

Sigma: An Integrated Development Environment for Logical Theories

Adam Pease and Christoph Benz Müller*

Articulate Software, Angwin, CA, USA

IKBET-2010, August 16-17, 2010 Lisbon, Portugal

*supported by DFG grant BE 2501/6-1

What is Sigma?

Sigma supports the development of formal ontologies (logical theories) in various ways

- ▶ browsing and display
- ▶ analysis and debugging
- ▶ inference
- ▶ mapping, merging and translation

What is Sigma?

Sigma supports the development of formal ontologies (logical theories) in various ways

- ▶ browsing and display
- ▶ analysis and debugging
- ▶ inference
- ▶ mapping, merging and translation

Sigma has been created to support the development of the Suggested Upper Ontology SUMO

However, it is not restricted to SUMO!

Sigma supports the development of (1) **formal ontologies** (logical theories) in various ways

- ▶ (2) **browsing and display**
- ▶ (3) **analysis and debugging**
- ▶ (4) **inference**
- ▶ (5) **mapping, merging and translation**

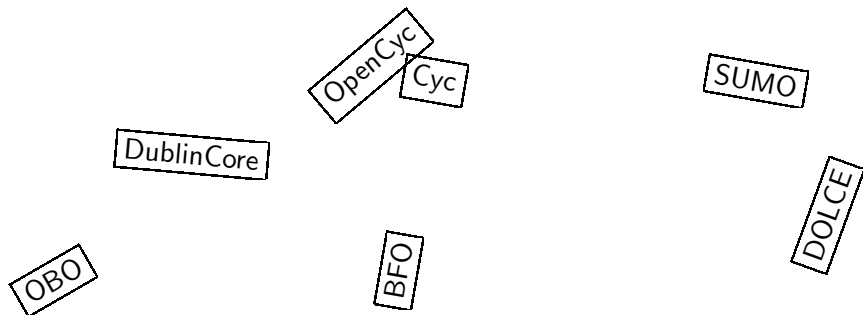
Sigma has been created to support the development of the Suggested Upper Ontology (1) **SUMO**

However, it is not restricted to SUMO!

