Light-weight Semantics with RDFS

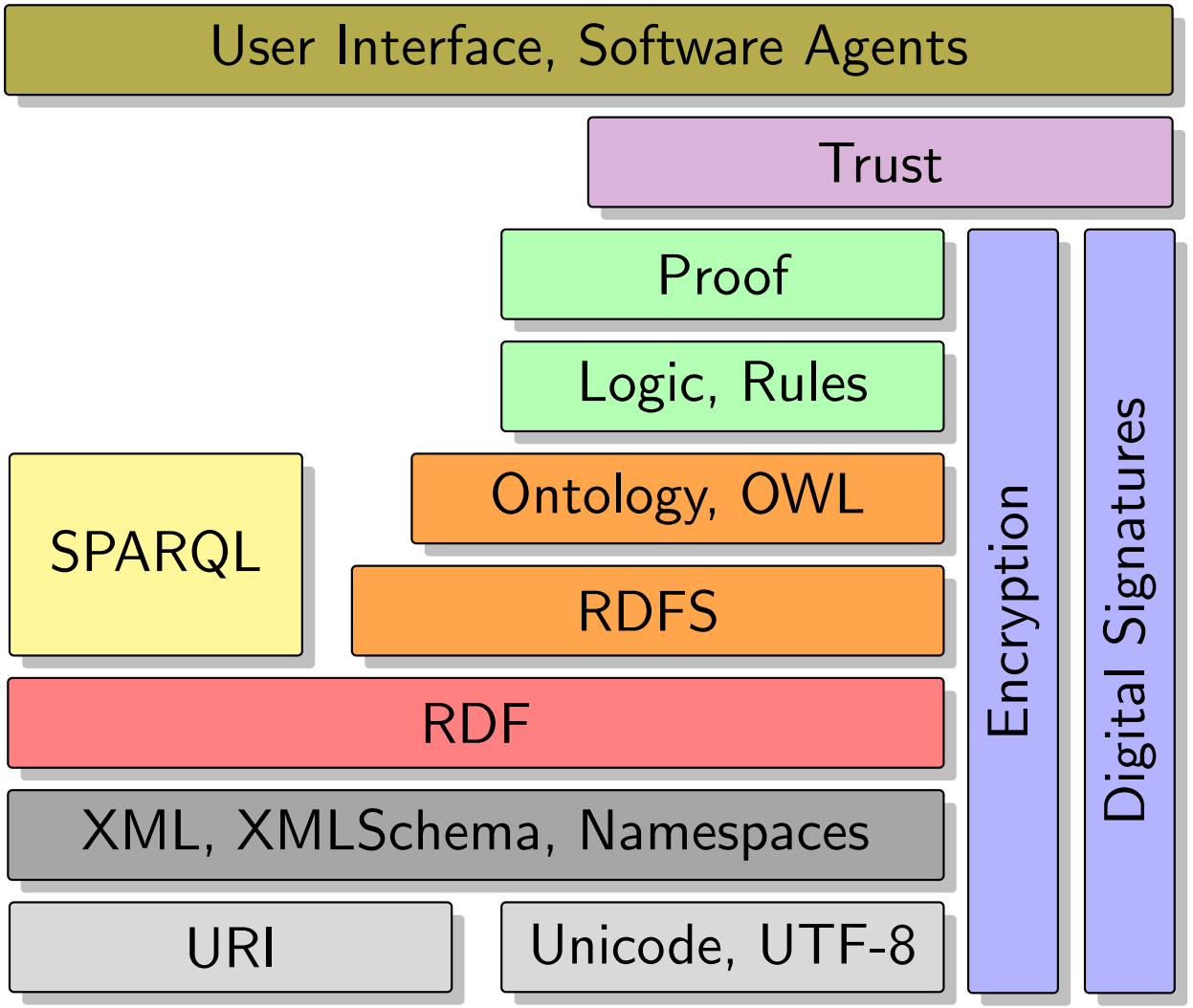
Camilo Thorne

Room 00.012
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
+49 (0) 711 685-81369
camilo.thorne@ims.uni-stuttgart.de

Semantic Web, SS 2017
(based on slides by W. Kessler)

Institut für Maschinelle Sprachverarbeitung

The Semantic Web Stack [W3C, Tim Berners-Lee]



