Softlang Working Group
http://softlang.wikidot.com/

MegaL
Traceabiltiy Recovery

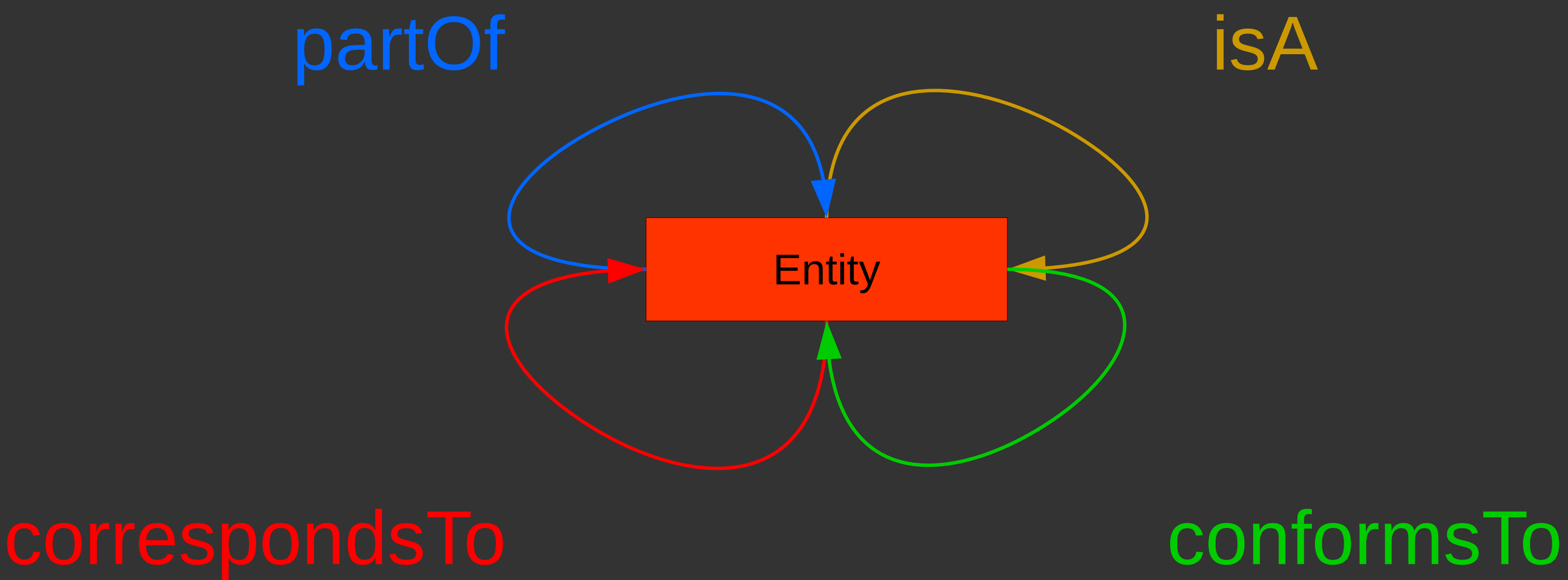
https://github.com/maxmeffert/megal-tr

