Assessing bidirectional model transformation tools

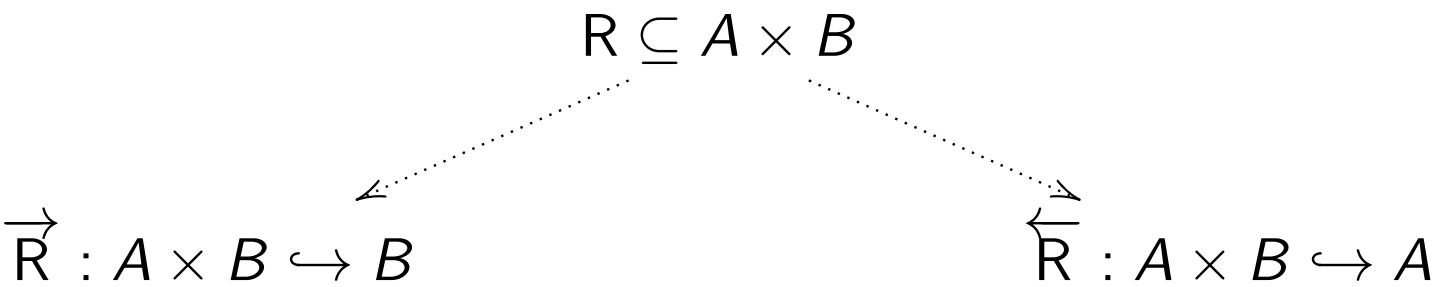
Alcino Cunha

BIRS BX workshop, Banff
December, 2013

Motivation

- Survey (operational) BX tools
- Focus on expressiveness
- Example driven
- Systematic exploration of the consistency design space
- Clarification of supported BX laws
- Repository of unit tests for (future) tool developers

Constraint maintainers



Basic properties of relations

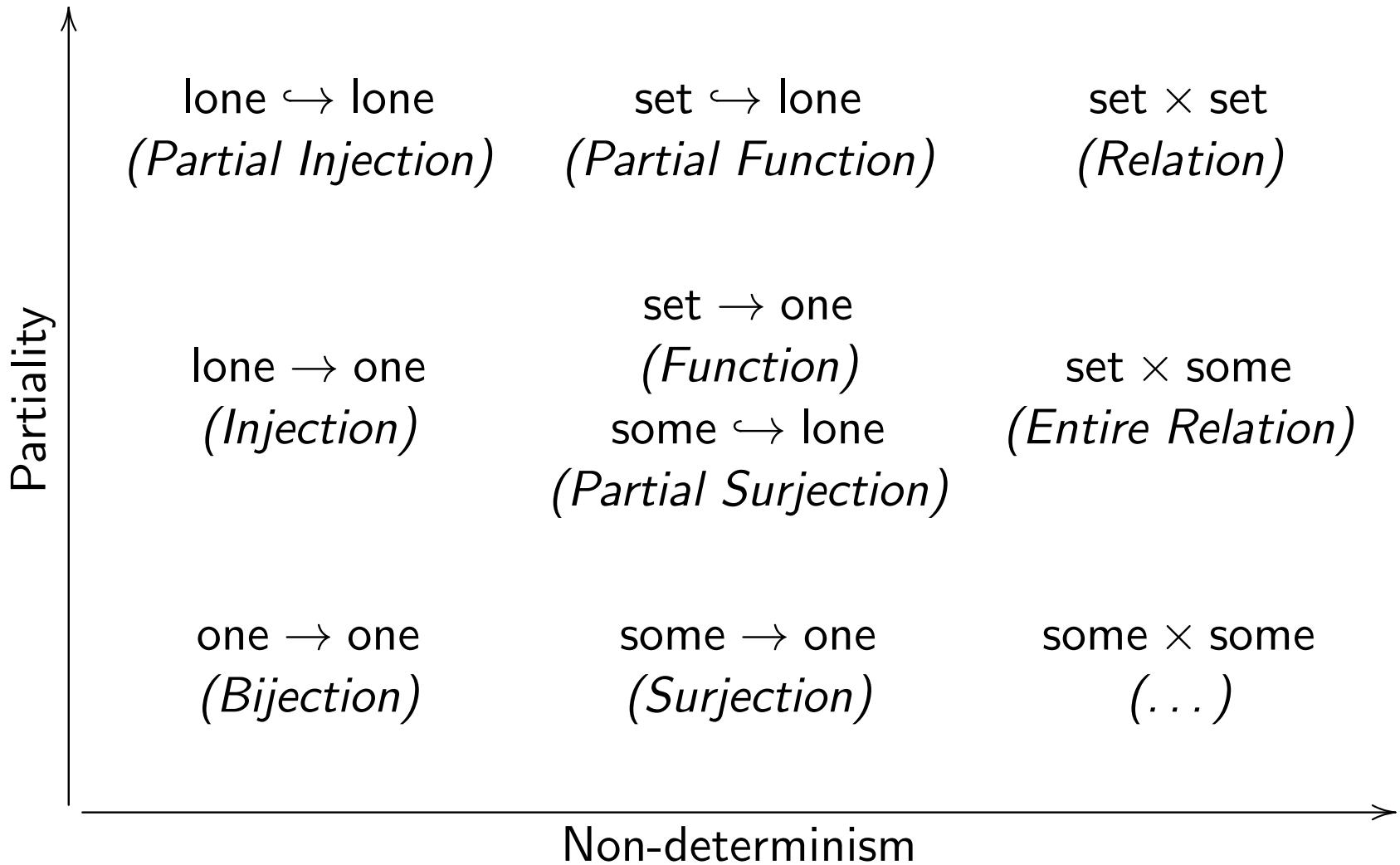
$$\frac{\mathsf{R}(a,b) \quad \mathsf{R}(a,b')}{b = b'} \qquad \mathsf{R} \subseteq A \times \mathsf{lone}\ B \qquad (\textsc{Simple})$$