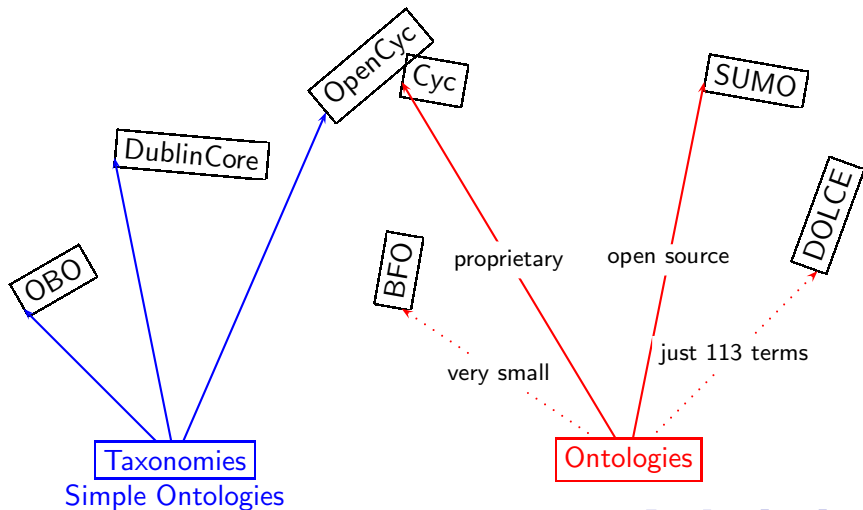


(Formal) Ontologies and SUMO

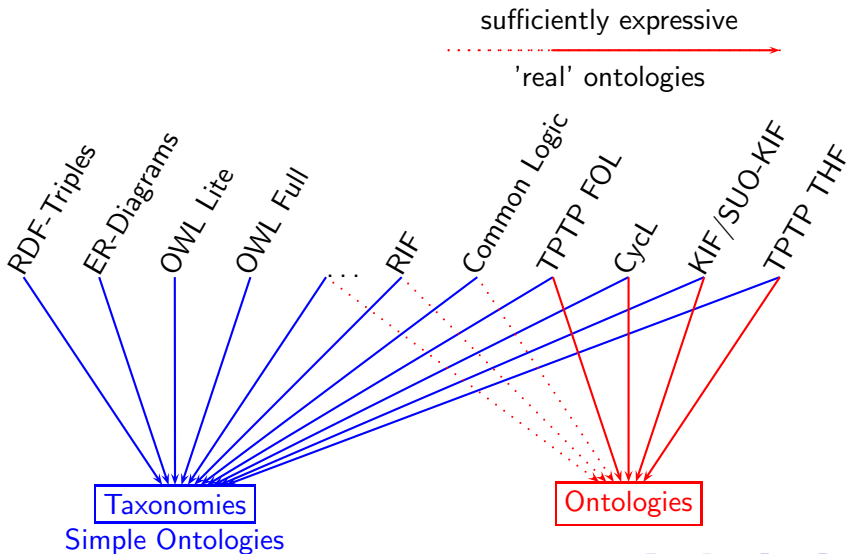
(Formal) Ontology: An Overstretched Notion?



Ontology: An (Over-)Stretched Notion?



Ontology: An (Over-)Stretched Notion?



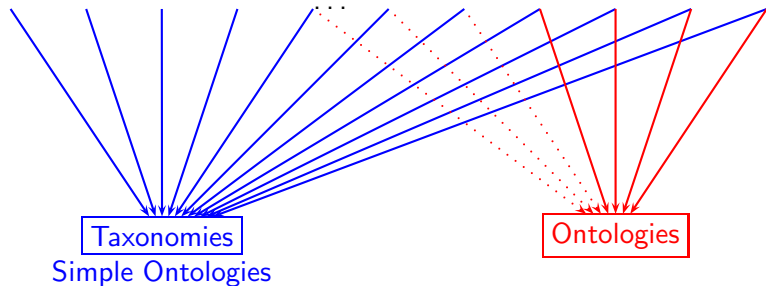
Sigma is not a graphically-based authoring tool!

structured editing

(cf. GUI builder)

flexible editing

(cf. large scale SW devel.)



Taxonomy versus Ontology: Examples

Simple Taxonomy (excerpt from SUMO's base taxonomy)

```
(subclass Physical Entity)
(subclass Abstract Entity)
(partition Entity Physical Abstract)
```

```
(subclass Object Physical)
(subclass Process Physical)
(partition Physical Object Process)
```

...

The SUMO Ontology is more than just a Taxonomy

```
(subclass Physical Entity)
(subclass Abstract Entity)
(partition Entity Physical Abstract)
```

```
(subclass Object Physical)
(subclass Process Physical)
(partition Physical Object Process)
```

```
(<=>
  (instance ?PHYS Physical)
  (exists (?LOC ?TIME)
    (and
      (located ?PHYS ?LOC)
      (time ?PHYS ?TIME))))
```

Taxonomy versus Ontology: Examples

The SUMO Ontology is more than just a Taxonomy

```
(subclass Physical Entity)
(subclass Abstract Entity)
(partition Entity Physical Abstract)
```

```
(subclass Object Physical)
(subclass Process Physical)
(partition Physical Object Process)
```

```
(=<=>
  (instance ?PHYS Physical)
  (exists (?LOC ?TIME)
    (and
      (located ?PHYS ?LOC)
      (time ?PHYS ?TIME))))
  (subclass ObjectAttitude IntentionalRelation)
  (=>
    (and
      (instance ?REL ObjectAttitude)
      (?REL ?AGENT ?THING))
    (instance ?THING Physical)))
```

- ▶ developed since: > 10 years
- ▶ original motivation: education support
(e.g. government training applications)
- ▶ open source: www.ontologyportal.org
- ▶ SUMO versus SUMO:
 - SUMO: Suggested Upper-level Ontology
 - MILO: Mid-level Ontology
 - Specific Domain-level Ontologies } SUMO
- ▶ representation language: SUO-KIF
(adaptation of the Knowledge Interchange Format KIF)
- ▶ logic: mainly first-order; some higher-order extensions