Formal Foundations

Meeting 2016-03-17

University of Koblenz-Landau
Maximilian Meffert

Prelude



partOf

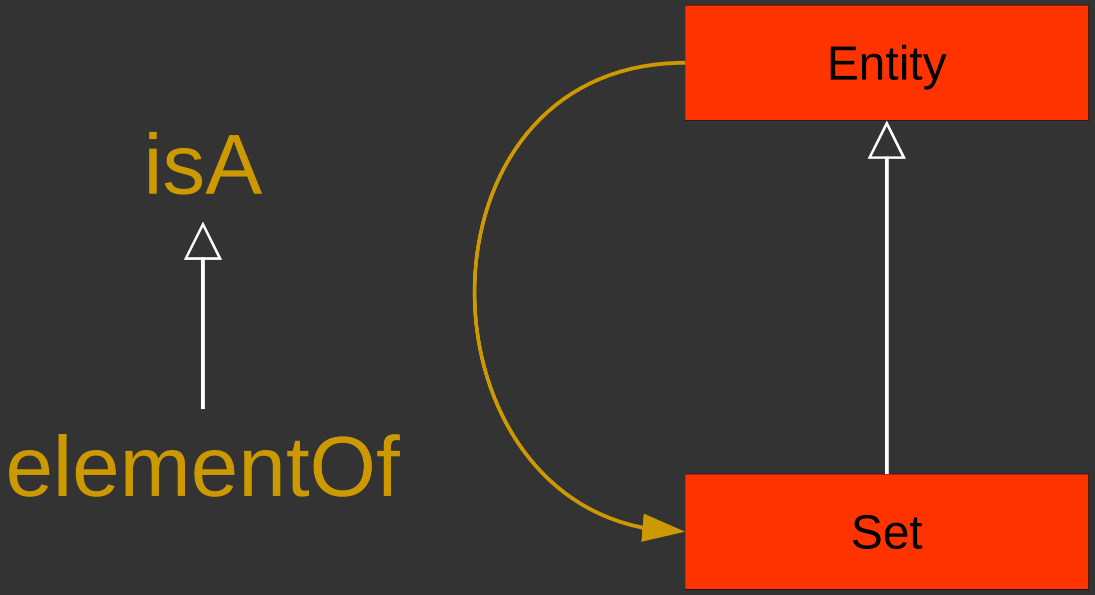
isA

Entity

correspondsTo