Institut für Maschinelle Sprachverarbeitung

Outline

1. Recap
2. RDFS/RDF Schema
3. Classes and Instances
4. Properties
5. More
6. Summary
7. References

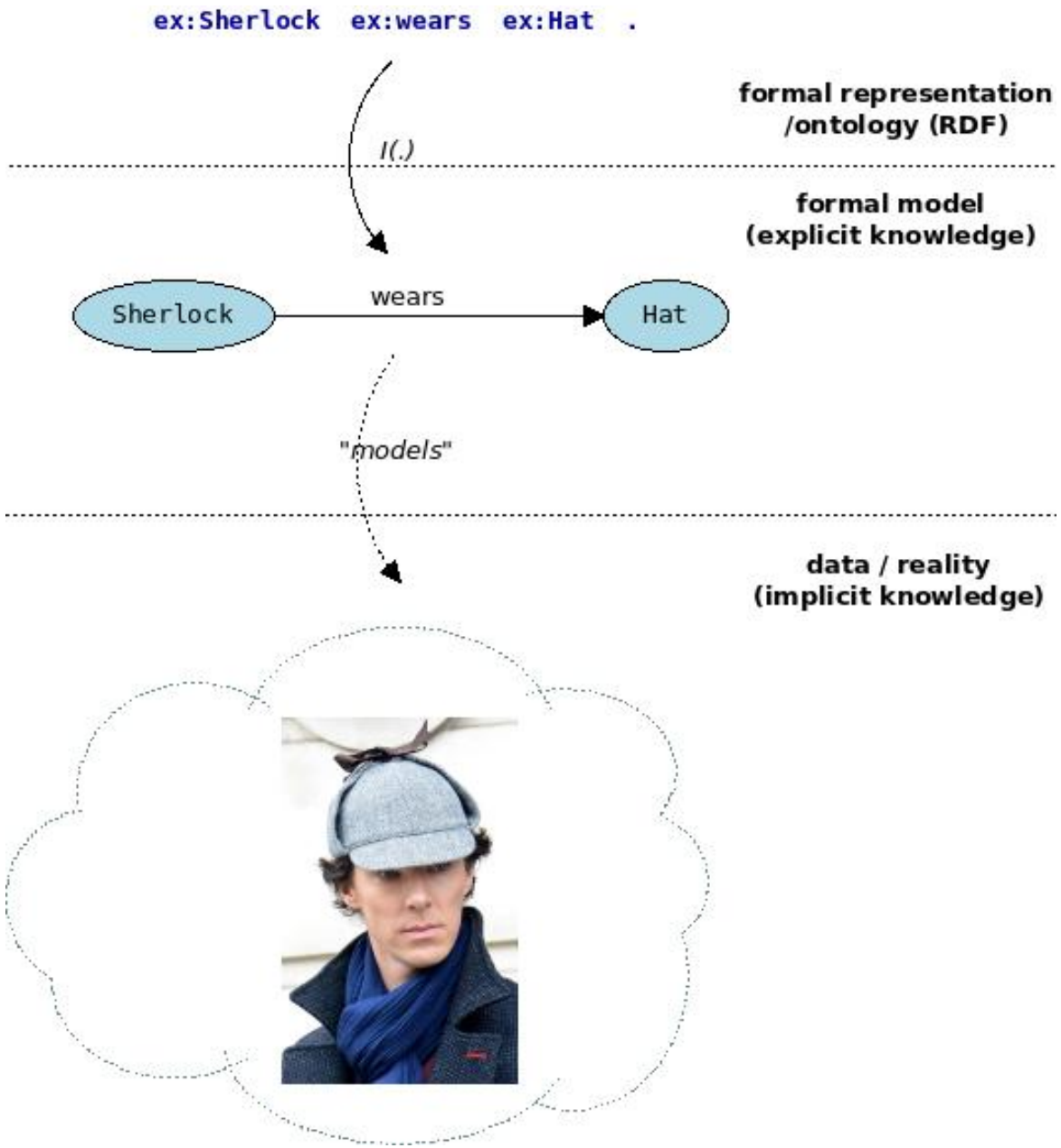
Institut für Maschinelle Sprachverarbeitung

Outline

1. Recap
2. RDFS/RDF Schema
3. Classes and Instances
4. Properties
5. More
6. Summary
7. References

Institut für Maschinelle Sprachverarbeitung

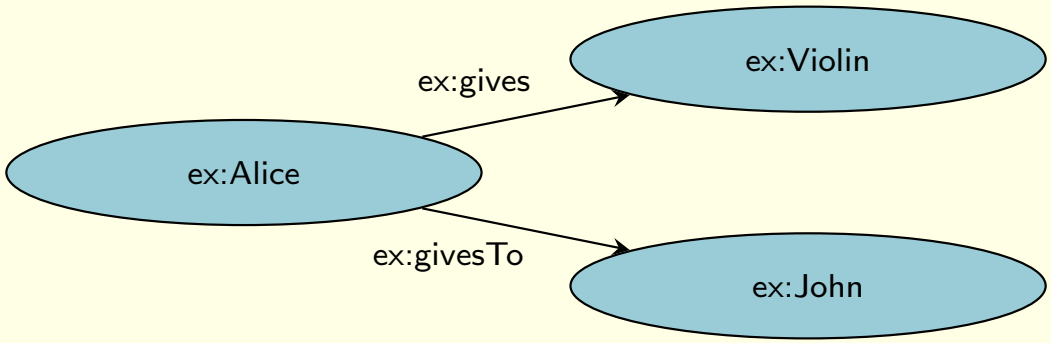
RDF Semantic Levels



Institut für Maschinelle Sprachverarbeitung

Blank Nodes: More than Binary

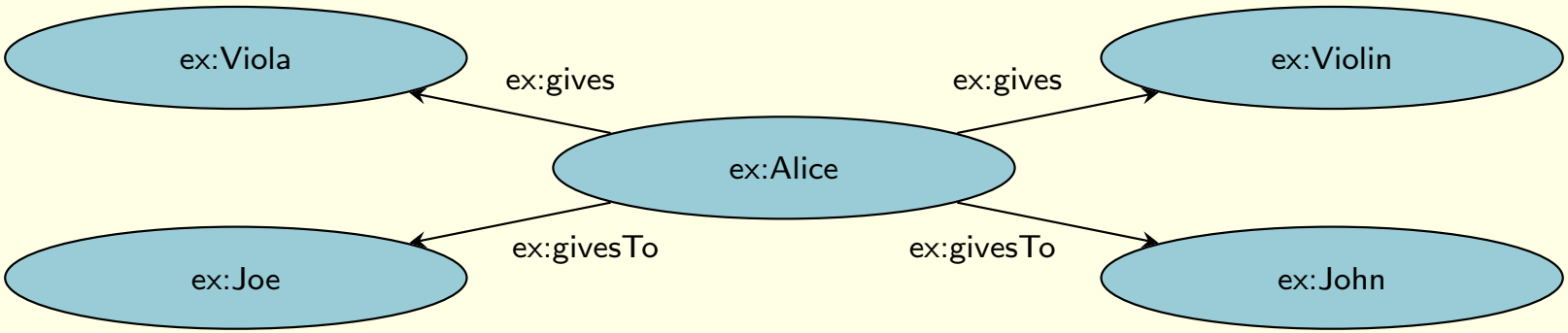
```
ex:Alice ex:gives ex:Violin .
ex:Alice ex:givesTo ex:John .
```



Blank Nodes: More than Binary

```
ex:Alice ex:gives ex:Violin .
ex:Alice ex:givesTo ex:John .

ex:Alice ex:gives ex:Viola .
ex:Alice ex:givesTo ex:Joe .
```

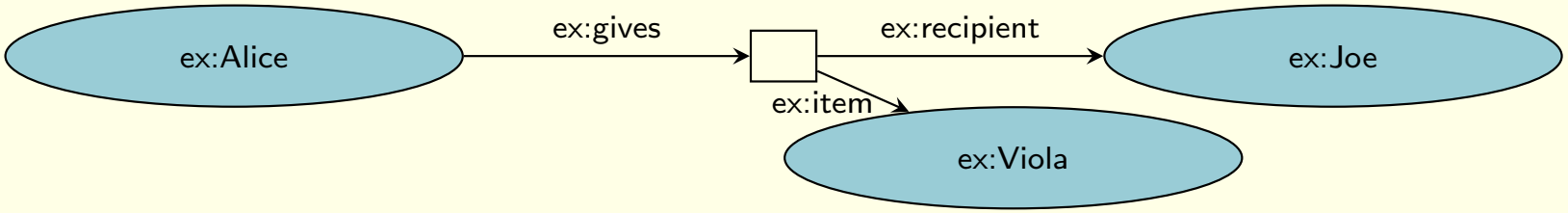


We cannot distinguish what is played where (it's a graph)!

