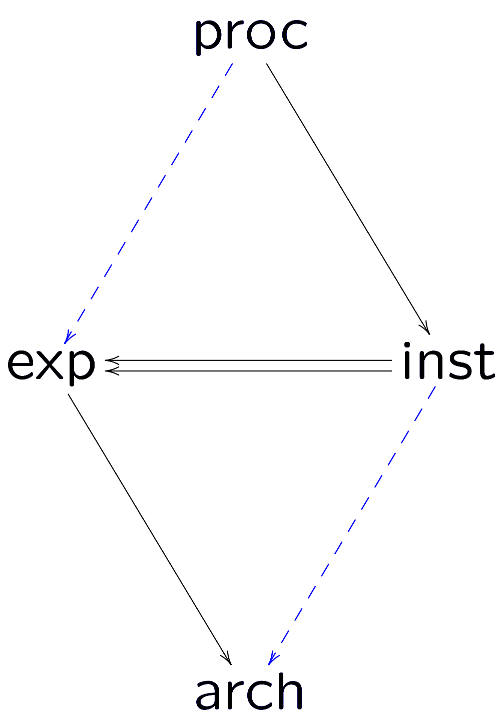
a pair of composable morphisms

$\Rightarrow$ *a 2-simplex*

with transitive closure (dashed arrow)



State machine: with a memory (persistence) (continued)



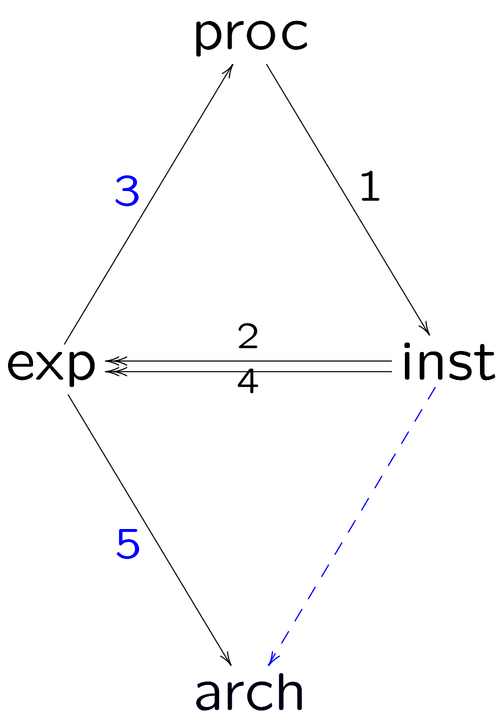
Two commutative triangles:

2 pairs of composable morphisms

$\Rightarrow$ *a simplicial 2-complex*

with transitive closures

$$\begin{array}{c} (inst \longrightarrow exp) \\ \eta \uparrow \; \downarrow \epsilon \\ (proc \longrightarrow arch) \end{array}$$

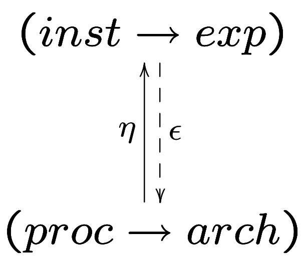


A complex 2-simplicial:

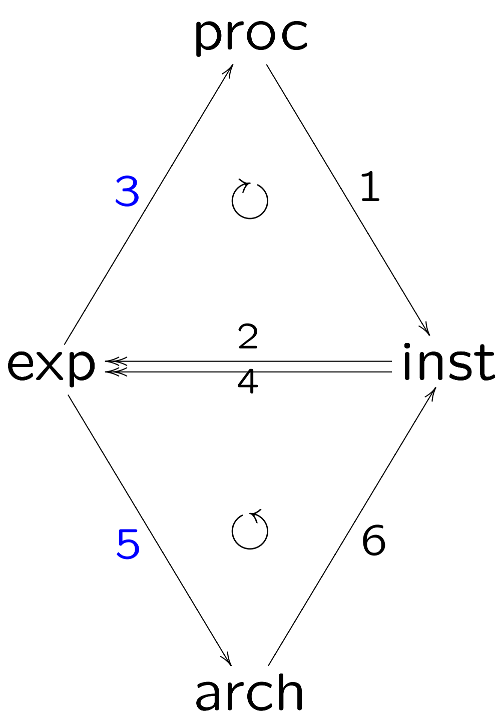
a composable morphism to calibrate[3] *the instrument*
⇒ *a simplicial 2-complex to measure*[5]

a path: and ordered sequence of 1-simplex with transitive closure

a functional system proceeding with comparisons
⇒ ∃ *a reference*
⇒ ∃ *a procedure* μ_{calib}



State machine: an automated system (continued)



A complex 2-simplicial:

a composable morphism to calibrate [3] *the instrument*
$\Rightarrow$ *a complex 2-simplicial to measure* [5]

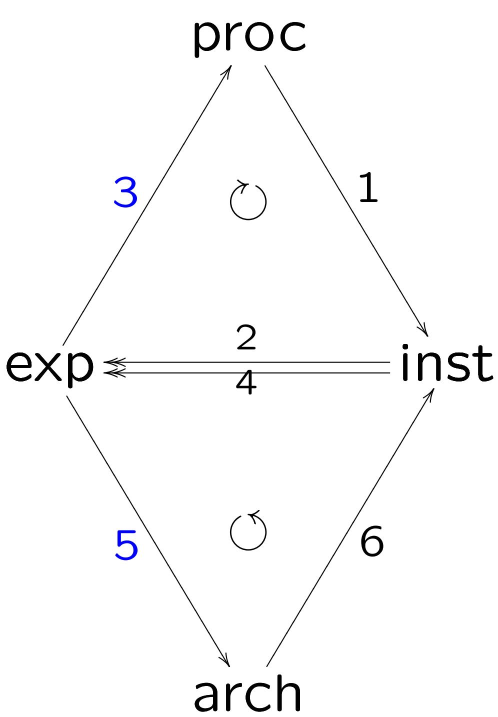
a path: and ordered sequence of 1-simplex
a closed set

A pair of states
a context (calibration aware) instrument type
$\exists$ *context sensitive types (colimits $\rightarrow \bullet \leftarrow$):*

$$\vdash inst \mapsto exp \rightarrow measurement$$

$$\begin{array}{c} (inst \rightarrow^{\circlearrowleft} exp) \\ \eta \uparrow \downarrow \epsilon \\ (proc \rightarrow arch) \end{array}$$

State machine: algebraic structure of automated systems (continued)



Let write:
($name, \rightarrow$*), the Kleisli-like objects. as bras:* $|name>$ *and*
($\leftarrow, name$*) as kets:* $<name|$

The path of the automaton corresponds to

$$|proc >< inst|.|arch >< exp|$$

The automation is a commutation between two states. We postulate that

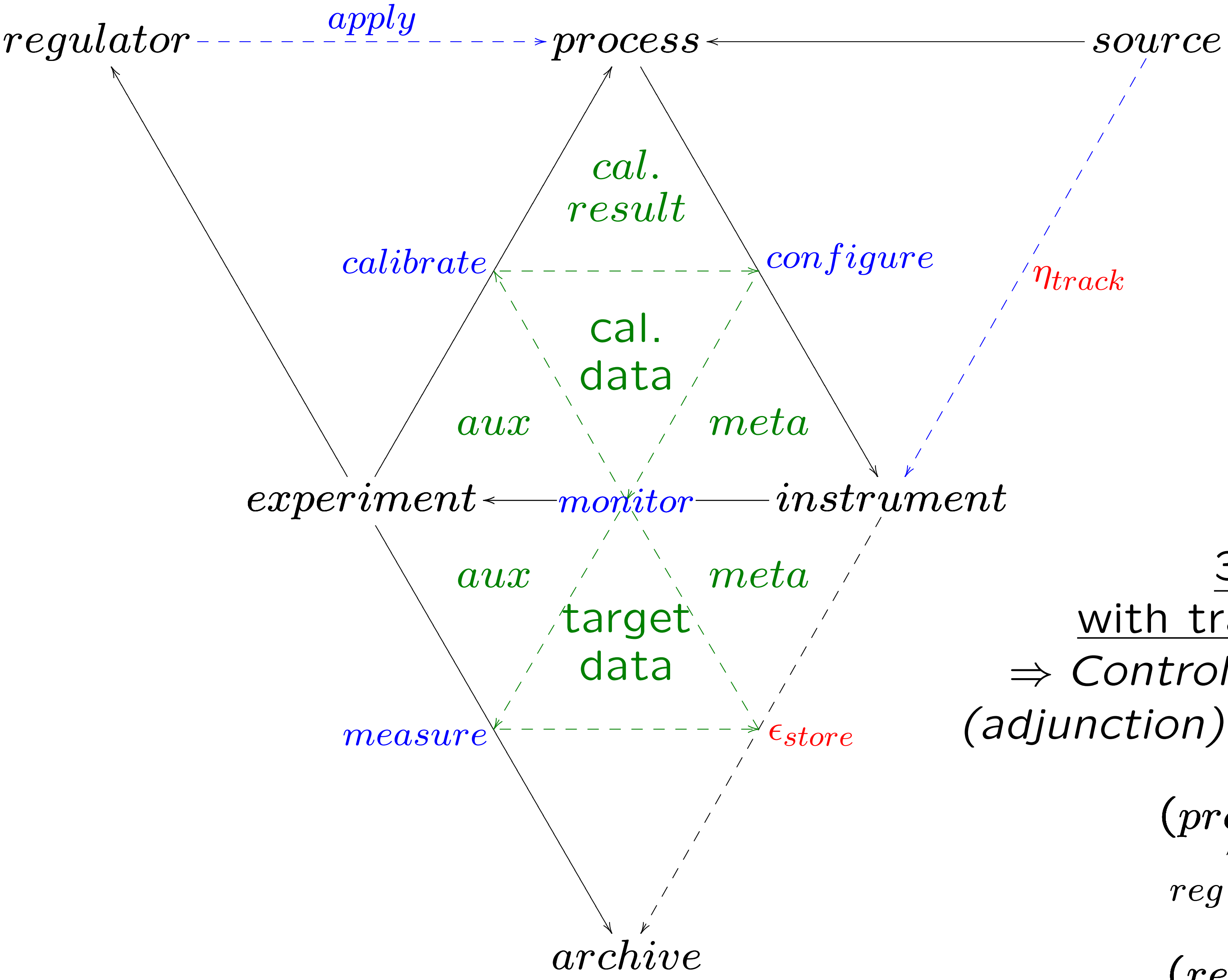
$$|proc >< inst|.|arch >< exp| = |proc >< exp|. < inst|arch >$$