$$\frac{\mathsf{R}(a,b) \quad \mathsf{R}(a',b)}{a = a'} \qquad \mathsf{R} \subseteq A\ \mathsf{lone} \times B \qquad (\textsc{Injective})$$

$$\frac{}{\exists b.R(a,b)} \qquad \mathsf{R} \subseteq A \times \mathsf{some}\ B \qquad (\textsc{Entire})$$

$$\frac{}{\exists a.R(a,b)} \qquad \mathsf{R} \subseteq A\ \mathsf{some} \times B \qquad (\textsc{Surjective})$$

The bestiary of relations



Data domain

- Shape
  - Tree
  - Graph
- Constraints
  - None
  - Order
  - Keys
  - Other

Some laws of interest

$$\frac{\overleftarrow{\mathrm{R}}(a, b') = a'}{\mathrm{R}(a', b')} \qquad \text{(Correctness)}$$

$$\frac{\mathrm{R}(a, b')}{\overleftarrow{\mathrm{R}}(a, b') = a} \qquad \text{(Stability)}$$

$$\frac{\exists a'.\mathrm{R}(a', b')}{\overleftarrow{\mathrm{R}}(a, b')\downarrow} \qquad \text{(Totality)}$$

$$\frac{\overleftarrow{\mathrm{R}}(a, b') = a' \quad \mathrm{R}(a', b)}{\overleftarrow{\mathrm{R}}(a, b) = a'} \qquad \text{(Coherence)}$$