- ▶ developed since: > 10 years
- ▶ original motivation: education support
(e.g. government training applications)
- ▶ open source: www.ontologyportal.org
- ▶ SUMO versus SUMO:
 - SUMO: Suggested Upper-level Ontology
 - MILO: Mid-level Ontology
 - Specific Domain-level Ontologies } SUMO
- ▶ representation language: SUO-KIF
(adaptation of the Knowledge Interchange Format KIF)
- ▶ logic: mainly first-order; some higher-order extensions

- ▶ developed since: > 10 years
- ▶ original motivation: education support
(e.g. government training applications)
- ▶ open source: www.ontologyportal.org
- ▶ SUMO versus SUMO:
 - SUMO: Suggested Upper-level Ontology
 - MILO: Mid-level Ontology
 - Specific Domain-level Ontologies
- ▶ representation language: SUO-KIF
(adaptation of the Knowledge Interchange Format KIF)
- ▶ logic: mainly first-order; some higher-order extensions

- ▶ developed since: > 10 years
- ▶ original motivation: education support
(e.g. government training applications)
- ▶ open source: www.ontologyportal.org
- ▶ SUMO versus SUMO:

SUMO: Suggested Upper-level Ontology
MILO: Mid-level Ontology
Specific Domain-level Ontologies } SUMO

- ▶ representation language: SUO-KIF
(adaptation of the Knowledge Interchange Format KIF)
- ▶ logic: mainly first-order; some higher-order extensions

- ▶ developed since: > 10 years
- ▶ original motivation: education support
(e.g. government training applications)
- ▶ open source: www.ontologyportal.org
- ▶ SUMO versus SUMO:
 - SUMO: Suggested Upper-level Ontology
 - MILO: Mid-level Ontology
 - Specific Domain-level Ontologies } SUMO
- ▶ representation language: SUO-KIF
(adaptation of the Knowledge Interchange Format KIF)
- ▶ logic: mainly first-order; some higher-order extensions

SUMO			
Total Terms	relations	Total Axioms	Rules
1173	353	4741	932
MILO			
Total Terms	relations	Total Axioms	Rules
1662	159	5116	1183
Domain ontologies			
Total Terms	relations	Total Axioms	Rules
17312	708	77974	2041
Total for all ontologies			
Total Terms	relations	Total Axioms	Rules
20147	1220	87831	4156

Table 1: SUMO term and axiom statistics



SIGMA: Browsing and Display

SIGMA: Browsing and Display — Hyperlinked Text

The screenshot shows the Sigma knowledge engineering environment interface. At the top, there is a menu bar (File, Edit, View, History, Bookmarks, Tools, Help) and a browser address bar with the URL `http://sigma.ontologyportal.org:4010/sigma/Browse.jsp?kb=SUMO&lang=EnglishLanguage&term=Walking&sin`. Below the browser, the interface displays the search results for the term "Walking".

Walking (walking)

appearance as argument number 1

(documentation [Walking](#) [EnglishLanguage](#) "ambulating relatively slowly, i.e. moving in such a way that at least one foot is always in contact with the ground.")

(externalImage [Walking](#) "http://upload.wikimedia.org/wikipedia/commons/0/0f/Robotpeintre.gif")

(externalImage [Walking](#) "http://upload.wikimedia.org/wikipedia/commons/6/6f/Walk-Cycle.gif")

(externalImage [Walking](#) "http://upload.wikimedia.org/wikipedia/commons/d/d2/Marcheur_en_comp%C3%A9tion.jpg")

(subclass [Walking](#) [Ambulating](#))

appearance as argument number 2

(partition [Ambulating](#) [Walking](#) [Running](#))

(subclass [Wading](#) [Walking](#))