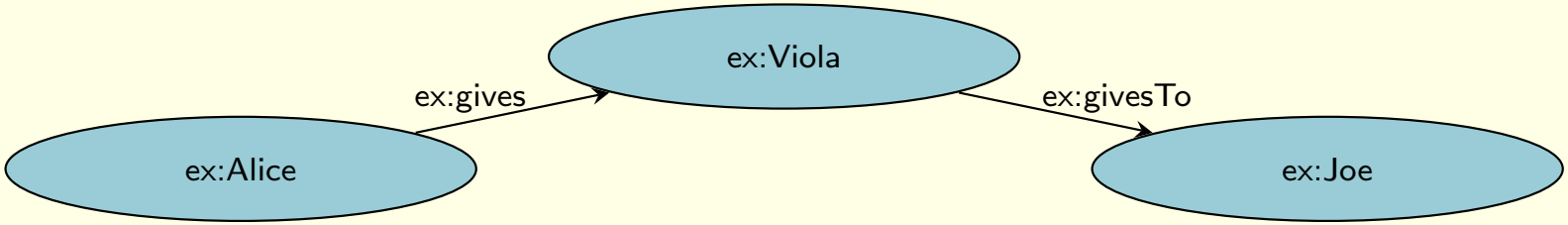
Blank Nodes: More than Binary

How would you express that **Alice** gives the **viola** to **Joe**?

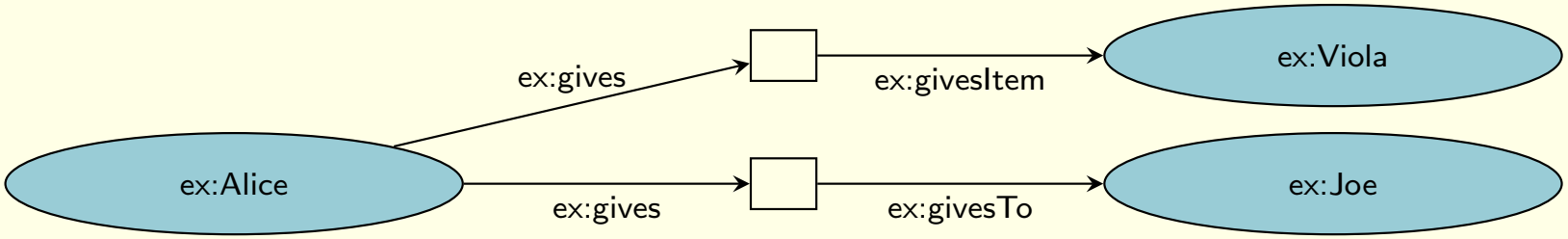
A)



B)



C)

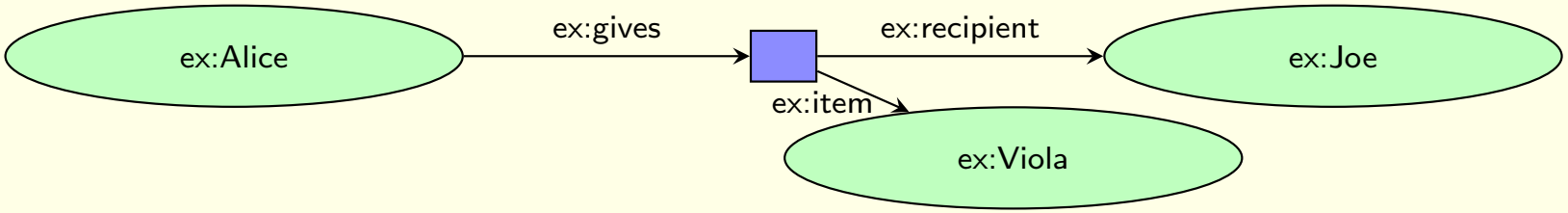


Institut für Maschinelle Sprachverarbeitung

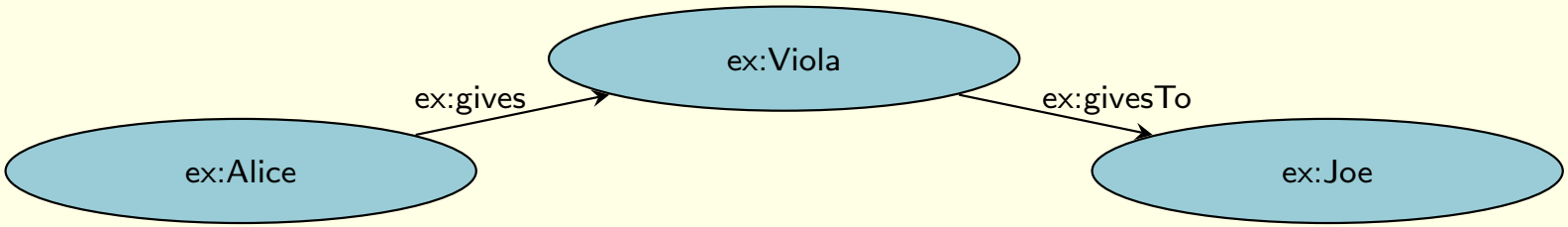
Blank Nodes: More than Binary

How would you express that **Alice** gives the **viola** to **Joe**?