Therfore there are:

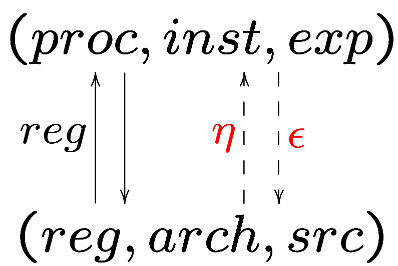
- *an* **action***: measure is the operator* $|proc >< exp|$ *and*
- **states***, measurement & instrument* $< inst|arch >$*, a duality.*

This is the Rota algebra.

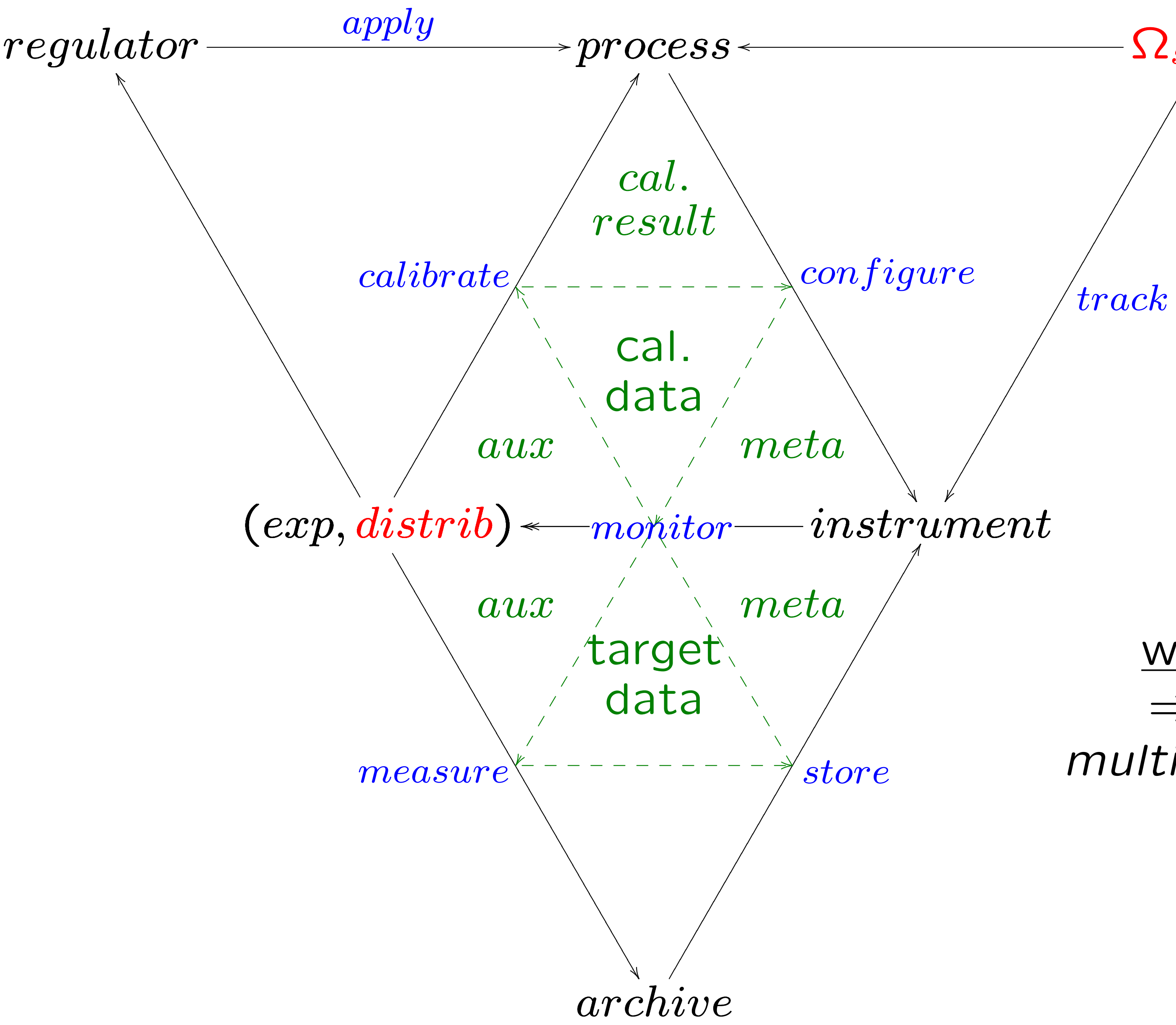
State machine: system partly automated (continued)



3-simplex
with transitive closure
⇒ *Control command system*
(adjunction) with an automaton:



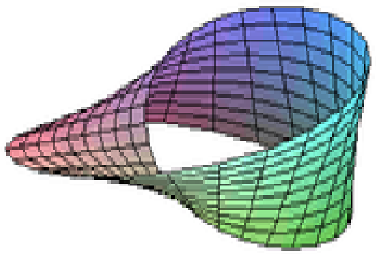
State machine: system partly automated (continued)



3-simplex
with internal Möbius
⇒ *Strict automaton*
multi-source ⇒ *distribution:*

$$(proc, inst, exp) \underset{reg}{\rightleftarrows} (reg, arch, src)$$

State machine: Möbius strip (continued)



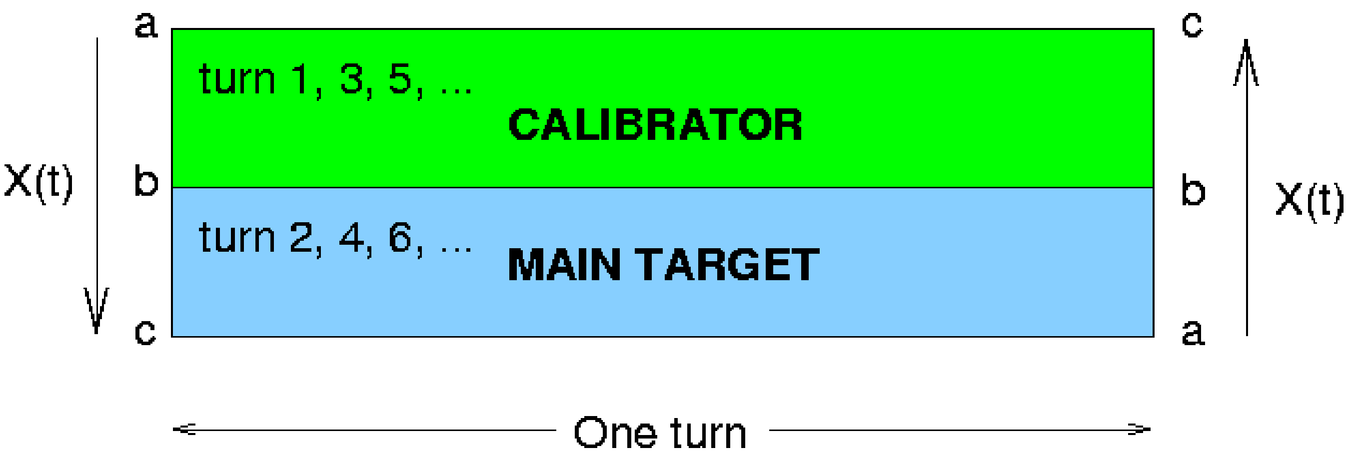
Common vertex: b=(proc,inst,exp/distrib)

Common path: ab=(reg,arch,src)

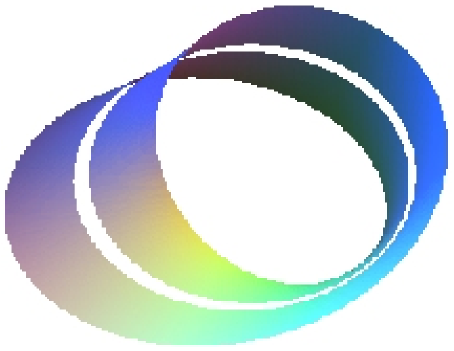
The distributor: a switch (sequential ON/OFF)

a measurement relative to a reference calibrator:

turn 1 cal, turn 2 tar/cal, turn 3 cal, turn 4 tar/cal ...