conformsTo

Prelude

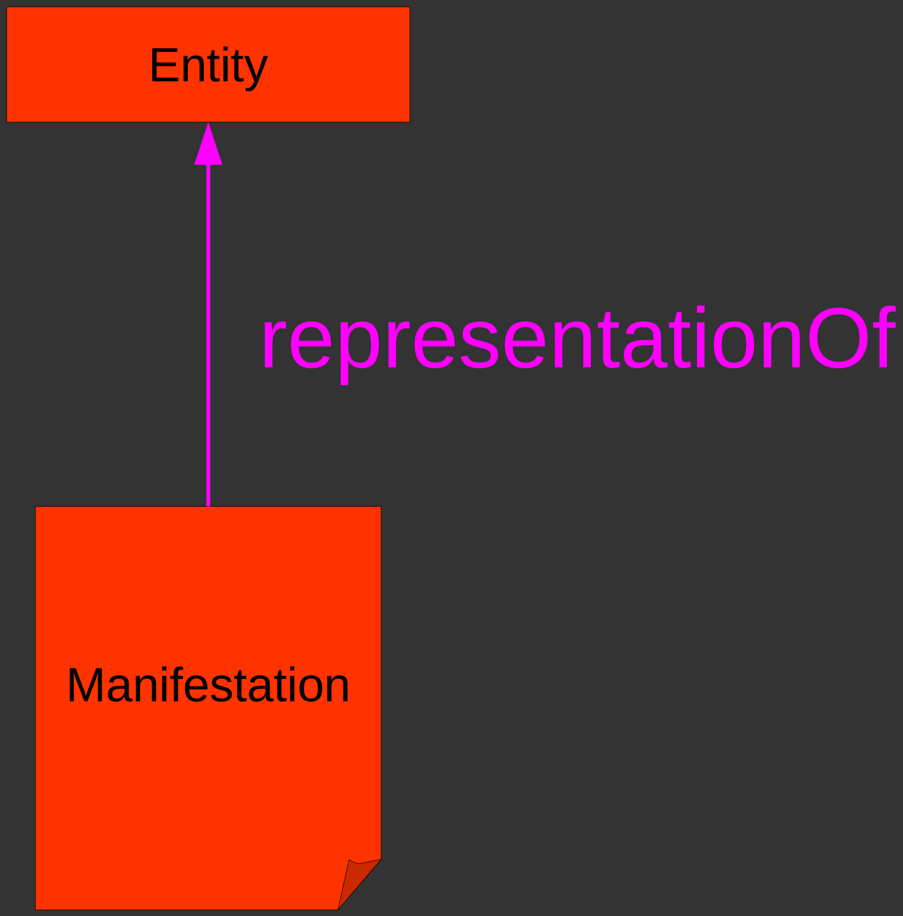


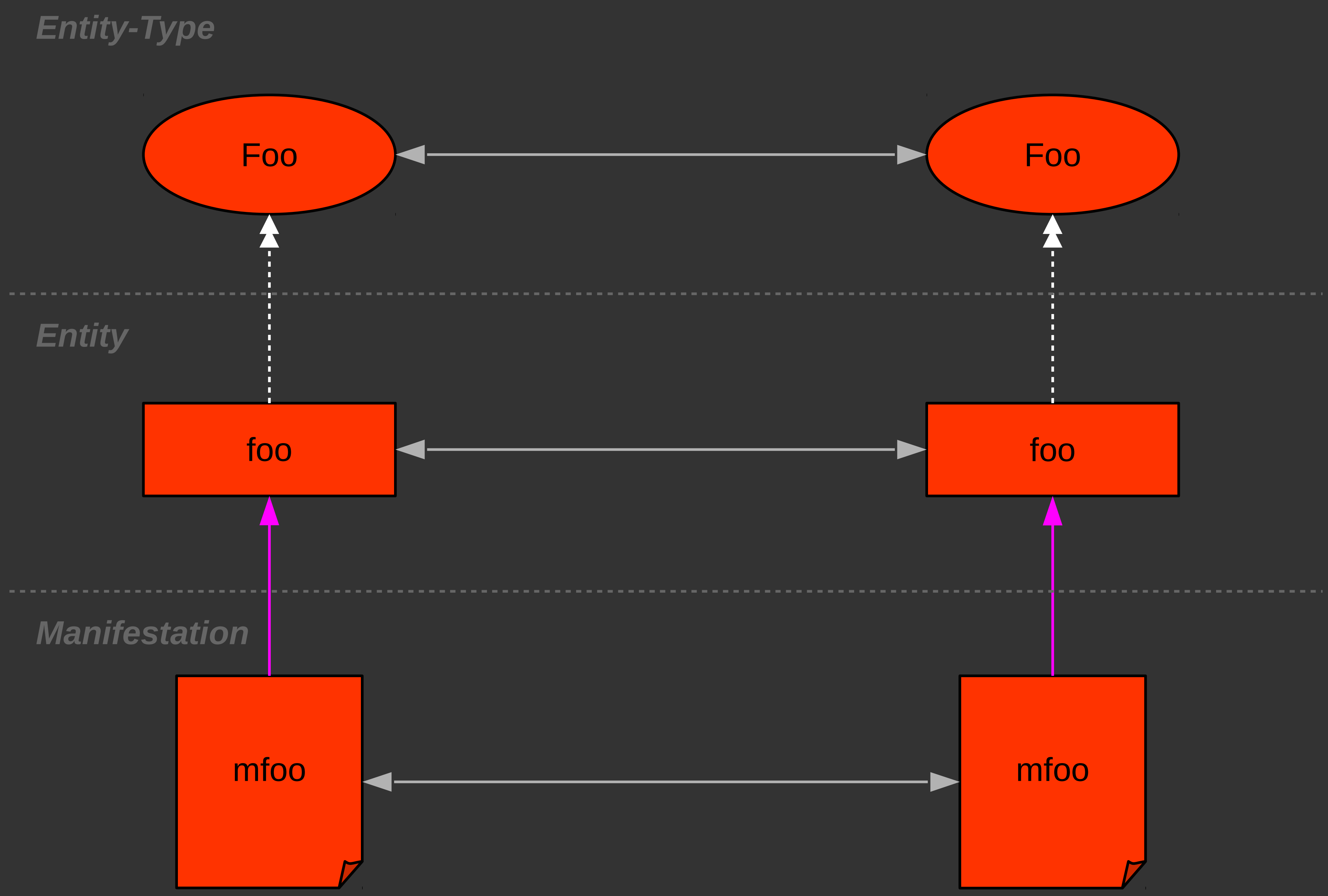
„Signum ∊ significat est. Ita a ∊ b legitur a est quoddam b“