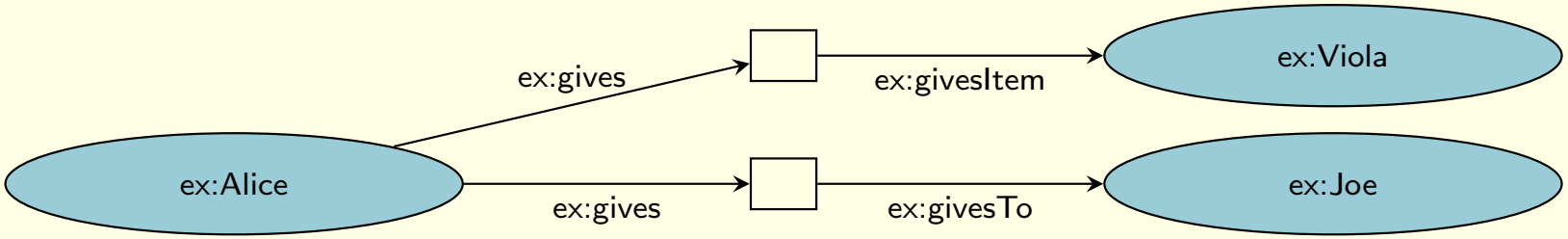
A)



B)



C)



Institut für Maschinelle Sprachverarbeitung

Exercise: Blank Nodes

Express the statements "Alice gives the violin to John" and "Alice gives the viola to Joe" in RDF using blank nodes

Compare your model and Turtle triples to those in slide 6

What do you observe? Is there a one-to-one relation between the information, the triples and the graphical model? Why?

Institut für Maschinelle Sprachverarbeitung

Outline

1 Recap

2 RDFS/RDF Schema

3 Classes and Instances

4 Properties

5 More

6 Summary

7 References

Institut für Maschinelle Sprachverarbeitung

RDFS/RDF Schema – A Vocabulary Description Language

RDF Schema

RDFS is a **shema language** used to **specify** ("constrain")

1. what kinds of resources there are
2. what relationships between resources exist
3. which properties apply to which kinds of subjects
4. what kinds of objects properties can take

- RDFS lets the user perform inference on RDF data models[1]
- RDFS does not define domain specific-vocabularies ($\neq$ specific namespaces)
- RDFS itself has a fixed standardized semantic interpretation
  - ▷ RDFS makes semantic information machine-accessible

[1] XML Schema specifies constraints over the **syntactic** structure of XML documents while RDFS describes **semantic** relations

Institut für Maschinelle Sprachverarbeitung

RDFS in Turtle

- RDFS is part of the W3C recommendation for RDF
  - ▷ RDFS is "only" an specific RDF vocabulary
- We can write RDFS in Turtle just like RDF
- RDFS namespace (usually abbreviated as `rdfs`):
  `http://www.w3.org/2000/01/rdf-schema#`

Institut für Maschinelle Sprachverarbeitung

Outline

1 Recap

2 RDFS/RDF Schema

3 Classes and Instances

4 Properties

5 More

6 Summary

7 References

Institut für Maschinelle Sprachverarbeitung

Classes and Instances

- To specify the vocabulary of a domain, we first have to define the "things" we talk about
- There are **two sorts** of "things":

Instances:	individual objects like "Theo" or "John Smith"
Classes:	types of objects like "cat" or "person"

Institut für Maschinelle Sprachverarbeitung

Classes and Instances

- To specify the vocabulary of a domain, we first have to define the "things" we talk about
- There are **two sorts** of "things":

Instances:	individual objects like "Theo" or "John Smith"
Classes:	types of objects like "cat" or "person"

- A **class** is a template, a description of what something has to fulfill to belong in this class
- An **instance** is a specific member of this class.

Institut für Maschinelle Sprachverarbeitung

Relations between Classes and Instances (1)

- If an individual objects is a member (instance) of some class, it is of the **type** defined by the class template
- Something can be an instance of multiple classes, e.g., "John Smith" can be teacher, pet owner and soccer fan
- A statement like "John Smith is a teacher" implicitly says that "John Smith" is an instance and "teacher" is a class

Institut für Maschinelle Sprachverarbeitung

Relations between Classes and Instances (1)

- If an individual objects is a member (instance) of some class, it is of the **type** defined by the class template
- Something can be an instance of multiple classes, e.g., "John Smith" can be teacher, pet owner and soccer fan
- A statement like "John Smith is a teacher" implicitly says that "John Smith" is an instance and "teacher" is a class

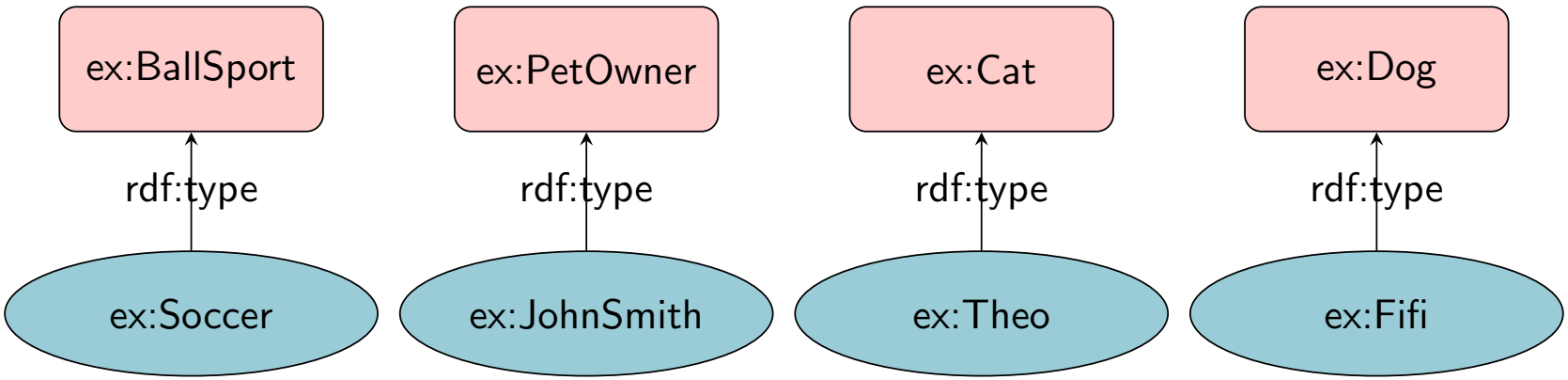
▷ RDFS formalizes these intuitions

Institut für Maschinelle Sprachverarbeitung

Relations between Classes and Instances (2)

```
ex:JohnSmith rdf:type ex:PetOwner .
ex:PetOwner rdf:type rdfs:Class .
```

