

Combining Ontologies: an Algebraic Approach

Gilles Falquet, Jacques Guyot, Claire-Lise
Mottaz Jiang

Université de Genève,
Centre universitaire d'informatique

Outline

Motivations

A Model for ontology repositories

Operations for combining ontologies

Conclusions

Motivations for this work

Ontologies are powerful tools for

- Storing a shared conceptualization of a domain
- Integrating or interconnecting different applications, data sets, models, ...
- Semantically indexing documents
- Enhancing standard applications with reasoning capabilities in a domain
- etc.

The question is: how do I find an adequate ontology for my domain/problem/application?

Generally, the “right” ontology does not exist.

“Onto₁ is too small”

“Onto₂ lacks some relations”

“Onto₃ is too large”

“Onto₄ is ... not really an ontology”

Diversity

Existing ontologies are expressed in different languages

- Graph based
- Description logics bases
- Frame based
- ...

With different levels of formalisation

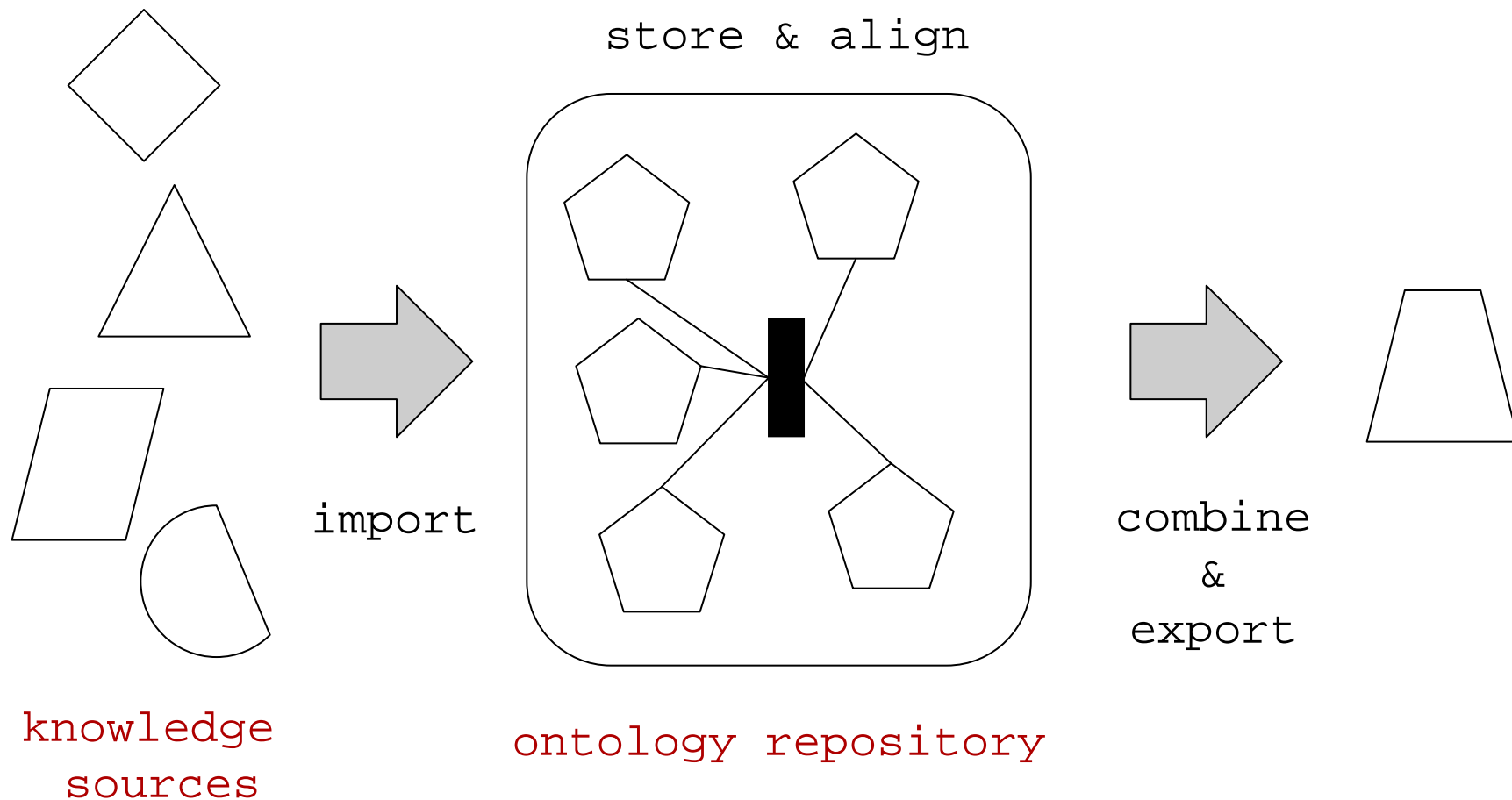
- Formal languages (OWL, FOL, ...)
- Semi-formal (Wordnet, semantic networks, ...)
- Non formal (definitions in natural language)

Diversity II

Some interesting knowledge resources are “pre-ontologies” or other “non-ontologies”

- Lexicons
- Thesaurus
- Glossaries
- Bilingual lexicons
- Encyclopedias
- Aligned texts