„The Symbol ∊ means is. Thus a ∊ b is read ***a is a b***“

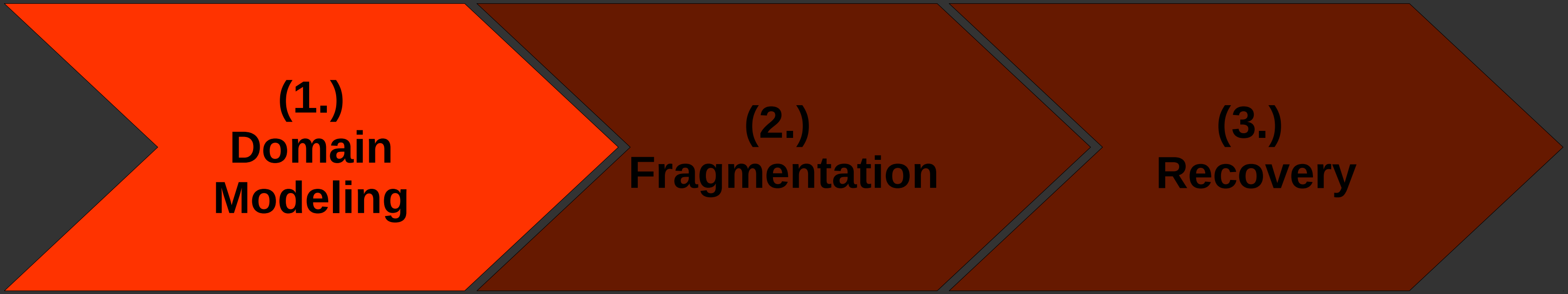
(Giuseppe Peano: Arithmetices principia nova methodo exposita, 1889, S. X)

Prelude



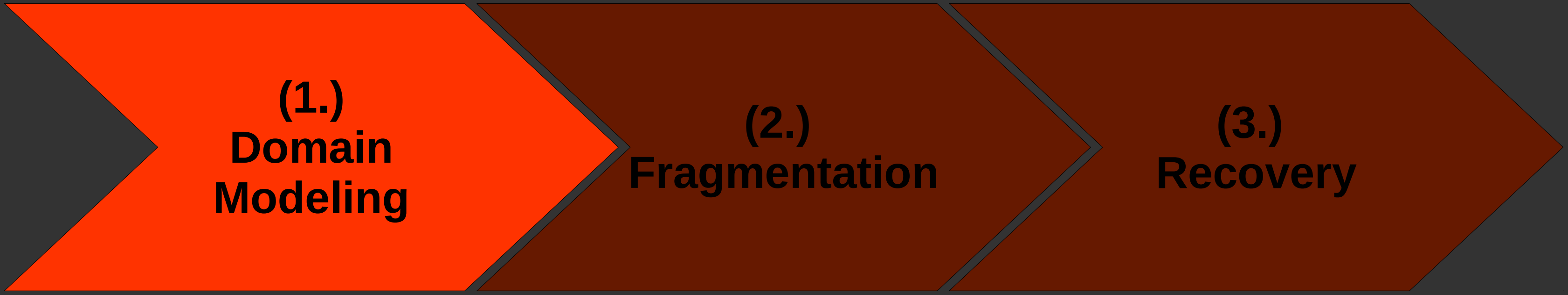

Entity-Type
Foo
Foo
Entity
foo
foo
Manifestation
mfoo
mfoo

Motivation



A domain model is a collection of (axiomatic) *statements* over entity-types and entities.