- The relation between an instance and the class it belongs to is expressed in RDF by `rdf:type`
- To specify that something is a class, the type `rdfs:Class` ("the class of all classes") can be assigned



Institut für Maschinelle Sprachverarbeitung

Relations between Classes (1)

- In a domain vocabulary, classes have relations to each other
- The most common relation between classes in a domain vocabulary is hierarchy: one class is a **subclass** of the other
- Properties of the subclass relation:
  1. **inheritance:** instances of a subclass are also instances of the parent class: if "Theo" is a "cat" and "cat" is an "animal", then "Theo" is an "animal"
  2. **transitivity:** if "cat" is a "mammal" and "mammal" is an "animal", then "cat" is an "animal".
  3. **reflexivity:** a class is subclass of itself

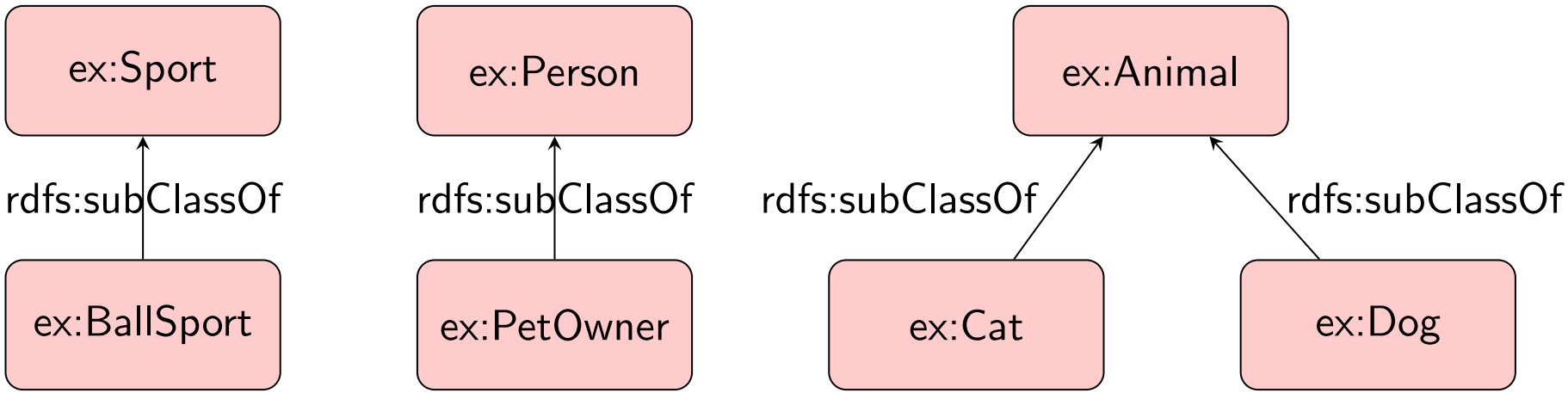
Institut für Maschinelle Sprachverarbeitung

Relations between Classes (2)

The subclass relation is specified by `rdfs:subClassOf`:

```
ex:PetOwner rdfs:subClassOf ex:Person .
```

Graphically, this yields:



Institut für Maschinelle Sprachverarbeitung

RDFS Entailment

- RDFS extends RDF semantics not only by explicitly typing resources, but also by specifying what statements can be inferred from other statements
  - ▷ logical **entailment**
- Informally, there is (roughly) an entailment **rule**
  - for each syntactic element in RDFS
  - for each property of such syntactic element

Institut für Maschinelle Sprachverarbeitung

RDFS Entailment

- RDFS extends RDF semantics not only by explicitly typing resources, but also by specifying what statements can be inferred from other statements
  - ▷ logical **entailment**
- Informally, there is (roughly) an entailment **rule**
  - for each syntactic element in RDFS
  - for each property of such syntactic element

Remark: We will not study the formal, model-theoretic semantics of RDF(S) in the course of these lectures, but discuss rather its main intuitions; for the math, see [HKR09]

Institut für Maschinelle Sprachverarbeitung

Entailment Rule (1) for `rdfs:subClassOf` (= Inheritance)

- **Example:** from the statements

```
ex:JohnSmith rdf:type ex:PetOwner .
ex:PetOwner rdfs:subClassOf ex:Person .
```

we can infer

```
ex:JohnSmith rdf:type ex:Person .
```

Institut für Maschinelle Sprachverarbeitung

Entailment Rule (1) for `rdfs:subClassOf` (= Inheritance)

- **Example:** from the statements

```
ex:JohnSmith rdf:type ex:PetOwner .
ex:PetOwner rdfs:subClassOf ex:Person .
```

  we can infer

```
ex:JohnSmith rdf:type ex:Person .
```

- For `rdfs:subClassOf`, the **inheritance** rule is (informal description):

Inheritance

If we have the statements

```
A rdfs:subClassOf B .
X rdf:type A .
```

we can infer that

```
X rdf:type B .
```

Institut für Maschinelle Sprachverarbeitung

Entailment Rule (2) for `rdfs:subClassOf` (= Transitivity)

- **Example:** from the statements

```
ex:PetOwner rdfs:subClassOf ex:Owner .
ex:Owner rdfs:subClassOf ex:Person .
```

we can infer

```
ex:PetOwner rdfs:subClassOf ex:Person .
```

Institut für Maschinelle Sprachverarbeitung

Entailment Rule (2) for `rdfs:subClassOf` (= Transitivity)

- **Example:** from the statements

```
ex:PetOwner rdfs:subClassOf ex:Owner .
ex:Owner rdfs:subClassOf ex:Person .
```

  we can infer

```
ex:PetOwner rdfs:subClassOf ex:Person .
```

- For `rdfs:subClassOf`, the **transitivity** rule is (informal description):