(termFormat [EnglishLanguage](#) [Walking](#) "walking")

antecedent

```
(=>
  (and
    (instance ?WALK Walking)
    (instance ?RUN Running)
    (agent ?WALK ?AGENT)
    (agent ?RUN ?AGENT)
    (holdsDuring
      (WhenEh ?WALK
        (measure ?AGENT
          (SpendEn ?LENGTH1 ?TIME)))
      (holdsDuring
        (WhenEh ?RUN
          (measure ?AGENT
            (SpendEn ?LENGTH2 ?TIME))))
      (greaterThan ?LENGTH2 ?LENGTH1)))
```

consequence

```
Find: ontology Previous Next Highlight all Match case Reached end of page, continued from top
```

Done

KB: [SUMO](#) [Language:](#) [EnglishLanguage](#)

Walking (walking)

Rollerblade about amble amblulate ambulation angry walk backpack break bumble canter cavoco circumambulate clamber climb climb up clump clump cock coggle constitutional constitutionalism countermarch crab creep curvet dash debouch dodder digitate drag dressage stride stir escalate exhibit falter foot break file file in file out fire walking founce founder foot footer footstep footstep forage gait gallop...

externalImage [walking](#) and "http://upload.wikimedia.org/wikipedia/commons/0/0f/Robotpeintre.gif"

externalImage [walking](#) and "http://upload.wikimedia.org/wikipedia/commons/6/6f/Walk-Cycle.gif"

externalImage [walking](#) and "http://upload.wikimedia.org/wikipedia/commons/d/d2/ Marcheur_en_comp%C3%A9tion.jpg"

Walking is a subclass of [ambulating](#)

Ambulating is exhaustively partitioned into [walking](#) and [running](#)

Wading is a subclass of [walking](#)

term format english language: walking and "walking"

- If a process is an instance of [walking](#) and process is an instance of [running](#) and an agent is an agent of process and agent is an agent of process and the measure of agent is a length measure per a time duration holds during the time of existence of process and the measure of agent is length measure per time duration holds during the time of existence of process
- then length measure is greater than length measure

SIGMA: Browsing and Display — Hyperlinked Text

The screenshot displays the SIGMA knowledge engineering environment. At the top, there is a menu bar (File, Edit, View, History, Bookmarks, Tools, Help) and a browser address bar showing the URL `http://localhost:8080/sigma/TreeView.jsp?kb=SUMO&simple=yes&term=Object`. Below the browser, the application header includes the SIGMA logo, the text "Sigma knowledge engineering environment", and a "Simplified Browsing Interface" label. There are also links for "[Home]", "KB: SUMO", and "Language: EnglishLanguage".

The main interface features a search bar labeled "KB Term:" with the input "Object" and a "Show" button. On the left, a hierarchical tree view shows the ontology structure, with "Object" selected. The main content area displays the definition of "object" and a table of its relationships.

object

Corresponds roughly to the class of ordinary objects. Examples include normal physical objects, geographical regions, and locations of *processes*, the complement of *objects* in the *physical* class. In a 4D ontology, an *object* is something whose spatiotemporal extent is thought of as dividing into spatial parts roughly parallel to the time-axis.

Relationships		
Parents	physical	An entity that has a location in space-time. Note that locations are themselves understood to have a location in space-time.
Children	agent	Something or someone that can act on its own and produce changes in the world.
	artifact	An <i>object</i> that is the product of a <i>making</i> .
	collection	Collections have <i>members</i> like <i>classes</i> , but, unlike <i>classes</i> , they have a position in space-time and <i>members</i> can be added and subtracted without thereby changing the identity of the <i>collection</i> . Some examples are tools, football teams, and flocks of sheep.
	ContactSite	A <i>ContactSite</i> is an <i>object</i> , typically a Place or a <i>residence</i> or a <i>communication device</i> such as a <i>telephone</i> , that has some kind of address identifier and can serve as a point of contact for a <i>human</i> or <i>organization</i> .
	prepared food	Food that is the result of <i>cooking</i> .
	raw food	Food that is not the result of <i>cooking</i> .



SIGMA: Analysis and Debugging

Position: Inconsistencies are just a fact of life! Finding them improves quality of SUMO. However, 'global consistency' is not pre-requisite for usefulness of SUMO.