State machine: Möbius strip (continued)



Common vertex: b=(proc,inst,exp/distrib)

Common path: ab=(reg,arch,src)

The distribution: 2 fibers in concurrency
splitting the strip in 2 parts

State machine: Möbius strip (continued)

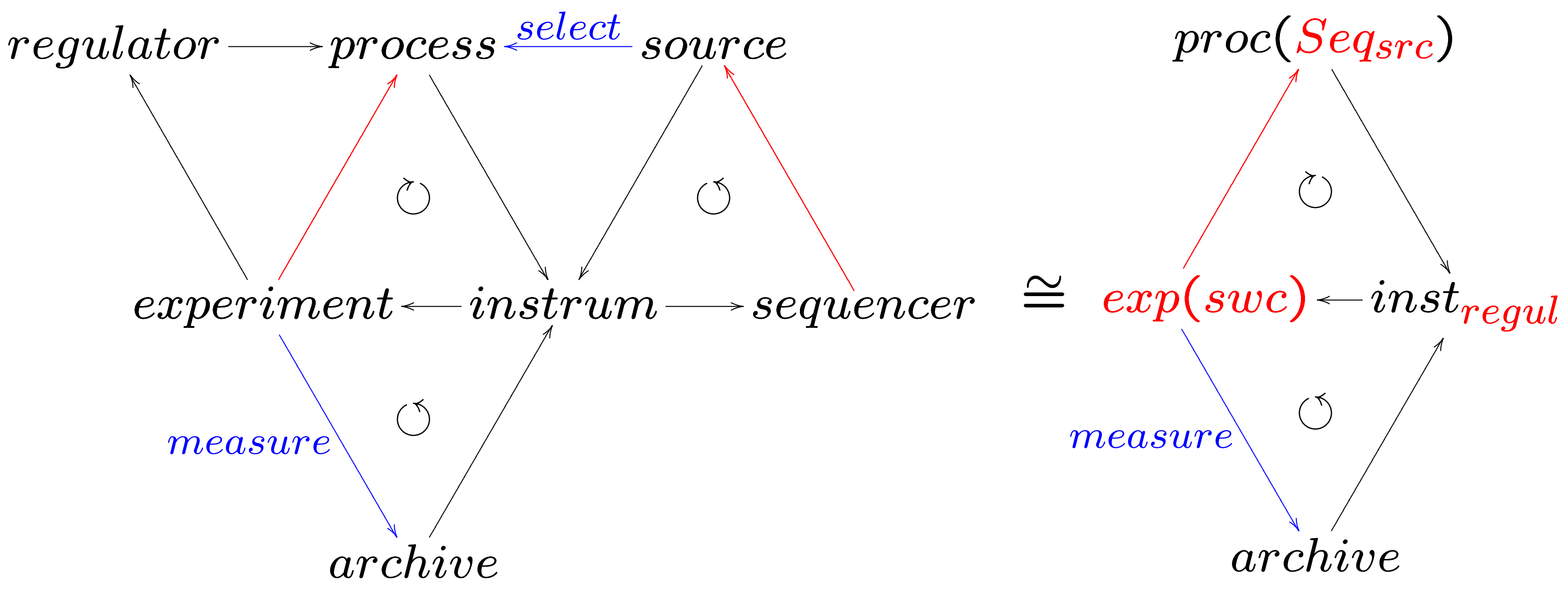
Common vertex: b = (proc,inst,exp/distrib)

↑↓ regularization

Common path: ac = (reg,arch,src)

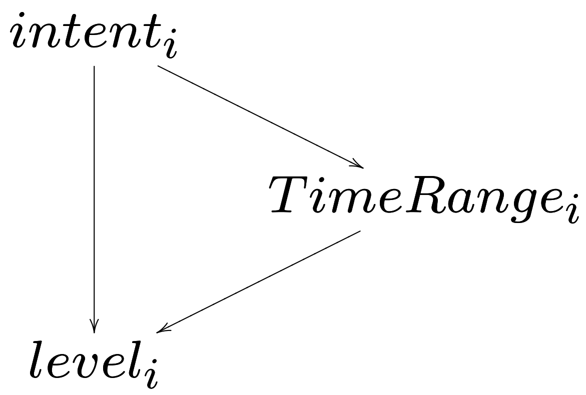
The distributor: | a switch iff X(t) or | two fibers in concurrency

State machine: sequential activity (continued)



System proceeding by comparison with references
Sources sequenced to compensate the temporal unstabilities (drifts) of the system
$\Rightarrow$ *A 2-colors Möbius strip (commonly regulated), switching colors vs time*
$\Rightarrow$ *An automaton with an embarqued program and experimental procedures*

State machine: the sequencer (continued)



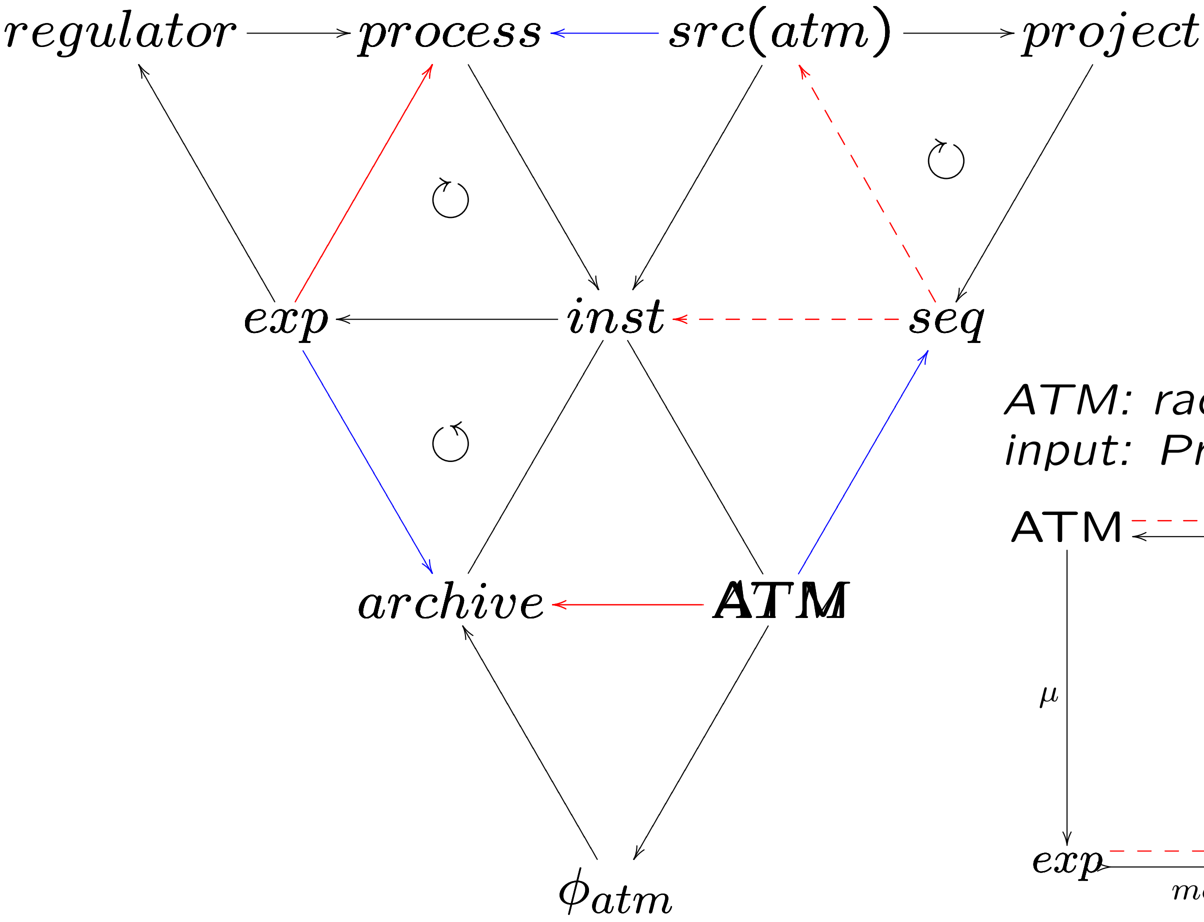
$$\forall i \mid j = i{+}1; TR_i = \sum_{n \in level_j} TR_n$$

- on a hierarchy of levels
- in each level the time axis splitted in ranges of times
- levels are mapped to enumerated intents
- system reconfiguration not permitted beyond some level