For curators

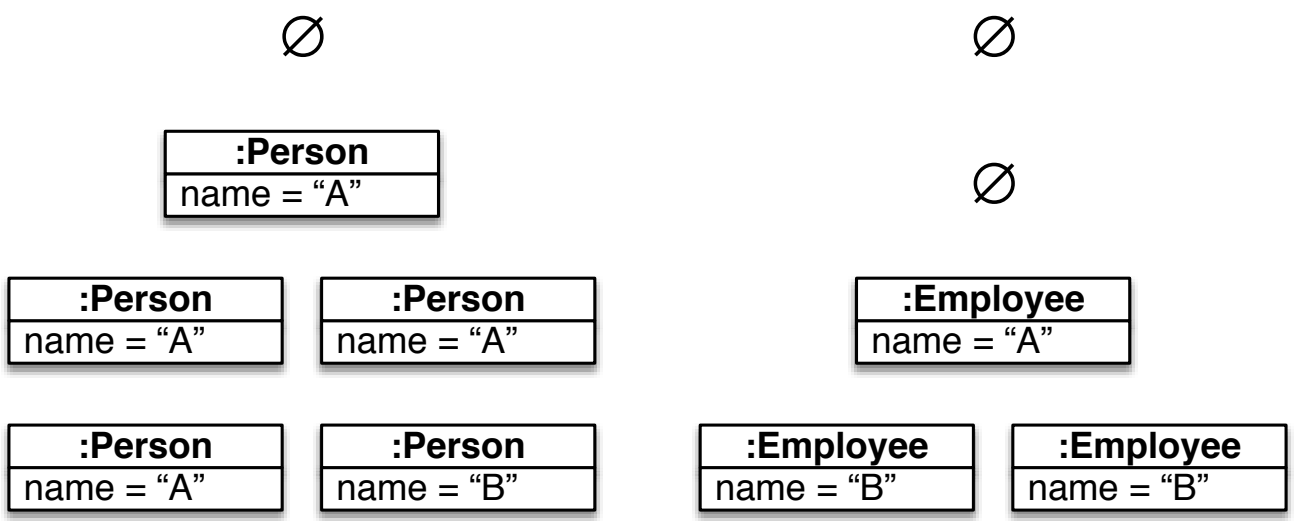
- Propose simple but illustrative examples:
  - Specify meta-models + consistency relation + unit tests.
  - UML diagrams + (hopefully) non-ambiguous natural language?
- Propose and formalize laws to be evaluated.
- Synthesize results by example (tool vs law).
- Synthesize global results (tool vs consistency vs domain).
- Publish results in a web-site (BX Wiki?).

For tool developers

- For each example:
  - Show how meta-models + consistency relation can be encoded.
  - Describe how unit tests can be implemented:
    - Show how to encode models.
    - Describe tool configuration.
    - Report results.
- Ideally, package tool + examples in a virtual machine.