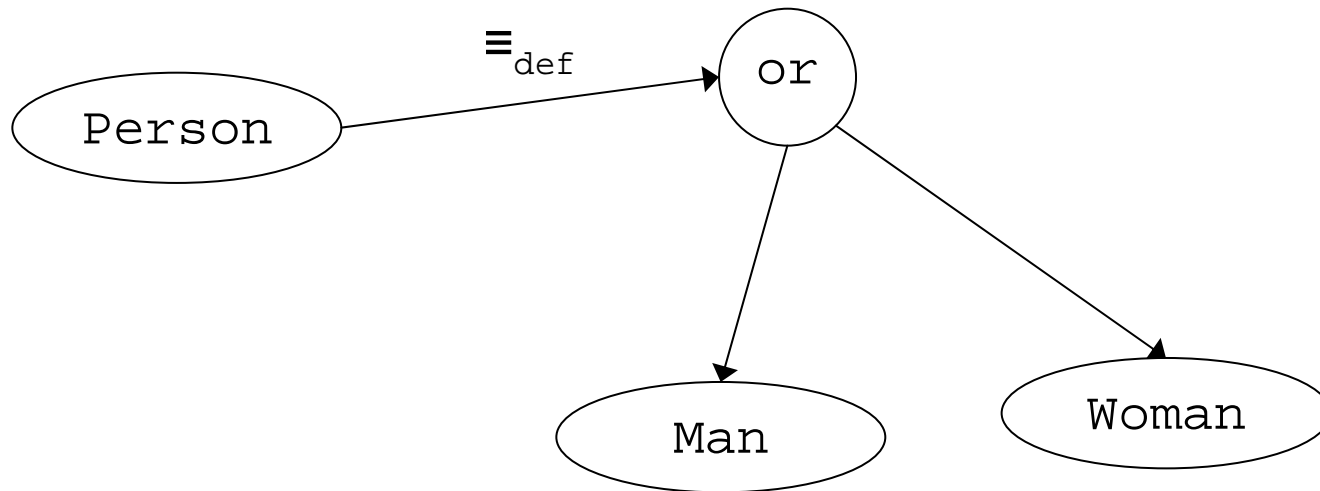
An approach to re-using ontological knowledge



Model for ontology repository

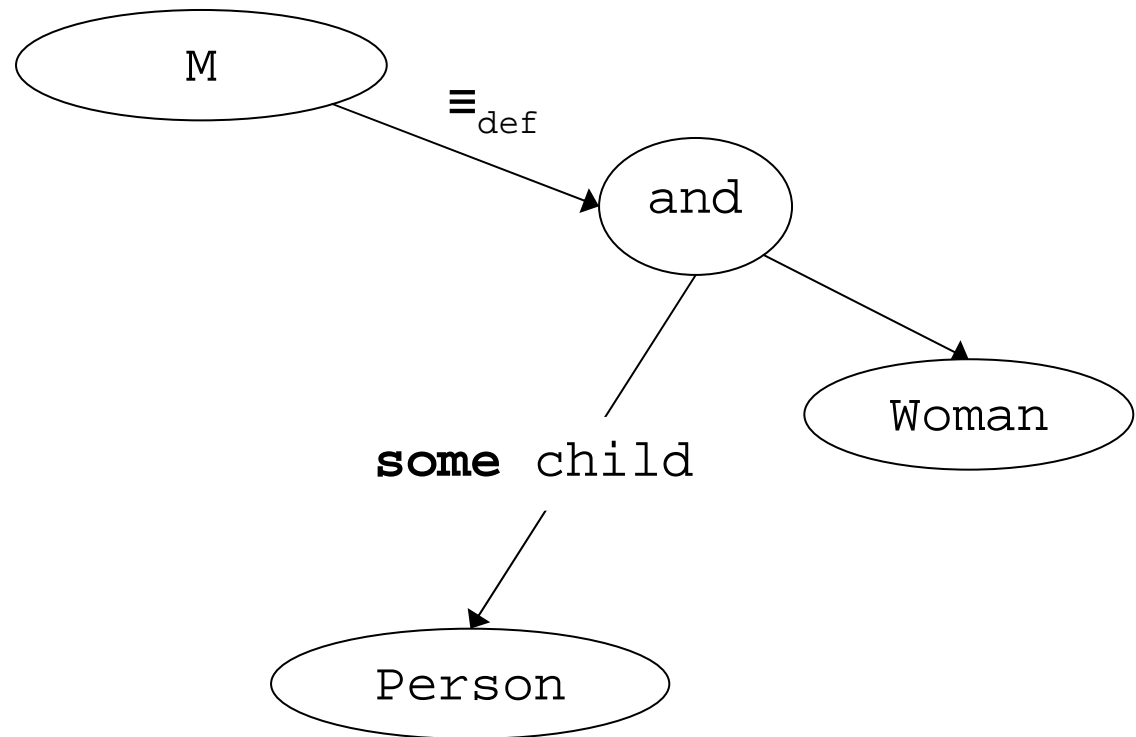
Graphs that represent Description logic (OWL) expressions