Motivation

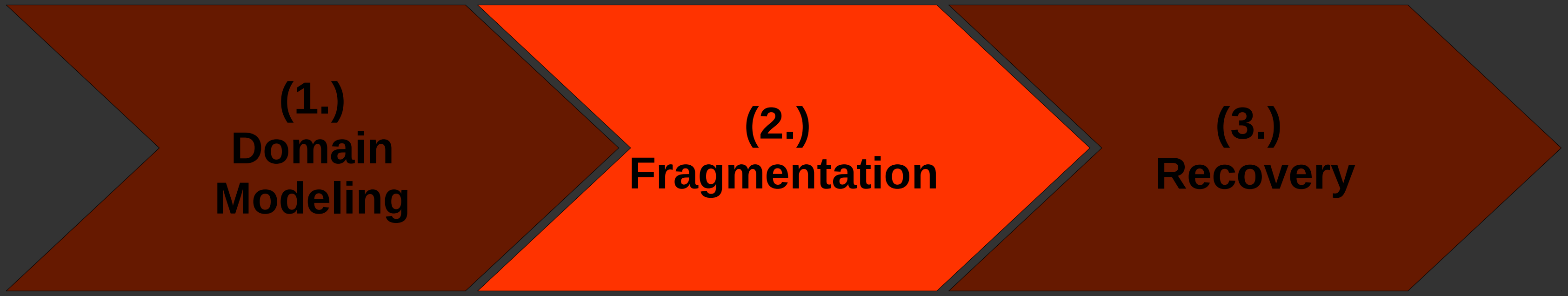


```
Foo < Entity
Bar < Entity

partOf < Foo * Bar
```

...

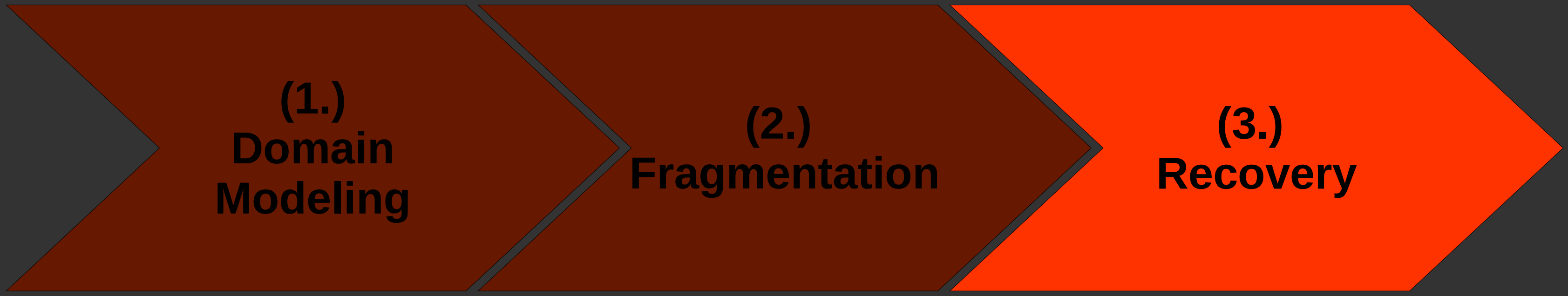
Motivation



Extends a KB with entities found in Manifestations.

Fragmentation is done by several plugins, specifically tailored to a domain.

Motivation



(1.) Domain Modeling

(2.) Fragmentation

(3.) Recovery

Can we provide singular plugins for recovery?

Unambiguous Definitions?

A partOf B :↔ ???

A conformsTo B :↔ ???

A correspondsTo B :↔ ???

A representationOf B :↔ ???

First Order Logic

Excursus

Logic Symbols

Quantifier symbol:	$\forall,\exists$
Logic Operator symbols:	$\neg,\wedge,\vee,\leftrightarrow,\rightarrow$
Description Operator symbol:	ι
Variable symbols:	$a,b,c,\ldots,x,y,z$ (lower case)
Equality/Identity symbol:	$=$
Parentheses symbols:	( ) [ ]

Non-logic Symbols

Predicate symbols:	$A,B,C,\ldots,X,Y,Z$ (upper case)

Terms

Any variable is a term.