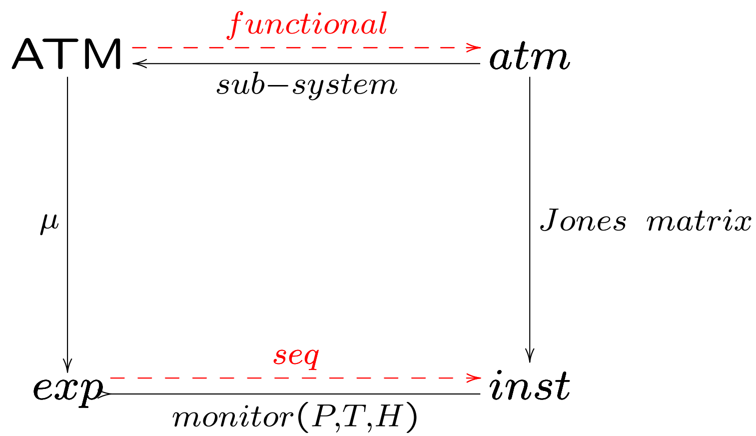
An experiment is controlled by functional intents (recursions)

State machine: regulator example (continued)

Monitoring atmospheric phase fluctations:

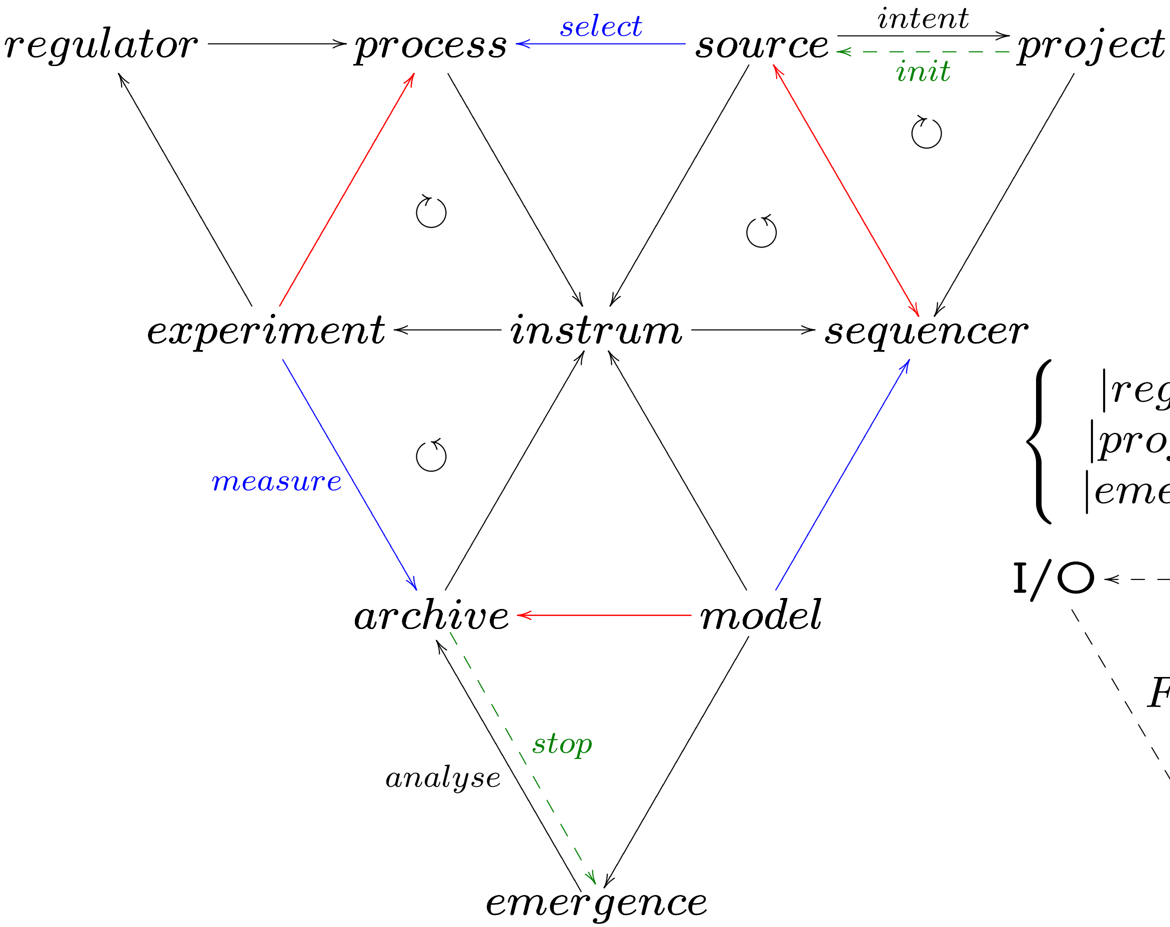


ATM: radiative transfer proc.
input: Pression, Temp., Hum.

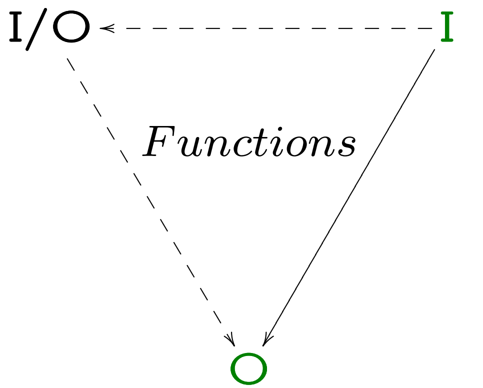


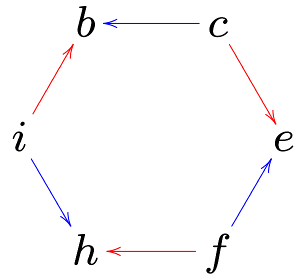
State machine: the whole topology (continued)

Execution of a project:



$$\left\{ \begin{array}{ccc} |regu >< proj| & . & < proc|src > \\ |proj >< emer| & . & < sequ|mod > \\ |emer >< regu| & . & < arch|exp > \end{array} \right.$$



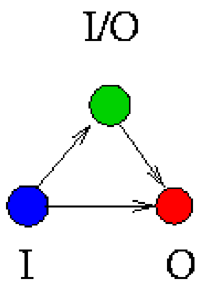


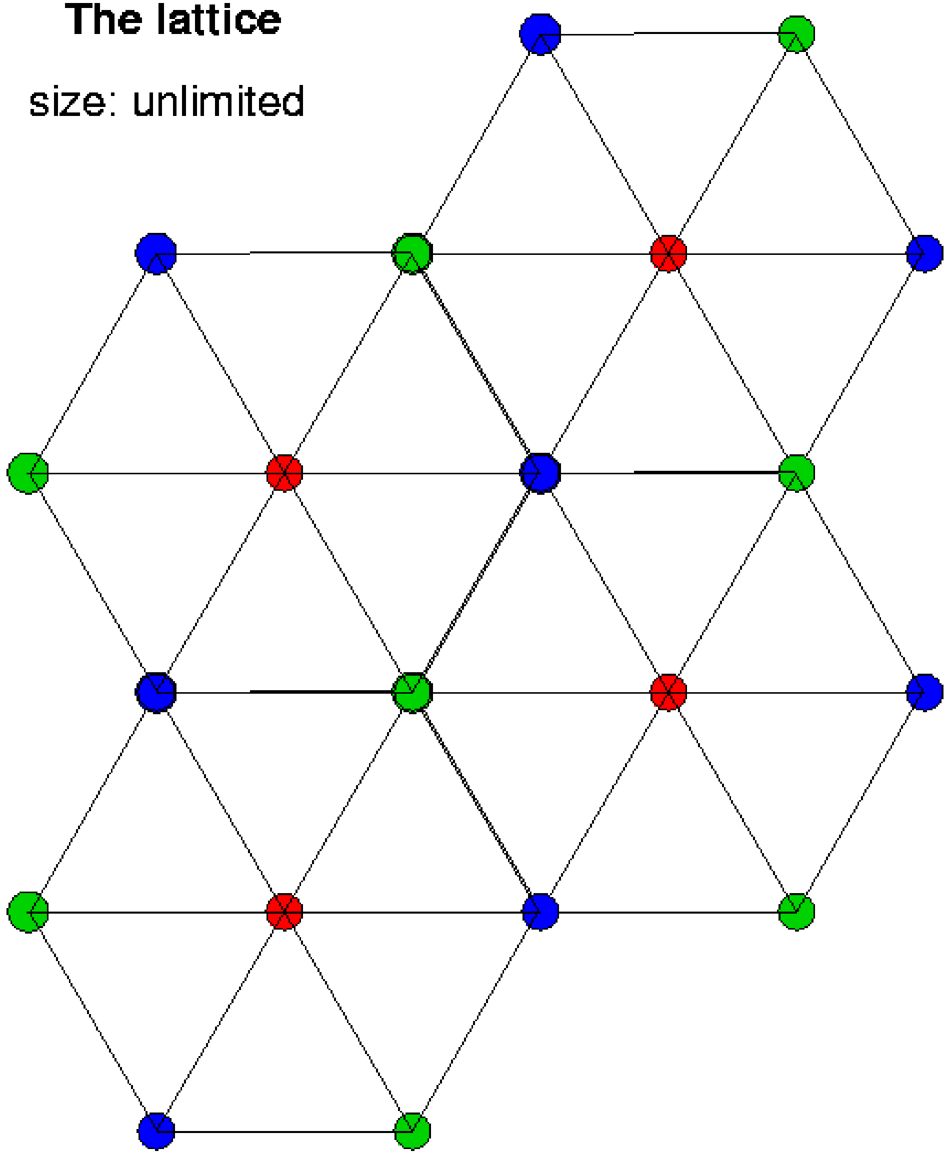
States

State machine: a lattice (continued)