- Concept expression => a subgraph
- Concept constructor => labeled node

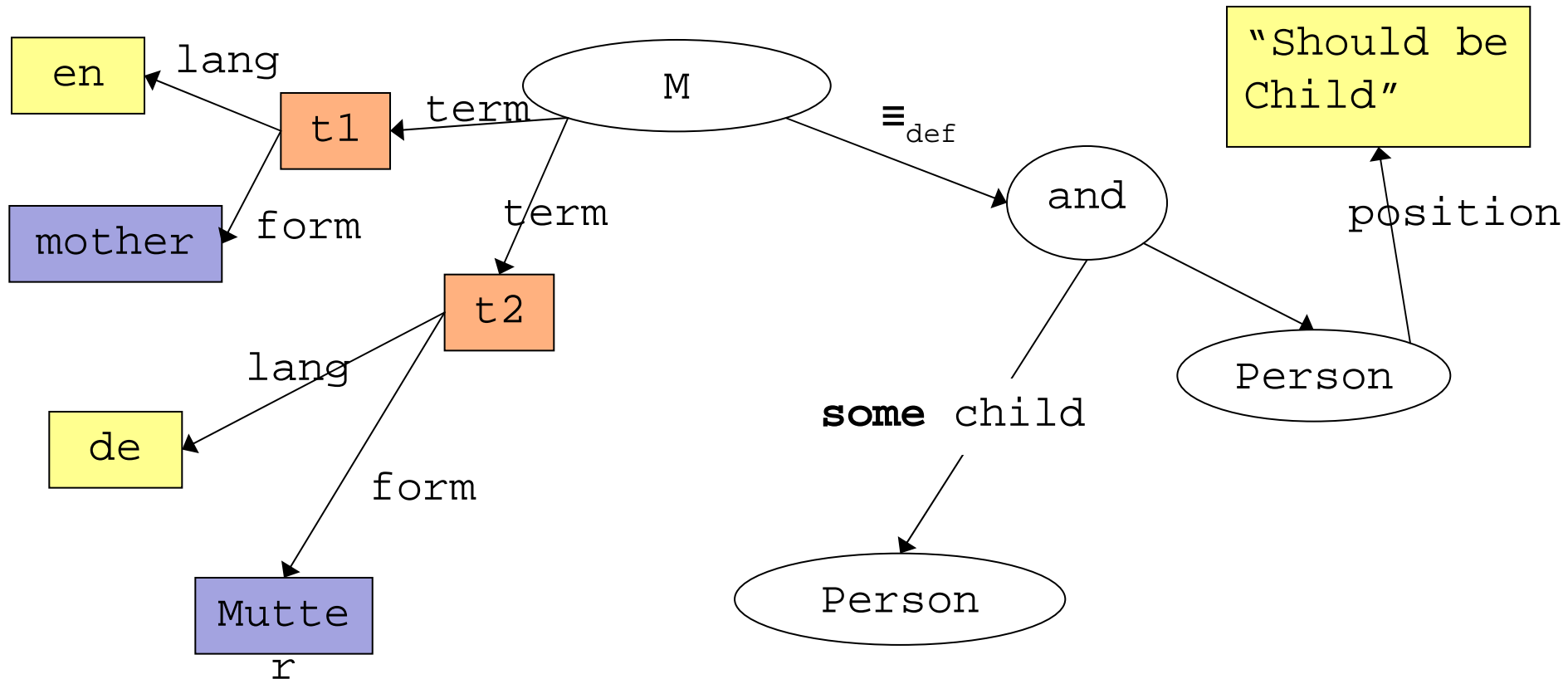


Model for ontology repository

$M \equiv \text{Woman and (some child . Person)}$



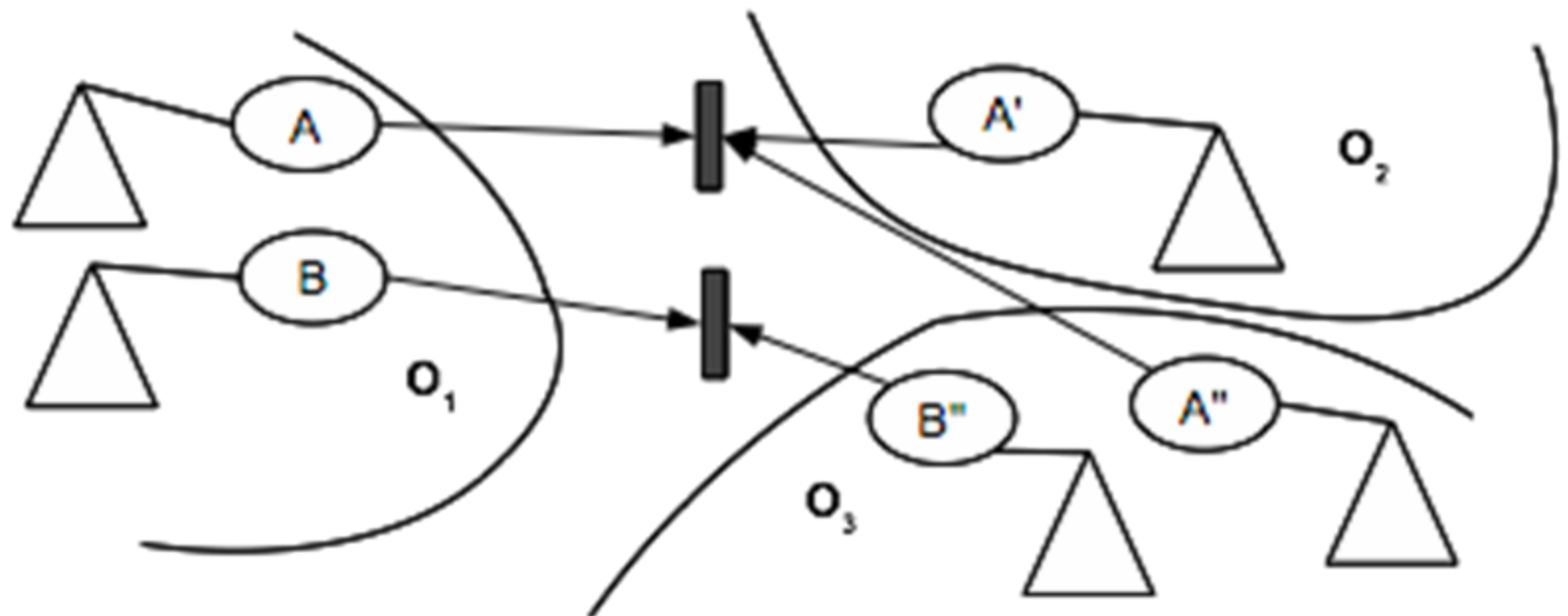
With Annotations



Alignment structure

Atomic concepts linked to alignment objects

Aligned concepts represent the same concept in differing contexts



Ontology Algebra