Transitivity

If we have the statements

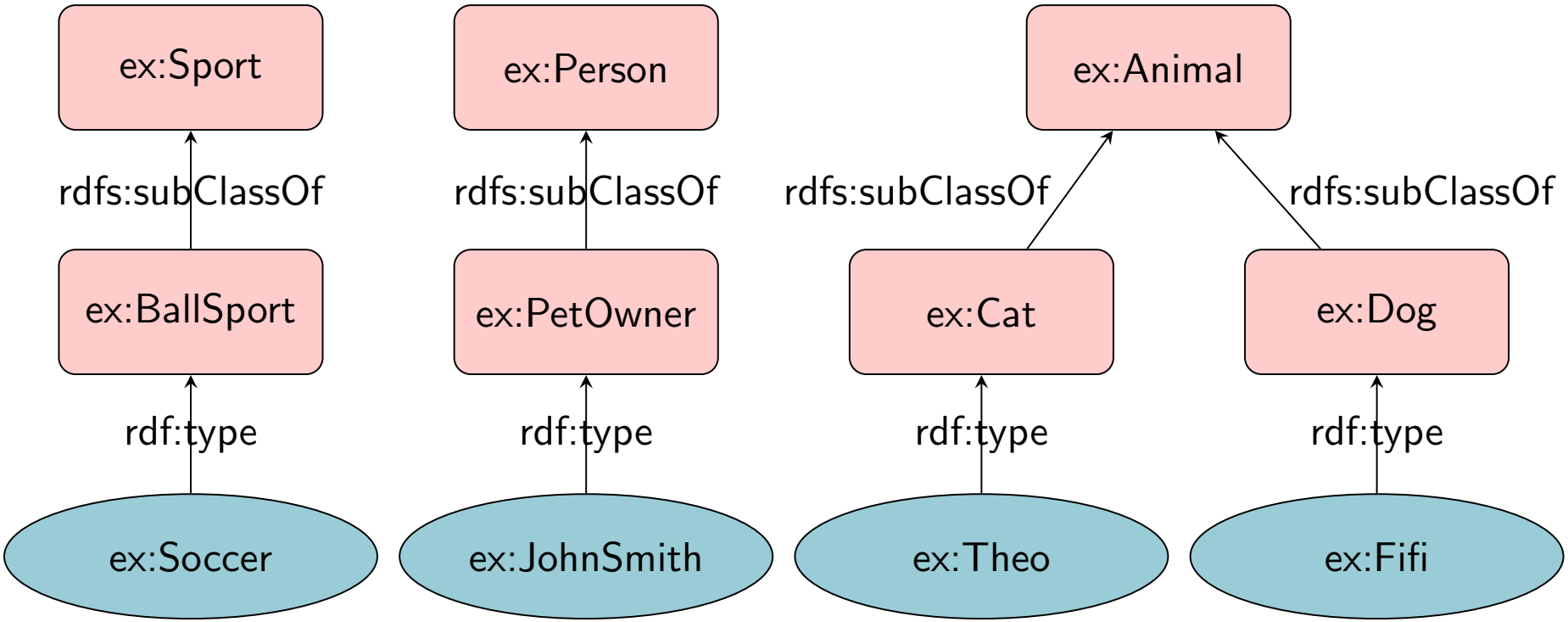
```
A rdfs:subClassOf B .
X rdfs:subClassOf A .
```

we can infer that

```
X rdfs:subClassOf B .
```

Institut für Maschinelle Sprachverarbeitung

RDFS – Example as Graph



Institut für Maschinelle Sprachverarbeitung

RDFS – Example as Triples

- Specify relationships between resources (instances / classes):

```
ex:JohnSmith rdf:type ex:PetOwner .
ex:Theo rdf:type ex:Cat .
ex:Fifi rdf:type ex:Dog .
ex:Soccer rdf:type ex:BallSport .
```

- Specify relationships between resources (classes):

```
ex:PetOwner rdfs:subClassOf ex:Person .
ex:Cat rdfs:subClassOf ex:Animal .
ex:Dog rdfs:subClassOf ex:Animal .
ex:BallSport rdfs:subClassOf ex:Sport .
```

Institut für Maschinelle Sprachverarbeitung

RDFS – Example as Triples

- Specify relationships between resources (instances / classes):

```
ex:JohnSmith rdf:type ex:PetOwner .
ex:Theo rdf:type ex:Cat .
ex:Fifi rdf:type ex:Dog .
ex:Soccer rdf:type ex:BallSport .
```

- Specify relationships between resources (classes):

```
ex:PetOwner rdfs:subClassOf ex:Person .
ex:Cat rdfs:subClassOf ex:Animal .
ex:Dog rdfs:subClassOf ex:Animal .
ex:BallSport rdfs:subClassOf ex:Sport .
```

- Infer entailed relationships:

```
ex:JohnSmith rdf:type ex:Person .
ex:Theo rdf:type ex:Animal .
ex:Fifi rdf:type ex:Animal .
ex:Soccer rdf:type ex:Sport .
```

Institut für Maschinelle Sprachverarbeitung

Quiz: RDFS Classes and Instances

```
ex:R2D2 ex:knows ex:C3PO .
ex:C3PO rdf:type ex:Robot .
ex:Chewbacca ex:likes ex:HanSolo .
ex:Jediism rdfs:subClassOf ex:Religion .
```

Given the above example, which resources are classes?