The lattice

size: unlimited





Physical Experiment: a monad for $\bigoplus$: (continued)

Let PE a physical experiment:

PE is a unit $\Rightarrow \exists$ compositions at all levels in the hierarchies:

$PE = PE_{\triangle t_1} \bigoplus PE_{\triangle t_2} \bigoplus \ldots$
$PE = PE_{src_1} \bigoplus PE_{src_2} \bigoplus \ldots$

$$PE = PE \oplus PE$$

PE is self-similar:

$$\forall i \in \{levels\} | i < j < k; PE = PE_i(PE_j(PE_k)) intent_i = \bigoplus_{i \in I} intent_i$$

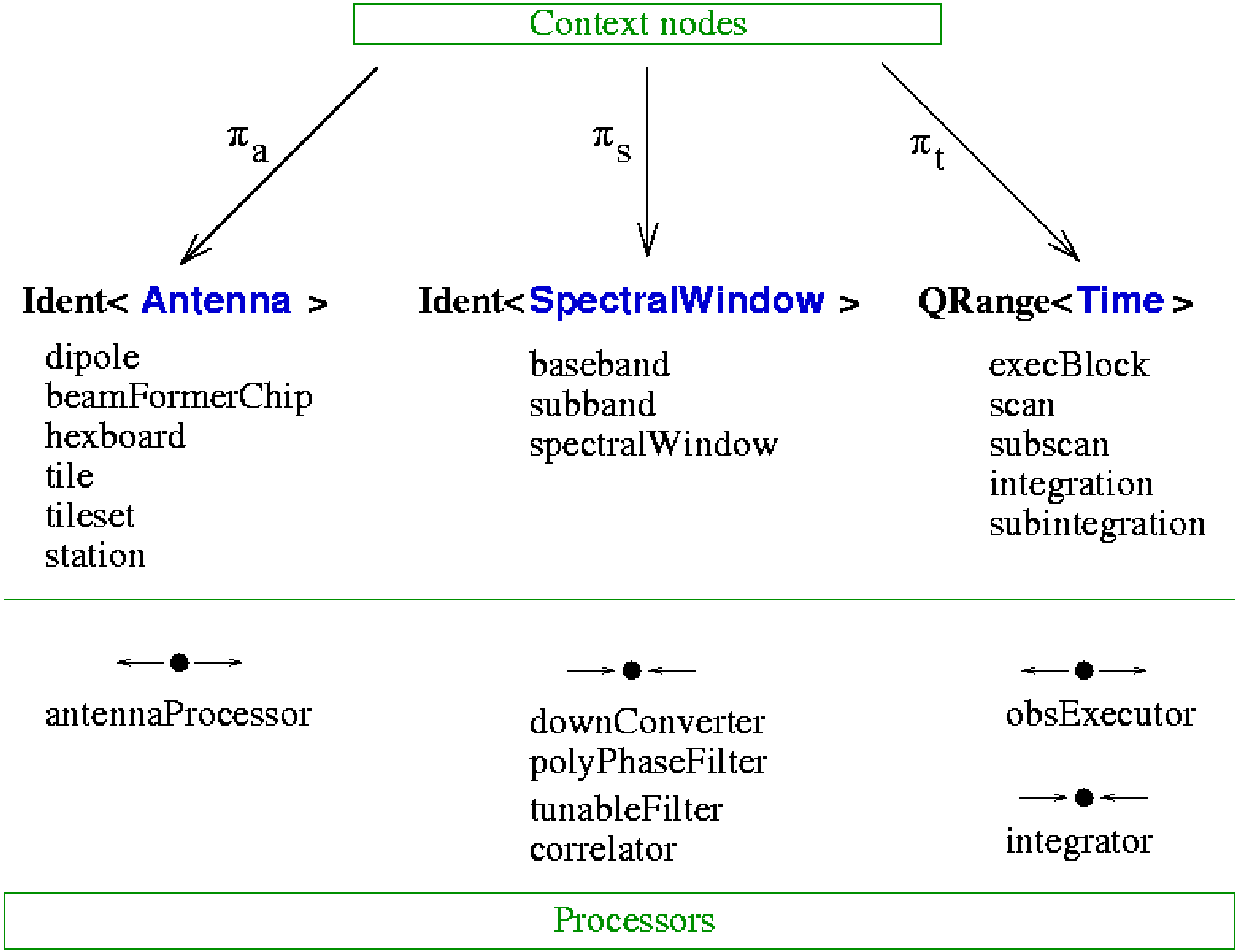
PE is reductible:

$$PE = calib(PE) PE = measure(PE)$$

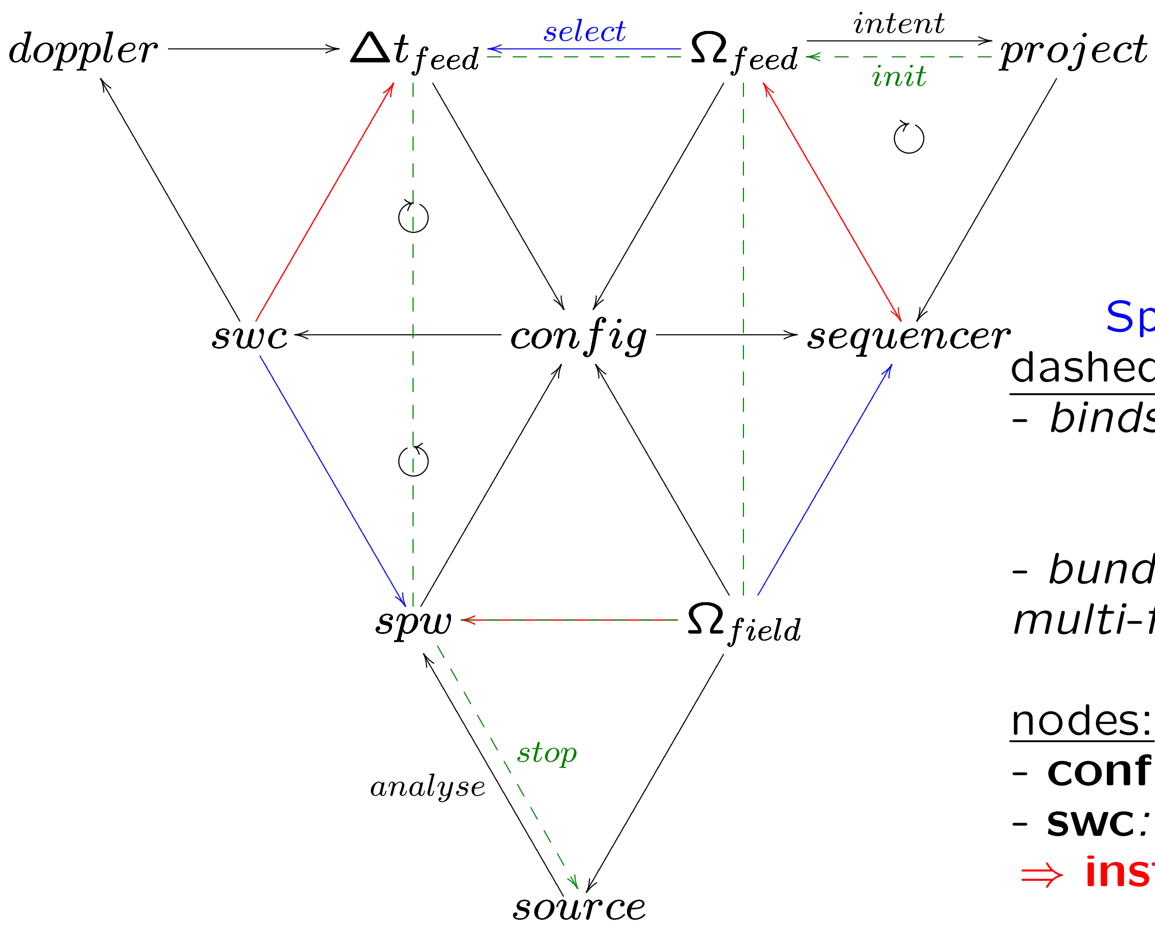
because $structure_{reduced} \in structure_{PE}$

Physical Experiment: instrumental axes (continued)

Topological space axis basis



State machine: the topology for a radio-telescope (continued)