Operations that take one or two ontologies as input and produce a new ontology as output.

Projection(properties) : ontology --> ontology

Selection(condition) : ontology --> ontology

Union : ontology, ontology --> ontology

Union_import : ontology, ontology --> ontology

Join : ontology, ontology --> ontology

Projection

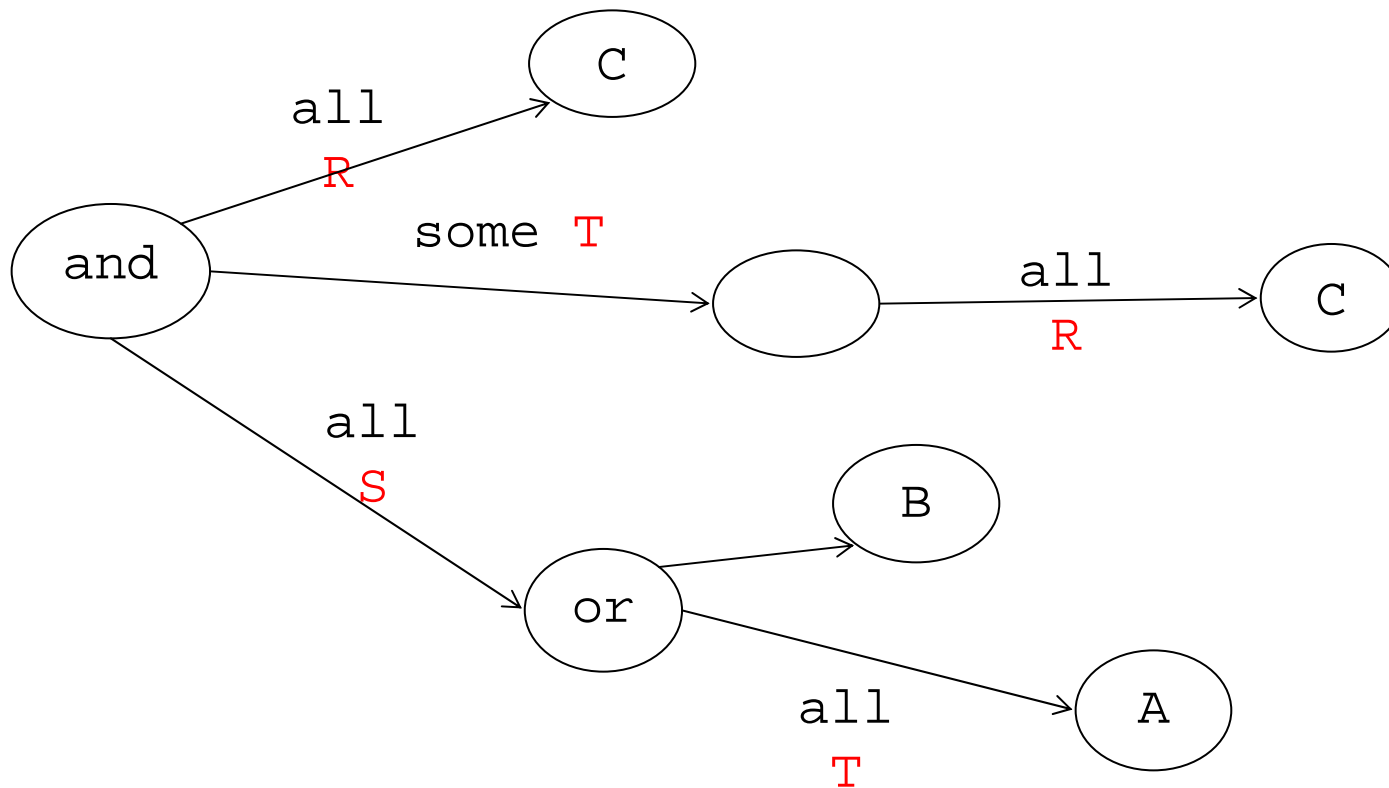
Remove properties that don't belong to a given list