A) `ex:R2D2`
B) `ex:C3PO`
C) `ex:likes`
D) `ex:Robot`
E) `ex:Chewbacca`
F) `ex:HanSolo`
G) `ex:Jediism`
H) `ex:Religion`

Institut für Maschinelle Sprachverarbeitung

Outline

1 Recap

2 RDFS/RDF Schema

3 Classes and Instances

4 Properties

5 More

6 Summary

7 References

Institut für Maschinelle Sprachverarbeitung

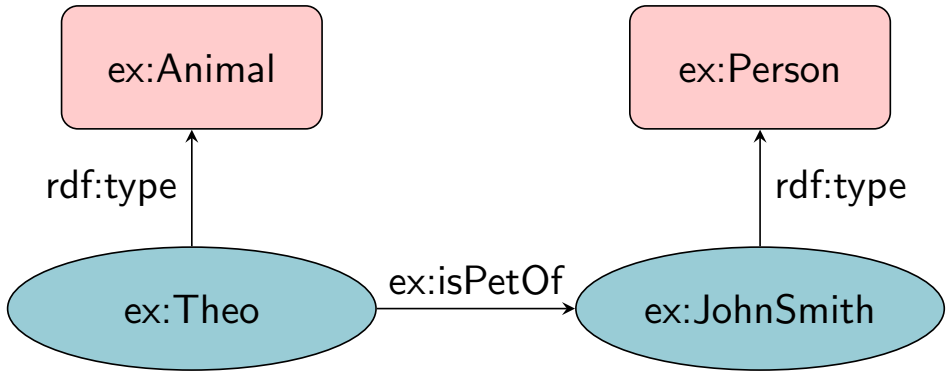
Generalizing Properties

- A statement like `ex:Theo ex:isPetOf ex:JohnSmith .` describes a relation between two specific instances
- We want to clarify that only animals can be the pet of someone and only people can have pets
- In other words, if we have a statement

```
A ex:isPetOf B .
```

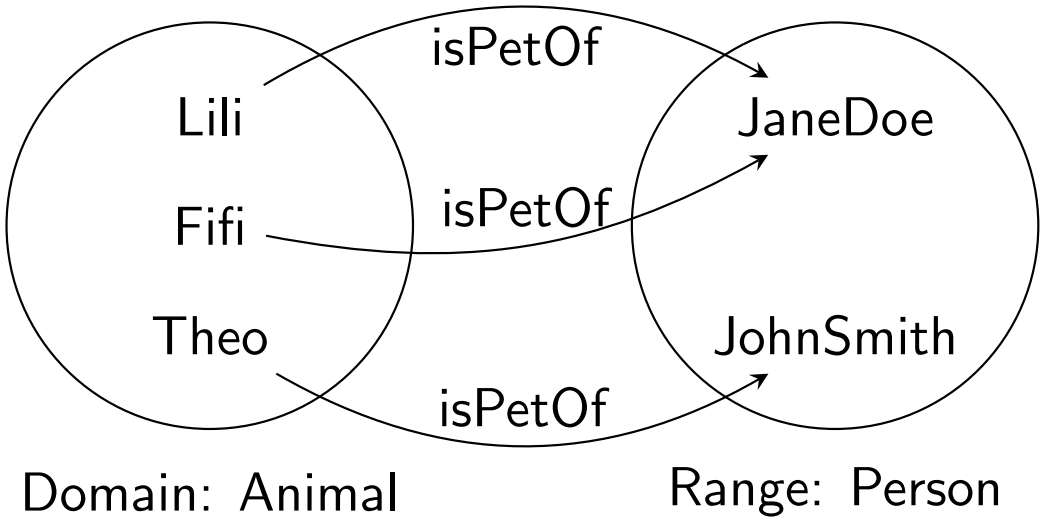
we want to assert that

```
A rdf:type ex:Animal .
B rdf:type ex:Person .
```



Institut für Maschinelle Sprachverarbeitung

Domain and Range of Properties (1)



- The **domain** of a property p specifies to which kinds of resources the property applies, i.e., the type of resources that may appear as subjects in a triple with p as predicate
- The **range** of a property p specifies what values the property can take, i.e., the type of resources that may appear as objects in a triple with p as a predicate

Institut für Maschinelle Sprachverarbeitung

Domain and Range of Properties (2)

```
ex:isPetOf rdf:type rdf:Property .
ex:isPetOf rdfs:domain ex:Animal .
ex:isPetOf rdfs:range ex:Person .
```

- Properties are of the class `rdf:Property`
- `rdfs:domain` and `rdfs:range` can be specified
- If the domain [range] is not specified, any resource can be the subject [object]

Remark: Several classes can be declared as domain [range], this means that any subject [object] must be an instance of all classes (i.e., in the intersection of classes)

Institut für Maschinelle Sprachverarbeitung

Quiz: Generalizing Properties

What is the semantics of the following statements?

```
ex:isPetOf rdf:type rdf:Property .
ex:isPetOf rdfs:domain ex:Animal .
ex:isPetOf rdfs:range ex:Person .
```

A) There is a link called `ex:isPetOf` between the two resources `ex:Animal` and `ex:Person`

B) There is a link called `ex:isPetOf` between an instance of the class `ex:Animal` and an instance of the class `ex:Person`

C) The subject of a statement with the predicate `ex:isPetOf` must always be `ex:Animal` and the object must be `ex:Person`

D) The subject of a statement with the predicate `ex:isPetOf` must always be an instance of the class `ex:Animal` and the object must be an instance of the class `ex:Person`

Institut für Maschinelle Sprachverarbeitung

Domain/Range vs. RDF statements

- **RDF** statements link two specific instances:

  ```
  ex:Theo ex:isPetOf ex:JohnSmith .
  ```

  The specific instance Theo has a relation called `ex:isPetOf` with the specific instance John Smith

  ▷ `ex:Theo` and `ex:JohnSmith` are instances

- **RDFS** domain (range) gives information about the possible subjects (objects) of RDF statements with this property:

  ```
  ex:isPetOf rdfs:domain ex:Animal .
  ex:isPetOf rdfs:range ex:Person .
  ```

  Anything occurring as subject in a statement with `ex:isPetOf` as a property has to be an animal, any object a person

  ▷ `ex:Animal` and `ex:Person` are classes

Institut für Maschinelle Sprachverarbeitung

Entailment Rules for `rdfs:domain` / `rdfs:range`

- **Example:** From the statements

  ```
  ex:isPetOf rdfs:domain ex:Animal .
  ex:Theo ex:isPetOf ex:JohnSmith .
  ```

  we can infer that the **subject** of the statement (`ex:Theo`) is an instance of the class given as the domain (`ex:Animal`)

  ```
  ex:Theo rdf:type ex:Animal .
  ```