Sigma provides both

- ▶ general purpose tools:
 - ▶ first-order theorem provers → next part
 - ▶ (new) higher-order theorem provers → ARCOE-2010 talk
- ▶ special purpose tools for error detection:
 - ▶ check for terms without root in subclass hierarchy
 - ▶ check for term with parents that are defined disjoint
 - ▶ (indicative) check for terms lacking documentation
 - ▶ (indicative) check for terms that do not occur in any rules

Diagnostics tool for SUMO–WordNet mappings

- ▶ finds WordNet synsets without SUMO term
- ▶ finds WordNet synsets not available in current KB
- ▶ taxonomy comparison component (hierarchy mismatch not always an error!)

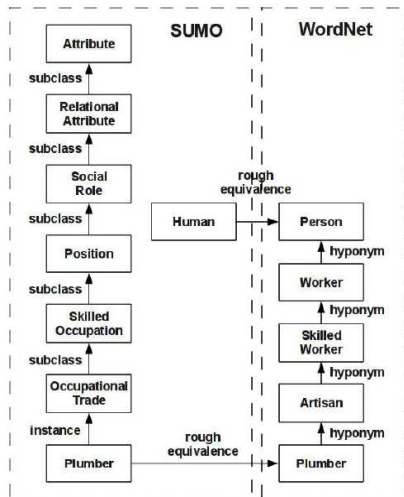


Figure 3: Comparing hierarchies of SUMO and WordNet

Reasoners integrated with Sigma

First-Order (work had strong impact on CASC: LTB division)

- ▶ KIFVampire
- ▶ SiNE relevance filter
- ▶ SystemOnTPTP FOL (> 40 reasoning systems)

Reasoners integrated with Sigma

First-Order (work had strong impact on CASC: LTB division)

- ▶ KIFVampire
- ▶ SiNE relevance filter
- ▶ SystemOnTPTP FOL (> 40 reasoning systems)

Higher-Order

- ▶ LEO-II
- ▶ SystemOnTPTP THF (soon; 6 reasoning systems)

Reasoners integrated with Sigma

First-Order (work had strong impact on CASC: LTB division)

- ▶ KIFVampire
- ▶ SiNE relevance filter
- ▶ SystemOnTPTP FOL (> 40 reasoning systems)

Higher-Order

- ▶ LEO-II
- ▶ SystemOnTPTP THF (soon; 6 reasoning systems)

Mainly used for

- ▶ question answering
- ▶ analysis and debugging

Reasoners integrated with Sigma

First-Order (work had strong impact on CASC: LTB division)

- ▶ KIFVampire
- ▶ SiNE relevance filter
- ▶ SystemOnTPTP FOL (> 40 reasoning systems)


Higher-Order (see ARCOE-2010 talk!)


- ▶ LEO-II
- ▶ SystemOnTPTP THF (soon; 6 reasoning systems)

Mainly used for

- ▶ question answering
- ▶ analysis and debugging

SIGMA: Inference with FO-ATPs in Question Answering

 Sigma knowledge engineering environment
SystemOnTPTP Interface

[Home](#) | [Graph](#) | [Prefs](#) | KB: SUMO | Language: 

Local SystemOnTPTP Remote SystemOnTPTP System: EP---0.999

Maximum answers: Query time limit:


TPTP Proof IDV-Proof tree Hyperlinked KIF

(SZS Status Theorem)
Answer 1. [definite] ?X6 = Red

1. (exists (?X6) (instance ?X6 PrimaryColor))	[Query]
2. (instance Red PrimaryColor)	[KB]
3. (instance Red PrimaryColor)	2
4. (not (instance ?X6 PrimaryColor))	1
5. (not (instance Red PrimaryColor))	4
6. True	3 5

(SZS Status Theorem)
Answer 1. [definite] ?X = Blue

1. (instance Blue PrimaryColor)	[KB]
2. (exists (?X1) (instance Blue PrimaryColor))	[Instantiated Query]
3. (not (exists (?X1) (instance Blue PrimaryColor)))	2
4. (not (instance Blue PrimaryColor))	3
5. (instance Blue PrimaryColor)	1
6. (not (instance Blue PrimaryColor))	4
7. True	6 5
8. True	7
9. True	8