Typical use :

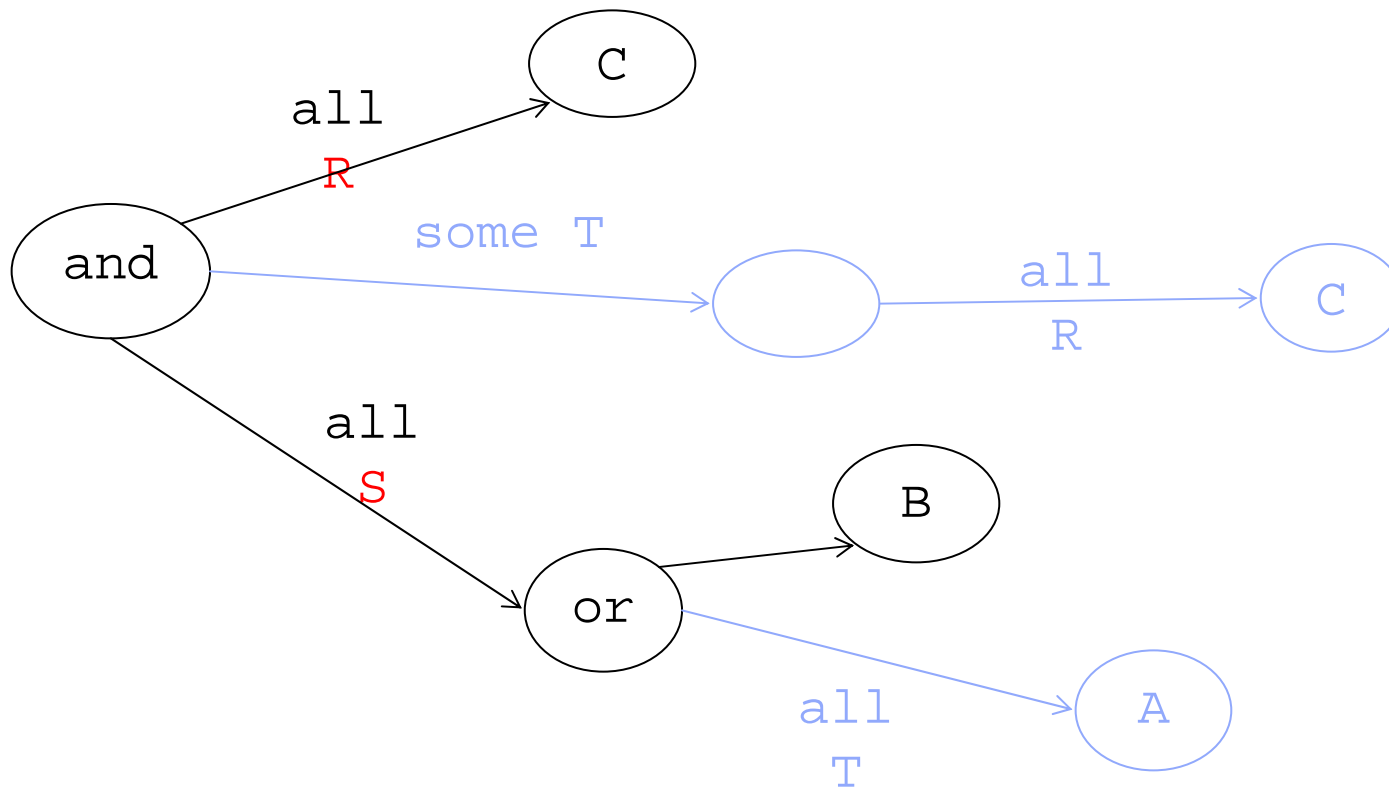
- simplify an ontology
- prepare to import properties into another ontology