Spectro-imager instruments:

dashed rectangle:

- *binds 2 triangles: duality ν, dir:*
  (Δt_{feed},feed,field)
  (Δt_{feed},spw,seq)
- *bundle of fibers for multi-beam and multi-frequency*

nodes:

- **config**: *lattice (sub-array) ident*
- **swc**: *commutator switch ν vs dir*

$\Rightarrow$ **instrument has same topology**

Pipeline: the structure of the reduced data (continued)

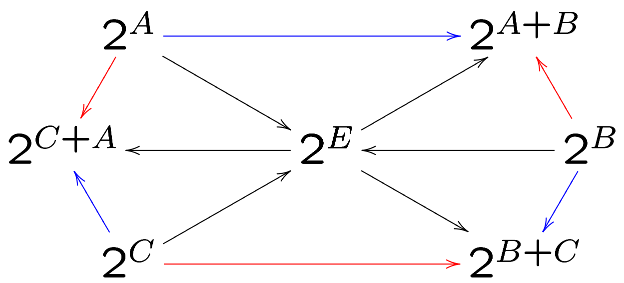
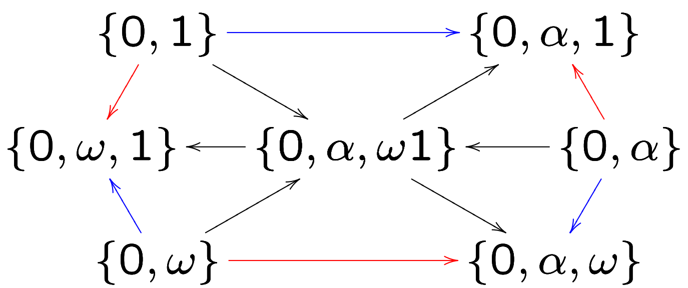
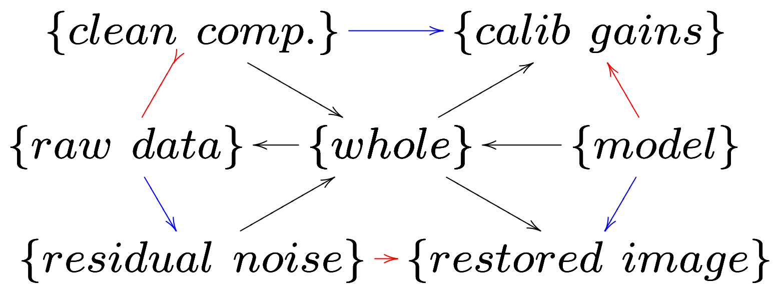


diagram in **Bool** of boolean algebra



Borromean object

ref. Guitart 2011



Structure reduced data

in aperture synthesis

Tripartition: $noise, signal, instrum_{calib}$

Cut-off boolean-based iterative algorithm:

$\Omega_{S/N} : 0 \leq \omega \leq 1 \equiv undetected \leq \omega \leq detected$

$\Omega_{cal} : 0 \leq \alpha \leq 1$

'restored image' is the last terminal object

Underlying topology: *conclusions*

A generic **single** topology with 3 tetradrons (2 sticked at any time) matches the structure required for:

1. the pre-observation activities
2. the on-line activity (controled subsystem)
3. the subsystems layout of a radio telescope
4. the structure of the reduced data

Proposition:

There is a formal theory
in the language of predicate logic.

Binary Relations: *cardinalities*, *types*, *structure*

Let κ the cardinality in a relation.

There are two foundamental quantifiers:

- the existential quantifer '$\exists$'
- the universal quantifier '$\forall$'

Let $\longrightarrow^{\kappa}$ be the statement $\forall x \in X \mid \mathcal{J}(\kappa)$
Let $^{\kappa}\longleftarrow$ be the statement $\forall y \in Y \mid \mathcal{J}(\kappa)$

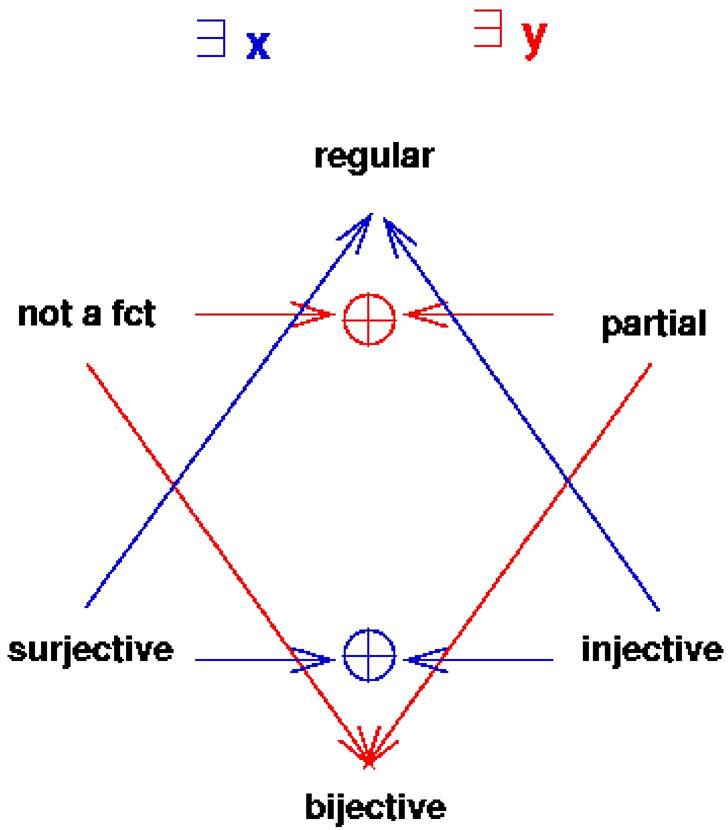
Note that $\longrightarrow^{\kappa}$ is equivalent to $^{1}\longrightarrow^{\kappa}$.

κ	Definition of $\mathcal{J}(\kappa)$	
1..*	$\exists :\subseteq Y$	$\in 2^Y$
1	$\exists y \in Y$	$\in Y$
0,1		$\in Y \vee \notin Y$
0	$y =_Y$	$\notin Y \wedge \in 2^Y$

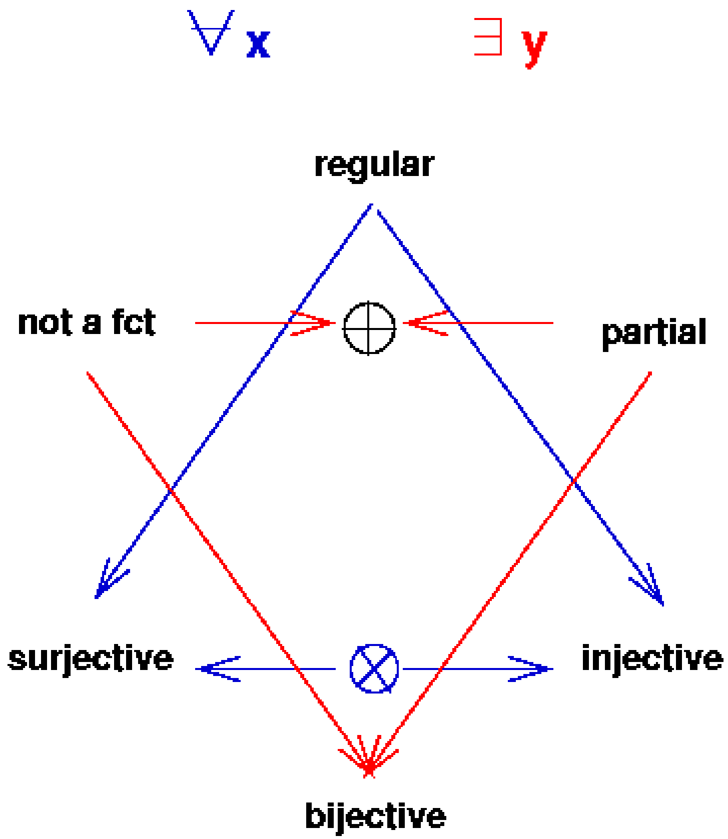
Binary Relations: *cardinalities, types, structure*

UML notation	Type of relation	Enoncé
$^1 \longrightarrow {}^1$	regular function	$\forall x, \exists\ 2^Y$
$^1 \longleftarrow {}^1$	reversible function	$\exists : Y$
$\longrightarrow^{1..*}$	not a function	$\forall x, \exists : Y$
$^1 \rightleftarrows {}^1$	symmetric relation	$\exists : 2^R$
$\longrightarrow^{0,1}$	partial function	$\forall x, \exists : Y \rightarrow \mathbf{2}$
$^{0,1} \longleftarrow$	injective function	$\forall y, \exists : X \rightarrow \mathbf{2}$
$\longrightarrow {}^1$	invertible function	$\exists : Y$
$^{1..*} \longleftarrow$	surjective function	$\exists X, \forall y$