SIGMA: Inference with FO-ATPs

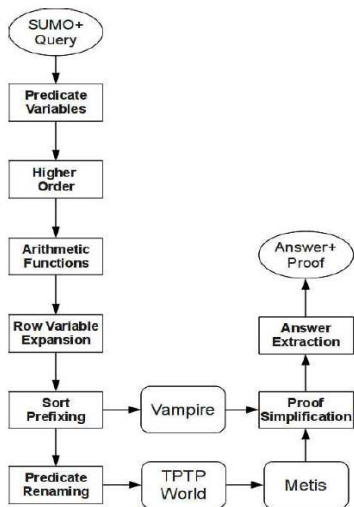


Figure 4: Sigma pre- and post-processing steps

Further Reading:

- [PeaseSutcliffe, ESARLT-2007]
- [TracSutcliffePease, PAAR-2008]
- [PeaseEtAl., AICom, 2010]

Predicate Variables

```
(instance part TransitiveRelation)
(<=>
  (instance ?REL TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (?REL ?INST1 ?INST2)
        (?REL ?INST2 ?INST3))
      (?REL ?INST1 ?INST3))))
```


Predicate Variables

```
(holds instance part TransitiveRelation)
(<=>
  (holds instance ?REL TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (holds ?REL ?INST1 ?INST2)
        (holds ?REL ?INST2 ?INST3))
      (holds ?REL ?INST1 ?INST3))))
```

Approach A: global 'dummy' predicate

Predicate Variables

```
(holds instance part TransitiveRelation)
(<=>
  (holds instance ?REL TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (holds ?REL ?INST1 ?INST2)
        (holds ?REL ?INST2 ?INST3))
      (holds ?REL ?INST1 ?INST3))))
```

Approach A: global 'dummy' predicate
→ worsens prover performance (indexing of **holds**)

Predicate Variables

```
(instance part TransitiveRelation)
(<=>
  (instance ?REL TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (?REL ?INST1 ?INST2)
        (?REL ?INST2 ?INST3))
      (?REL ?INST1 ?INST3))))
```

Predicate Variables

```
(instance part TransitiveRelation)
(<=>
  (instance part TransitiveRelation)
  (forall (?INST1 ?INST2 ?INST3)
    (=>
      (and
        (part ?INST1 ?INST2)
        (part ?INST2 ?INST3))
      (part ?INST1 ?INST3))))
```

- Approach B:** predicate instantiation
→ good performance
→ but semantics of **REL** modified

Higher-Order: Embedded Formulas

(believes John (likes Mary Jeff))

Query:

(believes John (likes Mary ?X))

Higher-Order: Embedded Formulas

(believes John '(likes Mary Jeff))

Query:

(believes John '(likes Mary ?X))

Approach:

Quoting of embedded formulas

Higher-Order: Embedded Formulas

```
(believes John  
  '(and  
    (likes Mary Jeff)  
    (likes Bill Sue)))
```

Query:

```
(believes John '(likes Mary ?X))
```

Approach:

Quoting of embedded formulas

→ connectives in embedded formulas lose their semantics

Higher-Order: Embedded Formulas

```
(believes John  
  '(and  
    (likes Mary Jeff)  
    (likes Bill Sue)))
```

Query:

```
(believes John '(likes Mary ?X))
```

- Approach:** Quoting of embedded formulas
- connectives in embedded formulas lose their semantics
 - solution with higher-order provers proposed

[BenzmüllerPease, PAAR-2010],[BenzmüllerPease, ARCOE-2010]

Arithmetic Expressions

```
(=<=>
  (lessThanOrEqualTo ?NUMBER1 ?NUMBER2)
  (or
    (equal ?NUMBER1 ?NUMBER2)
    (lessThan ?NUMBER1 ?NUMBER2)))
```

Arithmetic Expressions

```
(<=>
  $lesseq(?NUMBER1,?NUMBER2)
  (or
    (?NUMBER1 = ?NUMBER2)
    $less(?NUMBER1,?NUMBER2)))
```