Example

Projection on $\{R, S\}$ of



Projection



Selection

Selects a subset of the defined concepts
Based on a selection expression

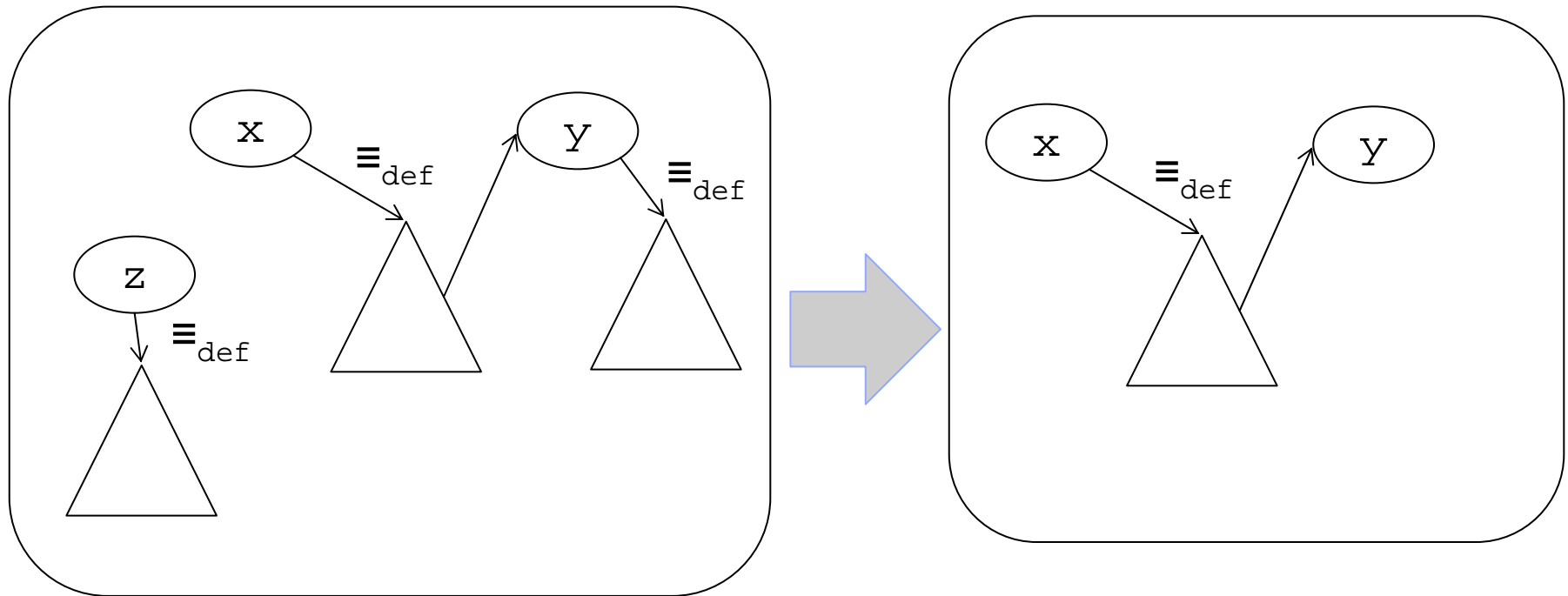
For example

- concepts that have **some child** in their definition
- concepts that refer to **Person**
- concepts aligned with a concept in ontology