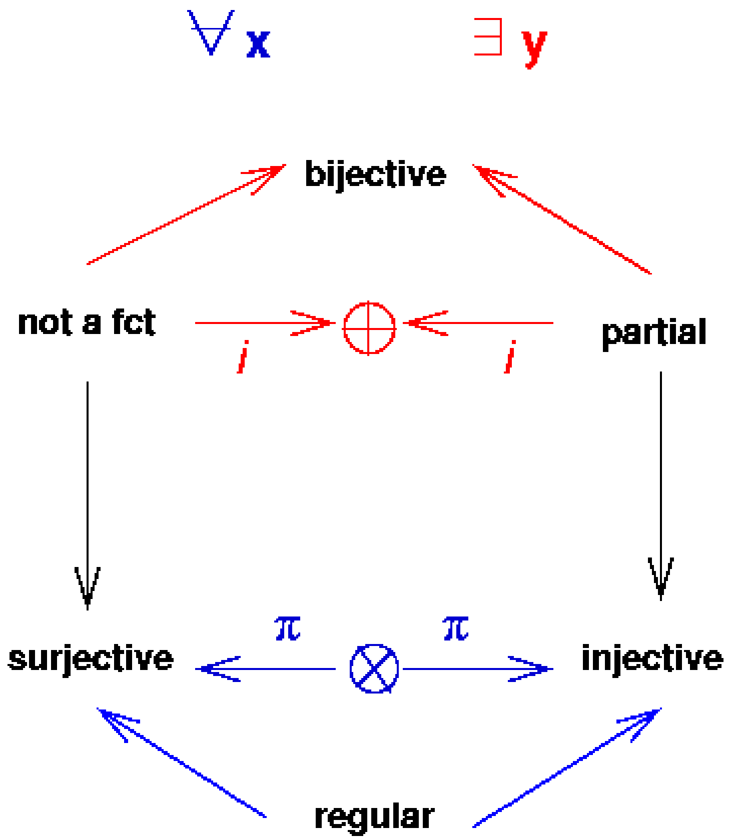
Binary Relations: *cardinalities, types, structure*



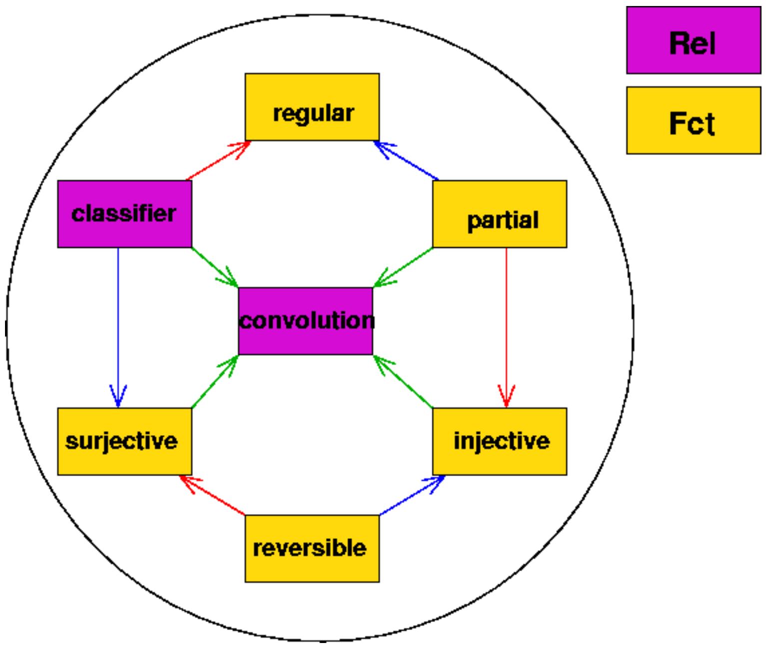
Binary Relations: *cardinalities, types, structure*



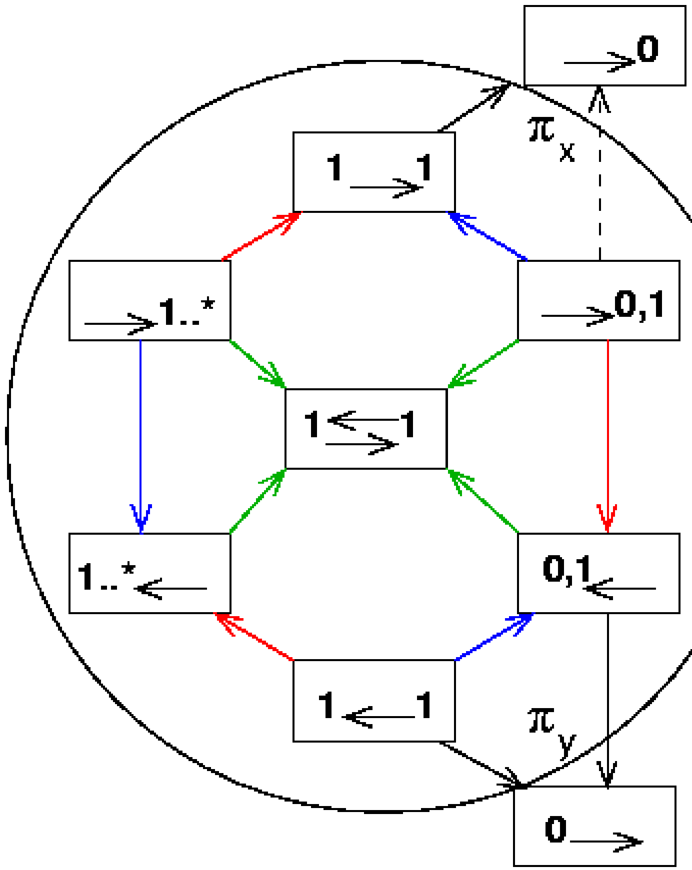
Binary Relations: *cardinalities, types, structure*



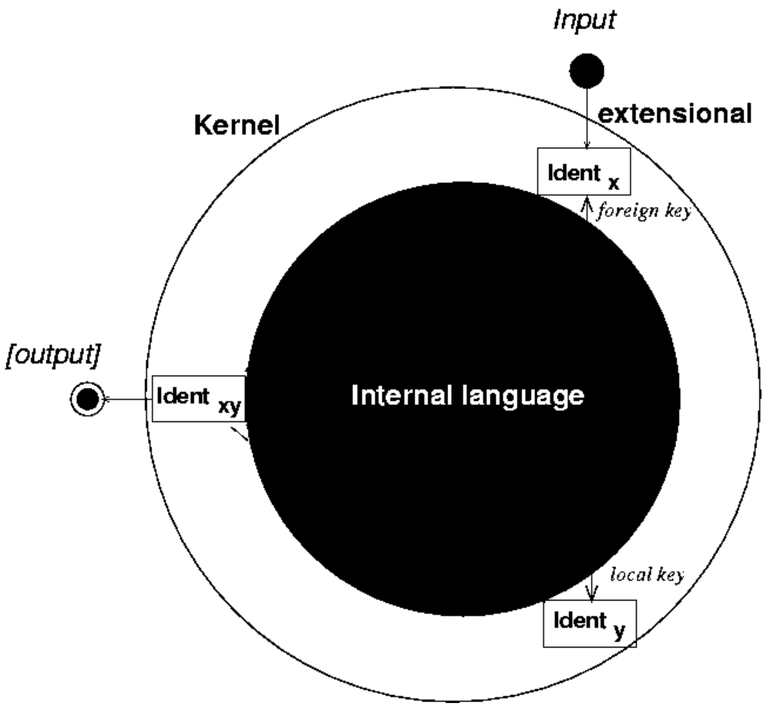
Binary Relations: *cardinalities, types, structure*



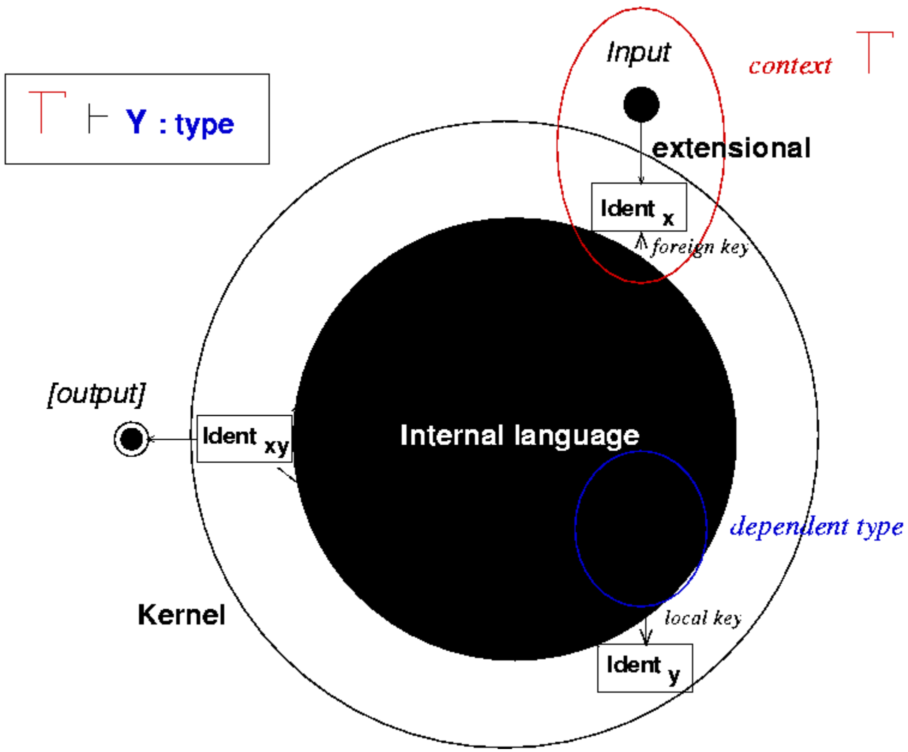
Binary Relations: *cardinalities, types, structure*