Bijection #1 (one → one), Sets

Every person is an employee and vice-versa

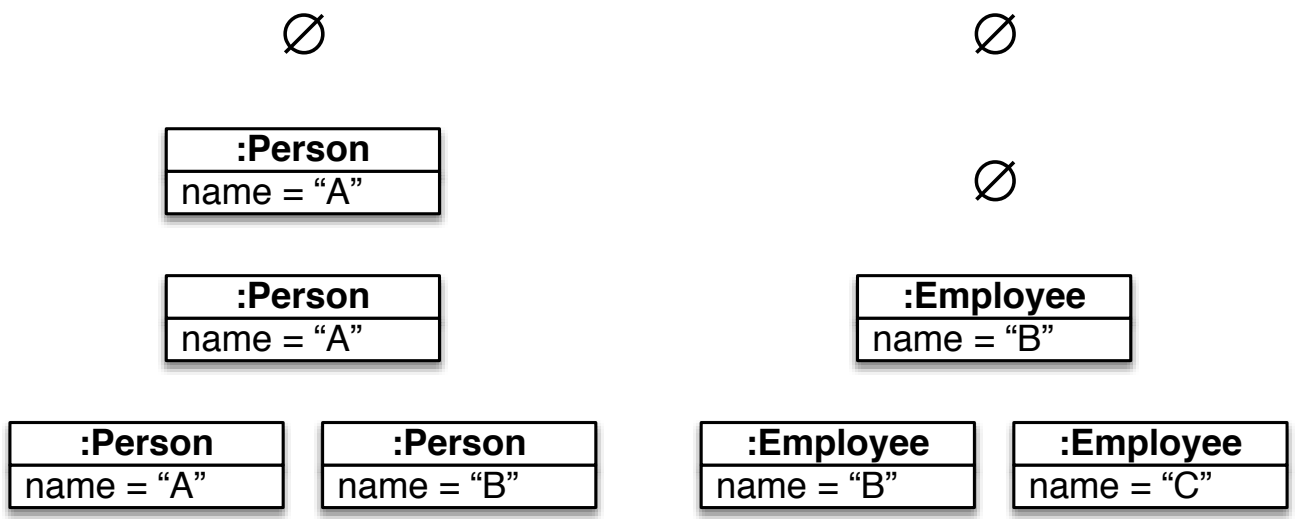


Bijection #2 (one → one), Sets with keys

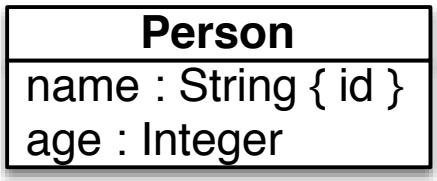
Person
name : String { id }

Employee
name : String { id }

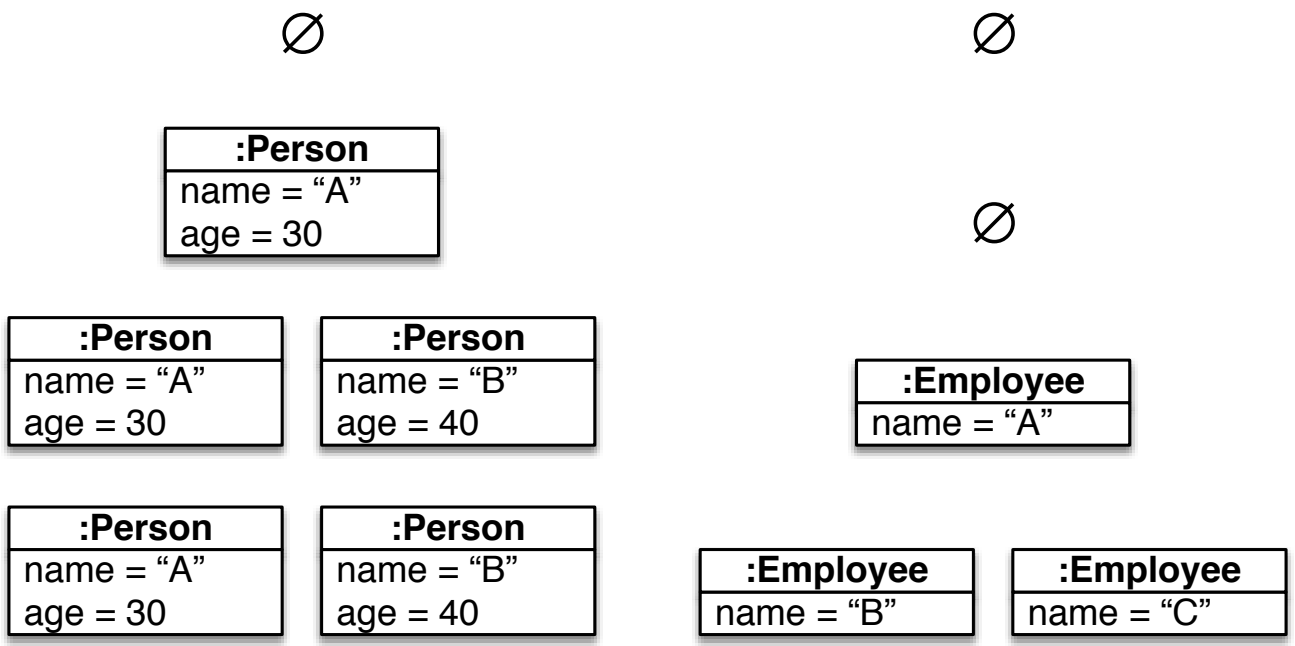
Every person is an employee and vice-versa



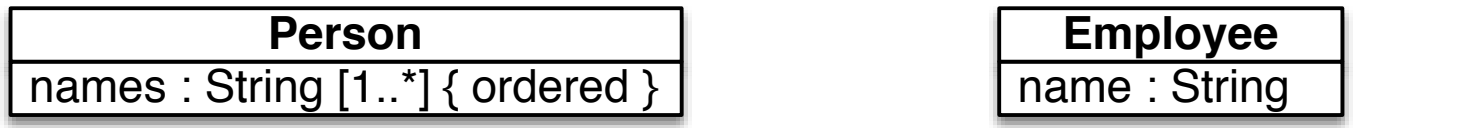
Surjection #1 (some → one), Sets with keys