O_2

Selection

Remove the definition of concepts not retained
Keep names of referred concepts

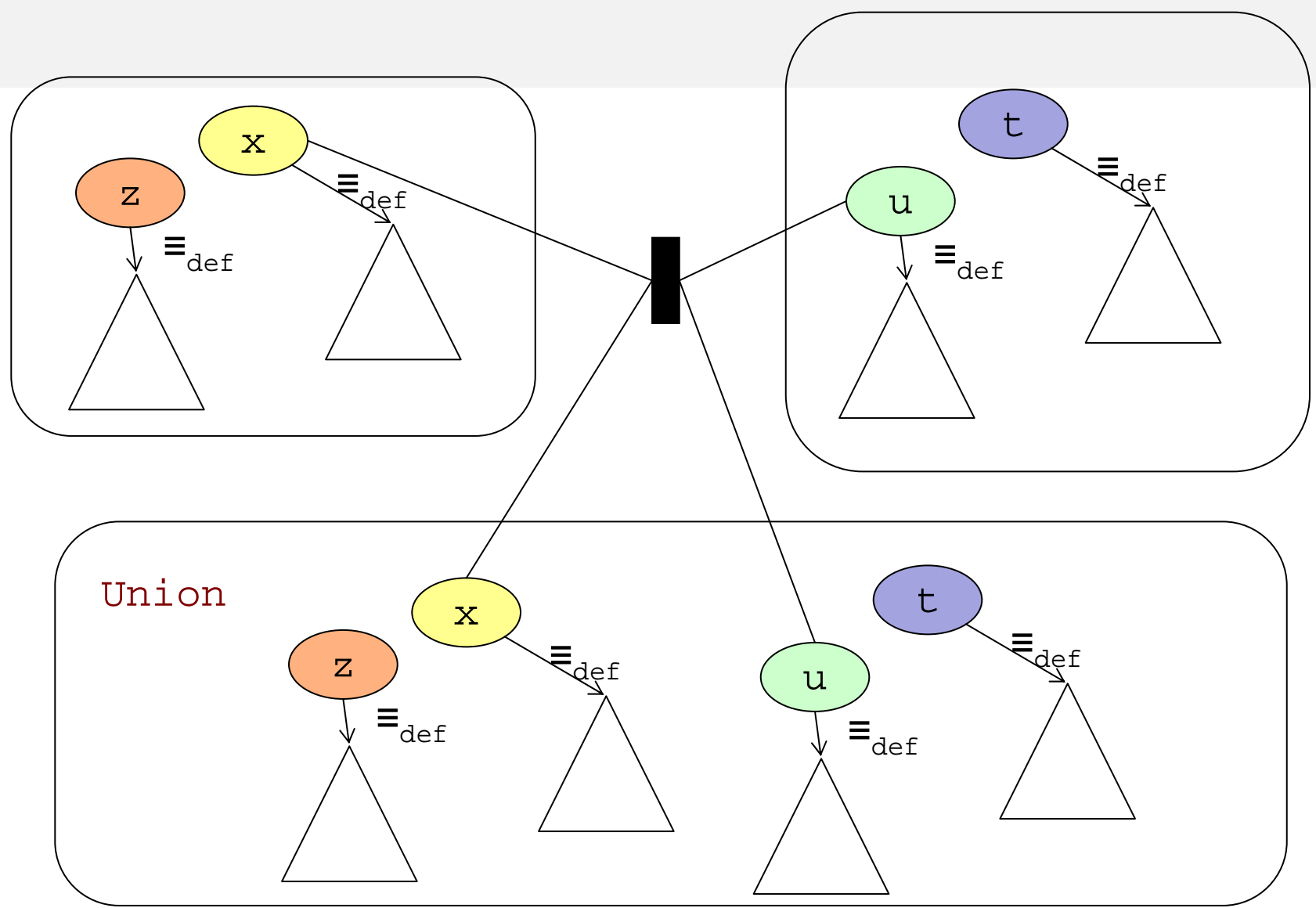


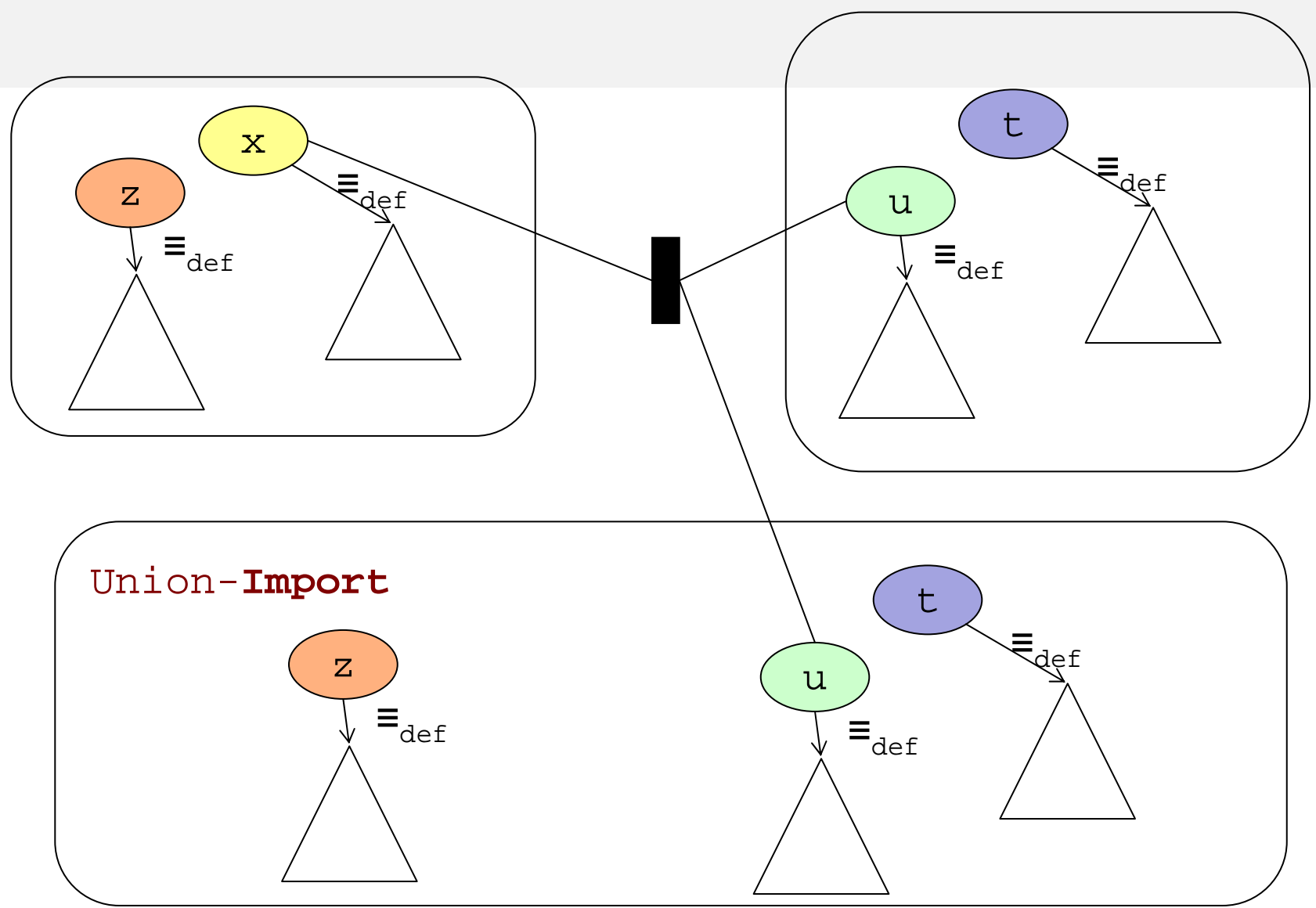
Union

The disjoint union of the ontologies' graphs

Keeps all the concepts and their definitions

Aligned concepts yield two different concepts
linked to the same alignment objet (equivalent
concepts)