Formulars

asdf

Mereology

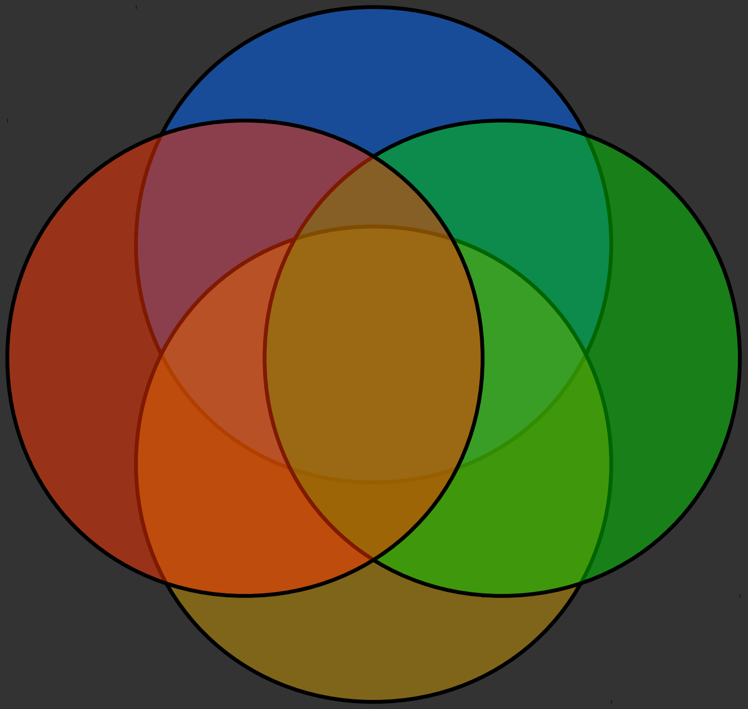
Excursus

The study of
wholes and their constituent ***parts***

Mereology

Excursus

Logic



Computer Science

Philosophy

Mathematics

Mereology

Excursus

An axiomatic system with focus on the ***parthood*** (partOf) predicate:

$P\ x\ y$

“x is part of y”

Ground Mereology

Excursus

(Reflexive)

$$P\,x\,x$$

(Antisymmetric)

$$P\,x\,y \wedge P\,y\,x \rightarrow x = y$$

(Transitive)

$$P\,x\,y \wedge P\,y\,z \rightarrow P\,x\,z$$

Mereotopology

Excursus

An axiomatic extension of a Mereology with predicates Φ other than parthood

$$\Phi\ x\ y$$

And statements like

$$P\ x\ y \rightarrow \Phi\ x\ y$$
$$\Phi\ x\ y \rightarrow P\ x\ y$$
$$P\ x\ y \leftrightarrow \Phi\ x\ y$$
$$\dots$$

Formalization #1

Formalization #1 : Example

`Foo < Entity`	$\text{Foo} \subset \text{Entity}$
`Bar < Entity`	$\text{Bar} \subset \text{Entity}$
`partOf < Foo * Bar`	$\forall\ (x,y) \in \text{Foo} \times \text{Bar}\ (\ P\ x\ y\ )$
`foo : Foo`	$\text{foo} \in \text{Foo}$
`bar : Bar`	$\text{bar} \in \text{Bar}$
`foo partOf bar`	$P\ \text{foo}\ \text{bar}$
`foo = mfoo`	$R\ \text{mfoo}\ \text{foo}$
`bar = mbar`	$R\ \text{mbar}\ \text{bar}$

Trace Recovery Rule

$$\Phi\ x\ y \wedge R\ a\ x \wedge R\ b\ y \rightarrow \Phi\ a\ b$$

Recovers the predicate Φ between two manifestations a and b.

(Assume: $\Phi \neq R$)

Formalization #1

Problem:

*Types are **not** handled properly!*

Typed Predicate Logic

Excursus

References

(1) asdf