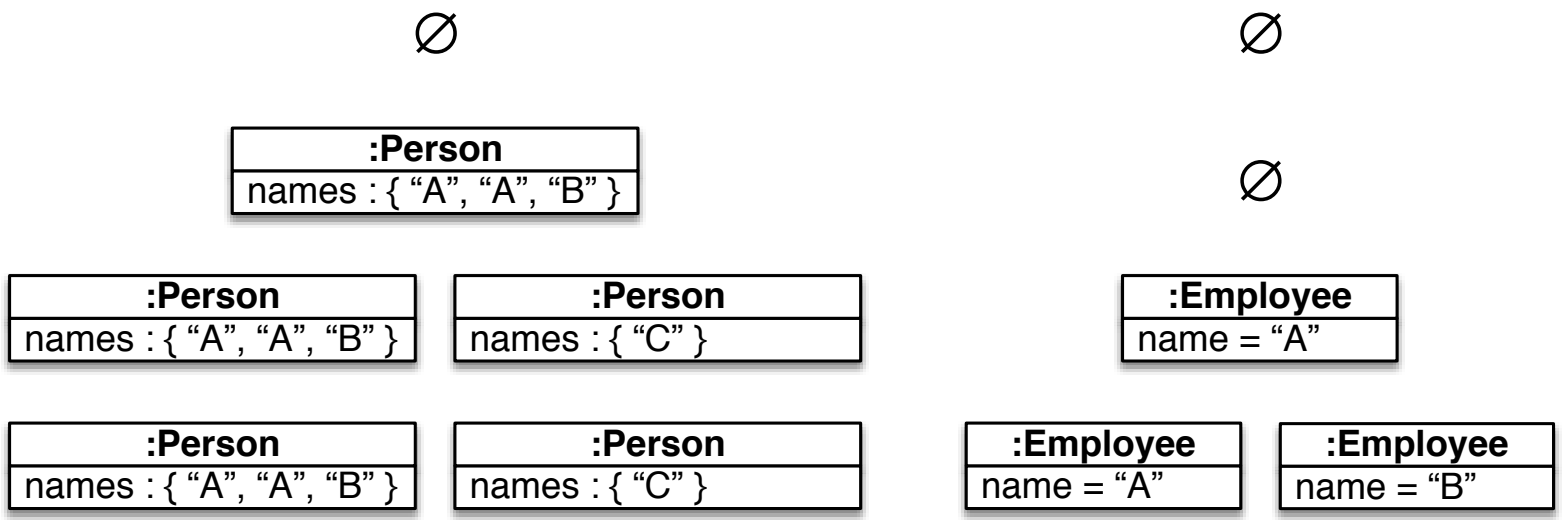
Every person is an employee and vice-versa



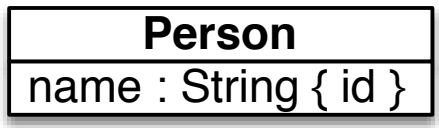
Surjection #2 (some → one), Sets with order



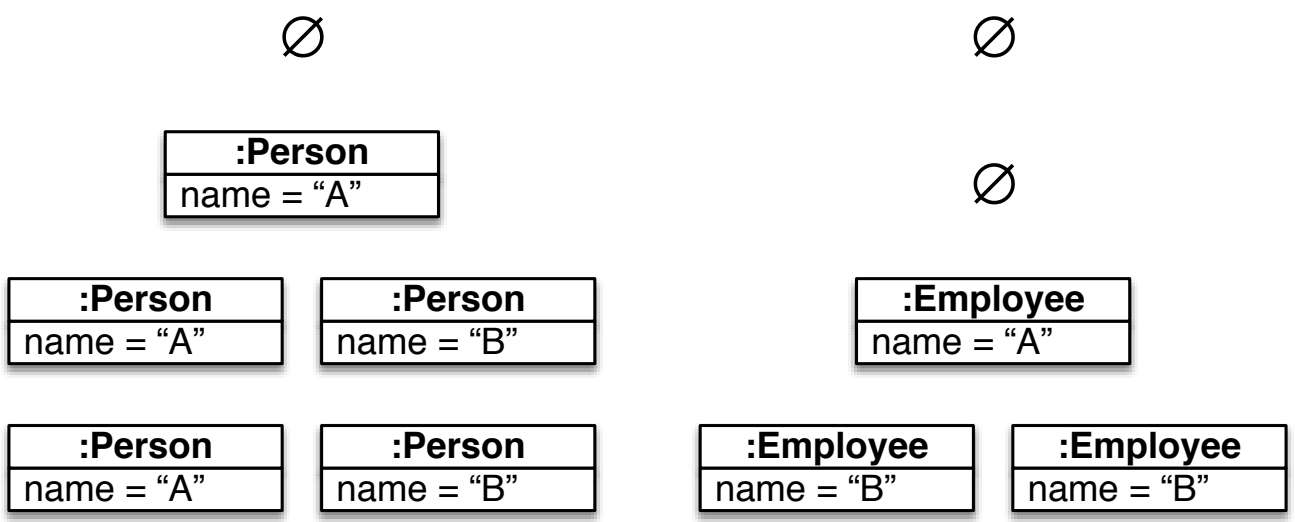
Every person is an employee with its first name and vice-versa