Join

Combine de definitions of aligned concepts

if

$A \equiv E1$ in $O1$

$A' \equiv E2$ in $O2$

A and A' are aligned

create a concept

$A'' \equiv E1$ **and** $E2$

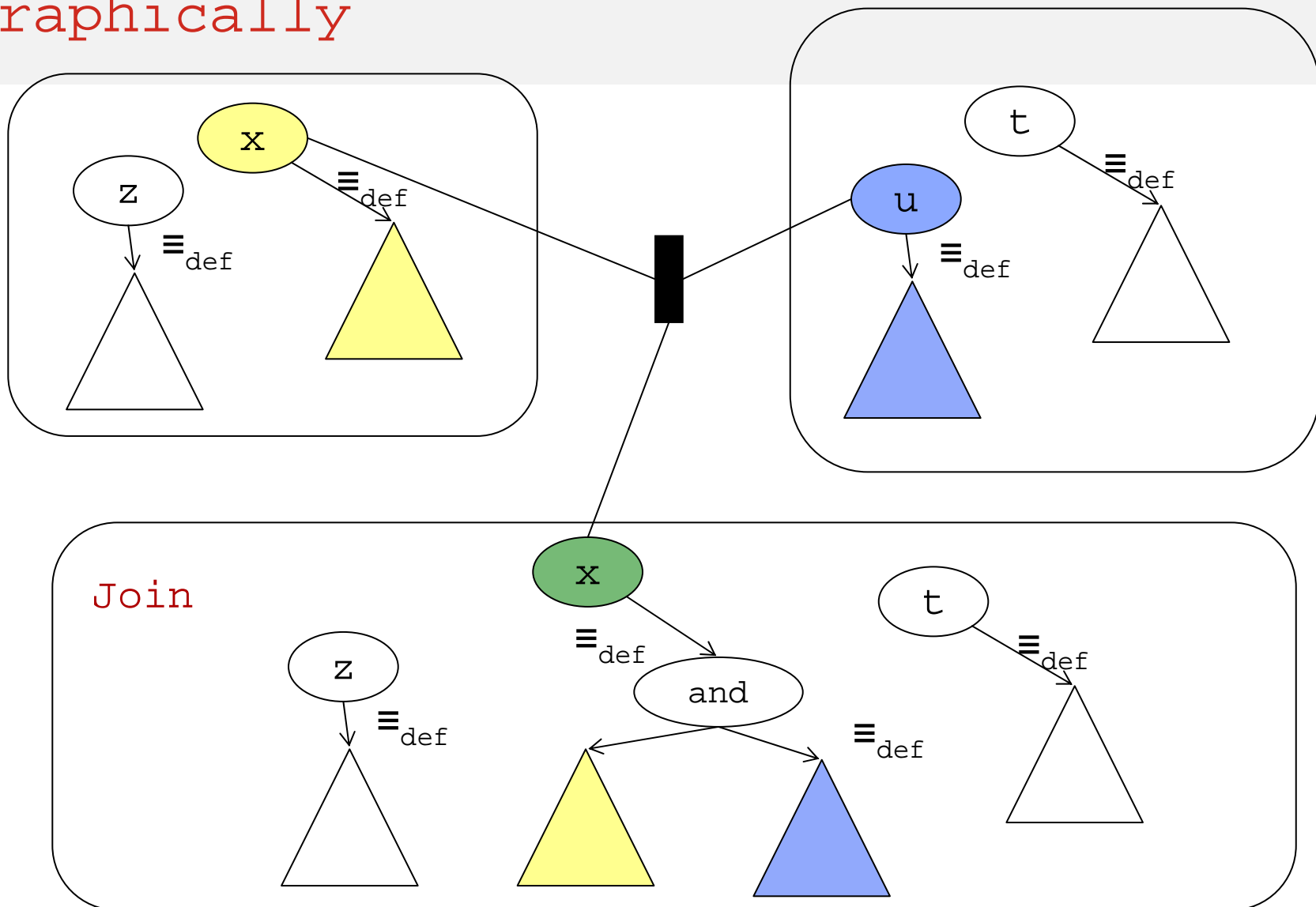
O1 : House \equiv Construction **and** (some part . Wall)
and (some part . Roof)

O2 : FamilyHouse \equiv Building **and** (some user .
Family)

If O1 : House and O2 : FamilyHouse are aligned,
the join will contain

O3 : H \equiv_{def} Construction **and** (some part . Wall)
and (some part . Roof) **and** Building
and (some user . Family)

Graphically



Typical use of the operations

Importing what is about a property P

```
join( $O_1$ , proj( $O_2$ ,  $P$ ))
```

Replace the is-a hierachy of O_1 with the one of O_2

```
join(proj( $O_1$ , not{subclassOf} ), proj( $O_2$ ,  
{subclassOf}))
```

Conclusion and future work

A way to express ontology combination for
ontology re-use

Implementation

- ontology repository = database system
- operation implementation with db operations and procedures
- importing/exporting OWL, Towntology, etc. is straightforward

Future

- work on the linguistic aspects: importing terminologies, glossaries, ...
- implement existing alignment algorithms
- represent time series of ontologies