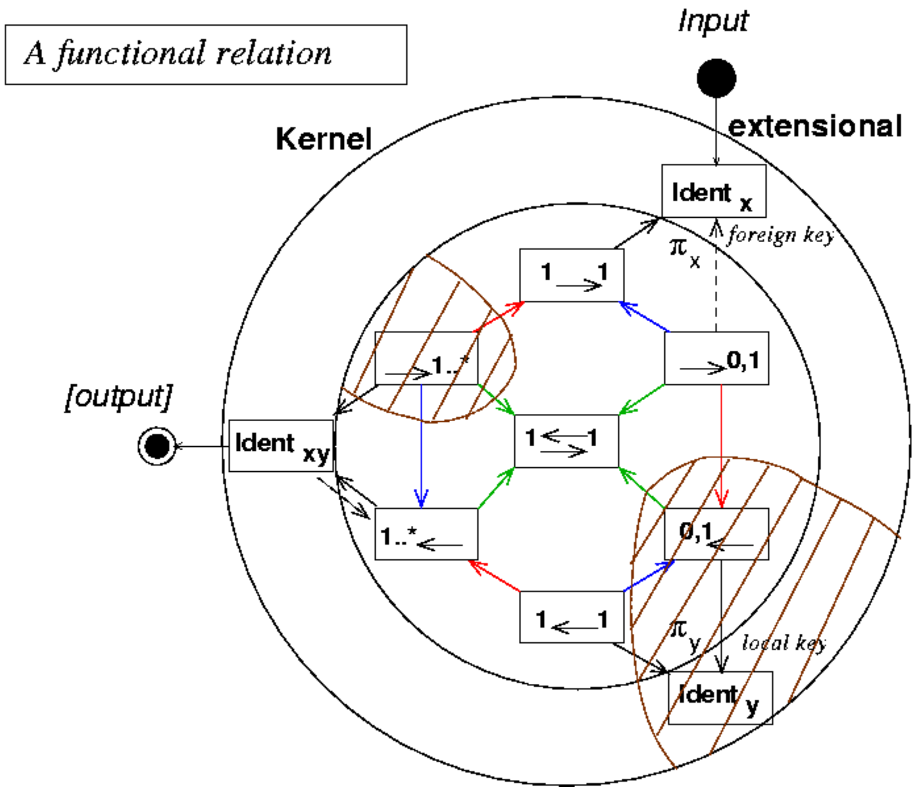
Binary Relations: *cardinalities, types, structure*



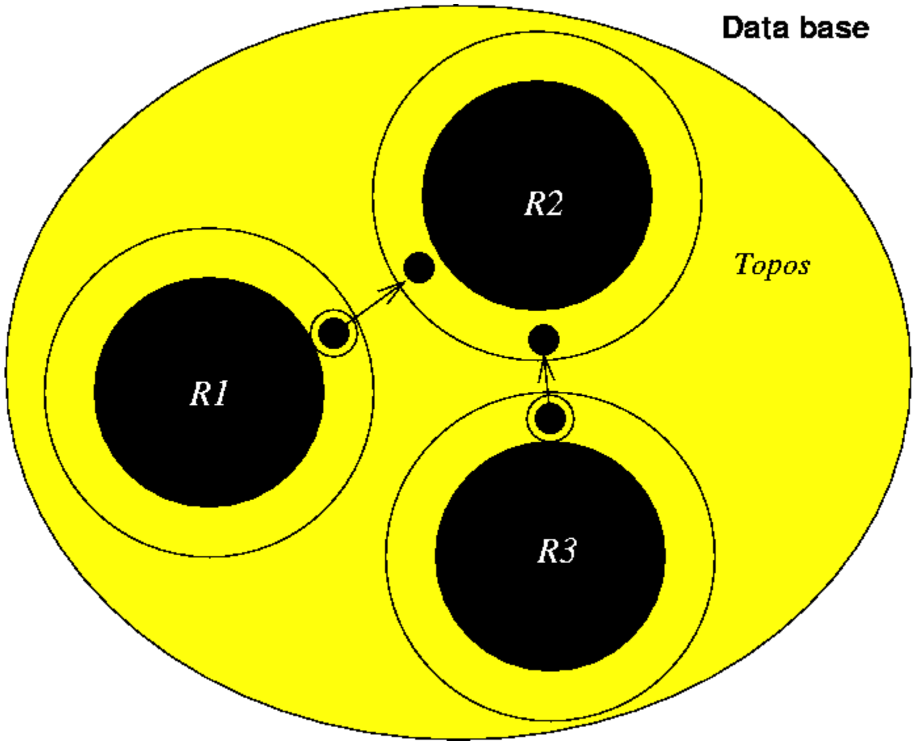
Binary Relations: *cardinalities, types, structure*



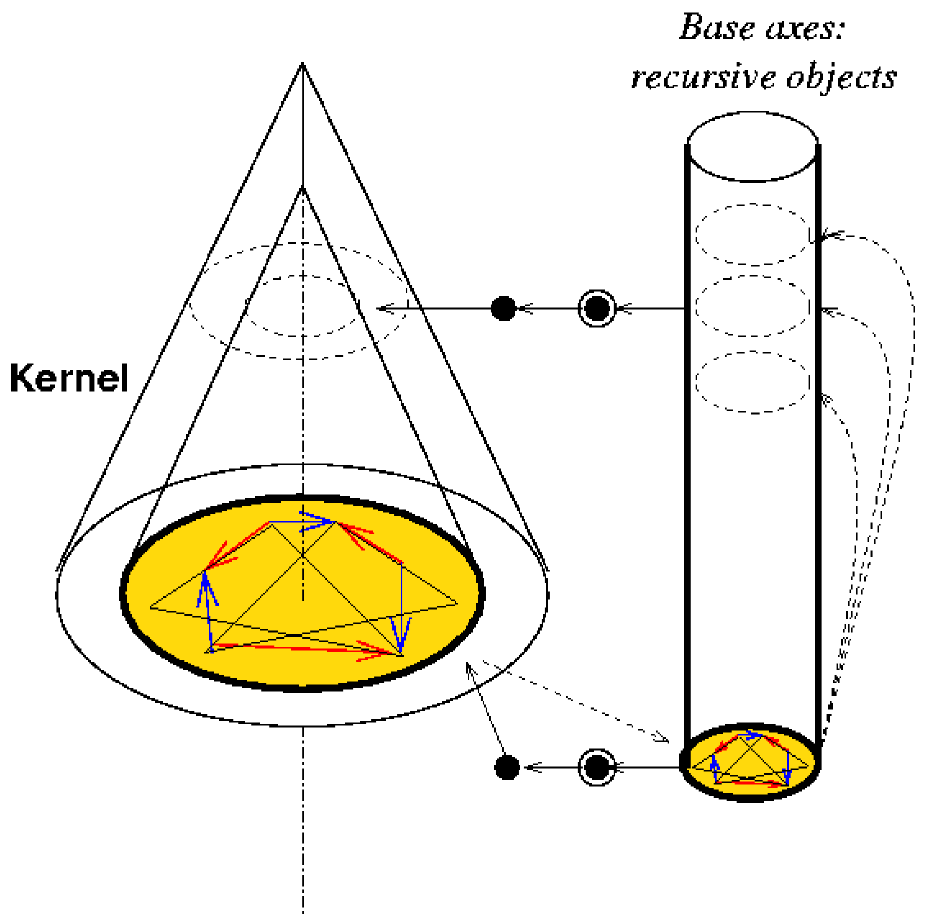
Binary Relations: *cardinalities, types, structure*



Relational DB: *molecular struct*, *multi-resolution*



Relational DB: *molecular struct, multi-resolution*



Relations: *partioning*

Defining a relation gives meaning to the kind of things to represent.

1. A list of attributes defines properties
2. Semantics is introduced by partioning this list
   e.g. 5 attributes allows 52 different partitions!
3. A partition is a product of its parts
4. A part is the co-product of its elements
5. A relation may be the composition of relations
6. A database is a set relations

Data model: *conclusions*

A fundamental generic structure has been found empirically from the logical discourse used to describe physical experiments. This structure is found to correspond also to the structure of the atoms chosen to reflect relations in the most general form.

1. The underlying space of the physical experiments has a borromean structure

2. The relational model has a borromean logic