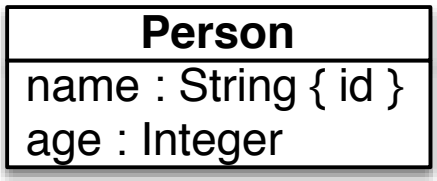
Injection #1 (lone → one), Sets with keys



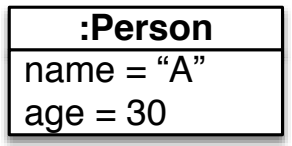
Every person is an employee and vice-versa

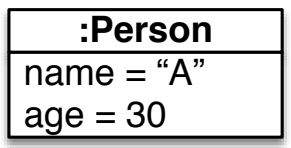


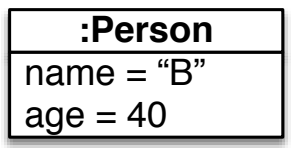
Function #1 (set → one), Sets with keys

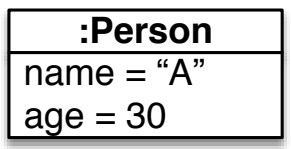


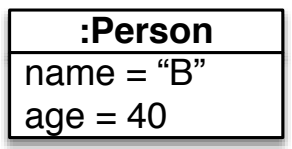
Every person is an employee and vice-versa

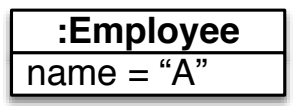


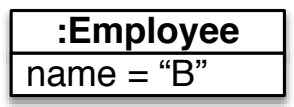


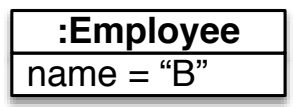




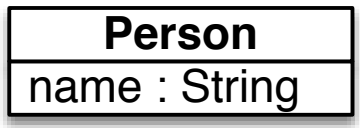






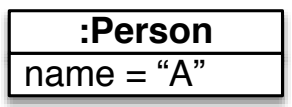


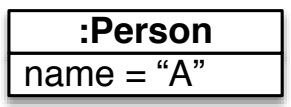
Surjective entire relation #1 (some × some), Sets

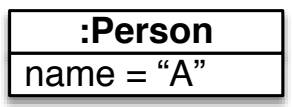


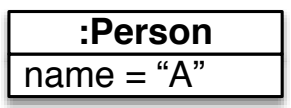
Any pair of models is consistent

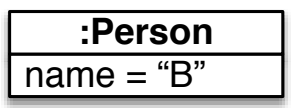
∅





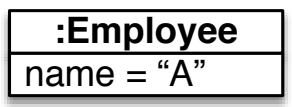


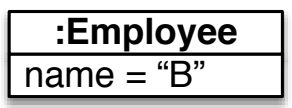


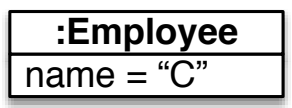


∅

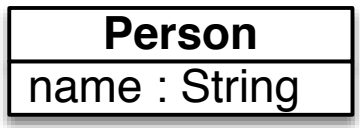
∅



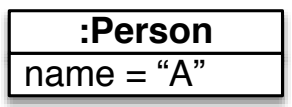


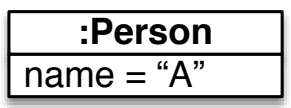


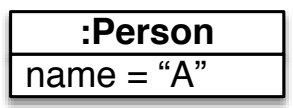
Surjective entire relation #2 (some × some), Sets

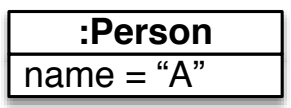


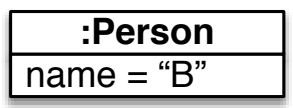
There are persons iff there are employees

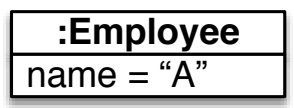


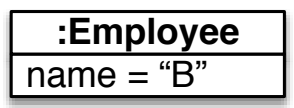


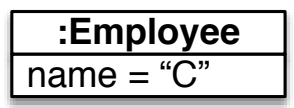












We want you!

Acknowledgements

- Perdita Stevens (inspiration, examples)
- Soichiro Hidaka (GRoundTram)
- Nuno Macedo (Echo)
- Tiago Jorge (web-site)