Approach: Translate into native symbols as required by provers

Arithmetic Expressions

```
(=<=>
  $lesseq(?NUMBER1,?NUMBER2)
  (or
    (?NUMBER1 = ?NUMBER2)
    $less(?NUMBER1,?NUMBER2)))
```

Approach: Translate into native symbols as required by provers
→ more and more FO-ATPs provide support for basic arithmetic

Row variables

```
(=>  
  (and  
    (subrelation ?REL1 ?REL2)  
    (?REL1 @ROW))  
    (?REL2 @ROW))
```

Row variables

```
(=>  
  (and  
    (subrelation ?REL1 ?REL2)  
    (?REL1 ?ROW1))  
  (?REL2 ?ROW1))
```

```
(=>  
  (and  
    (subrelation ?REL1 ?REL2)  
    (?REL1 ?ROW1 ?ROW2))  
  (?REL2 ?ROW1 ?ROW2))
```

Approach: expand up to maximum arity of 7

Row variables

```
(=>  
  (and  
    (subrelation ?REL1 ?REL2)  
    (?REL1 ?ROW1))  
  (?REL2 ?ROW1))
```

```
(=>  
  (and  
    (subrelation ?REL1 ?REL2)  
    (?REL1 ?ROW1 ?ROW2))  
  (?REL2 ?ROW1 ?ROW2))
```

- Approach:** expand up to maximum arity of 7
- few axioms with 2 row variables (max. 49 expanded axioms)
 - limits semantics of row variables

Sort Prefixing

```
(=>
  (and
    (instance ?TRANSFER Transfer)
    (agent ?TRANSFER ?AGENT)
    (patient ?TRANSFER ?PATIENT))
  (not
    (equal ?AGENT ?PATIENT)))
```

Sort Prefixing

```
(=>
  (and
    (instance ?TRANSFER Transfer)
    (agent ?TRANSFER ?AGENT)
    (patient ?TRANSFER ?PATIENT))
  (not
    (equal ?AGENT ?PATIENT)))

(domain agent 2 Agent)
(domain patient 2 Object)
```


Sort Prefixing

```
(=>
  (and
    (instance ?AGENT Agent)
    (instance ?PATIENT Object))
  (=>
    (and
      (instance ?TRANSFER Transfer)
      (agent ?TRANSFER ?AGENT)
      (patient ?TRANSFER ?PATIENT))
    (not
      (equal ?AGENT ?PATIENT))))
```

Approach: extract SUMO type information and add

Predicate Renaming

(`instance instance` BinaryPredicate)

Predicate Renaming

`(p_instance t_instance BinaryPredicate)`

Approach: decouple predicates by renaming

Predicate Renaming

Approach: decouple predicates by renaming
→ information loss for some predicates; here for `p_instance`



SIGMA: Mapping, Merging and Translation

Translation

- ▶ Sigma provides (some restricted) support for reading and writing OWL format
 - ▶ has been be applied to SUMO:
 - lossy translation
 - bulk of axioms become informative comments for users
- ▶ Sigma supports translations to
 - ▶ TPTP FOL (classical first-order logic)
 - ▶ TPTP THF (classical higher-order logic)
- ▶ export to PROLOG
- ▶ export (prototype) to SQL
- ▶ SUO-KIF is expressive: similar translation to and from other less expressive formats should be possible

Mapping and Merging

- ▶ simple algorithm developed
 - ▶ used for creating an initial alignment with the lightweight Open Biomedical Ontologies OBO
 - ▶ however, bulk of effort still spent by human in selecting accurate matches (due to massive number of false positives when trying it fully automatically)

Conclusion

- ▶ Sigma is a toolkit and testbed for (rich) ontology development and application
- ▶ Sigma has co-evolved with SUMO
- ▶ Sigma is open source: www.ontologyportal.org

Ongoing and future work

- ▶ package a formal release (no formal release for > 2 years)
- ▶ semantics for higher-order aspects of SUMO
- ▶ integration of higher-order reasoning systems
- ▶ support layer in Sigma for integrated, heterogeneous reasoning support
- ▶ inbuilt first-order theorem prover