Institut für Maschinelle Sprachverarbeitung

Entailment Rules for `rdfs:domain` / `rdfs:range`

- **Example:** From the statements

```
ex:isPetOf rdfs:domain ex:Animal .
ex:Theo ex:isPetOf ex:JohnSmith .
```

  we can infer that the **subject** of the statement (`ex:Theo`) is an instance of the class given as the domain (`ex:Animal`)

```
ex:Theo rdf:type ex:Animal .
```

- For `rdfs:domain` and `rdfs:range` the rules are (informal description):

Domain and Range

If we have the statements `p rdfs:domain A` and `X p Y`, we can infer that `X rdf:type A`

If we have the statements `p rdfs:range B` and `X p Y`, we can infer that `Y rdf:type B`

Institut für Maschinelle Sprachverarbeitung

Property Hierarchies

```
ex:isPetCatOf rdfs:subPropertyOf ex:isPetOf .
```

- Properties can have **sub-properties** specified with `rdfs:subPropertyOf`
- Similar rules as for `rdfs:subClassOf` apply (inheritance, transitivity)
- **Example** Given

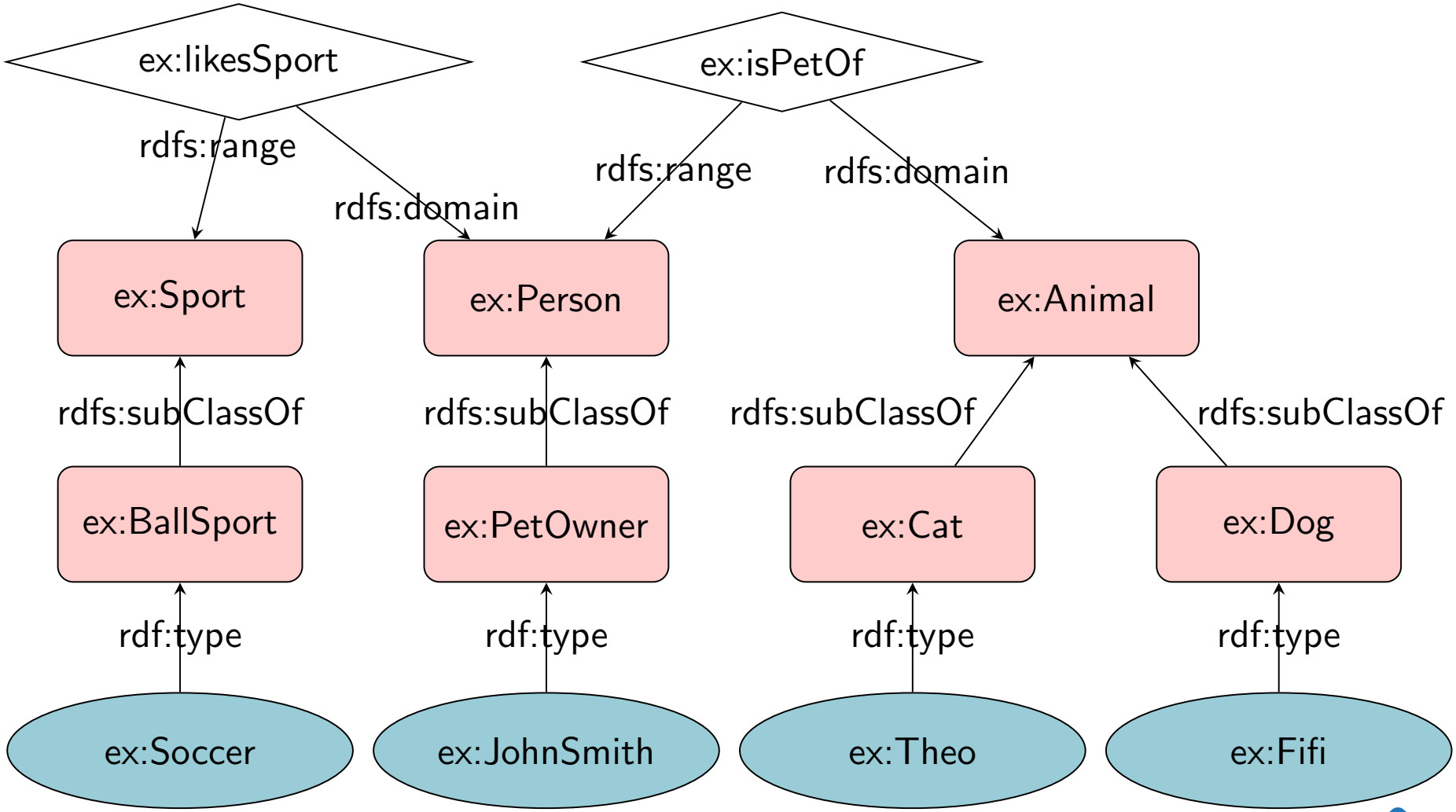
  ```
  ex:Theo ex:isPetCatOf ex:JohnSmith .
  ```

  we can infer

  ```
  ex:Theo ex:isPetOf ex:JohnSmith .
  ```

Institut für Maschinelle Sprachverarbeitung

RDFS – Example as Graph



Institut für Maschinelle Sprachverarbeitung

RDFS – Example as Triples

- Specify relationships between resources (properties):

```
ex:isPetOf rdf:type rdf:Property .
ex:isPetCatOf rdfs:subPropertyOf ex:isPetOf .
```

- Specify which properties apply to which kinds of resources:

```
ex:isPetOf rdfs:domain ex:Animal .
ex:likesSport rdfs:domain ex:Person .
```

- Specify what values properties can take:

```
ex:isPetOf rdfs:range ex:Person .
ex:likesSport rdfs:range ex:Sport .
```

Institut für Maschinelle Sprachverarbeitung

Quiz: RDFS Properties

Which statements correctly express that cows eat grass?

A)
```
ex:eats ex:Cow ex:Grass .
```

B)
```
ex:eats rdfs:domain ex:Grass .
ex:eats rdfs:range ex:Cow .
```

C)
```
ex:Cow ex:eats ex:Grass .
```

D)
```
ex:eats rdfs:domain ex:Cow .
ex:eats rdfs:range ex:Grass .
```

Institut für Maschinelle Sprachverarbeitung

Outline

1 Recap

2 RDFS/RDF Schema

3 Classes and Instances

4 Properties

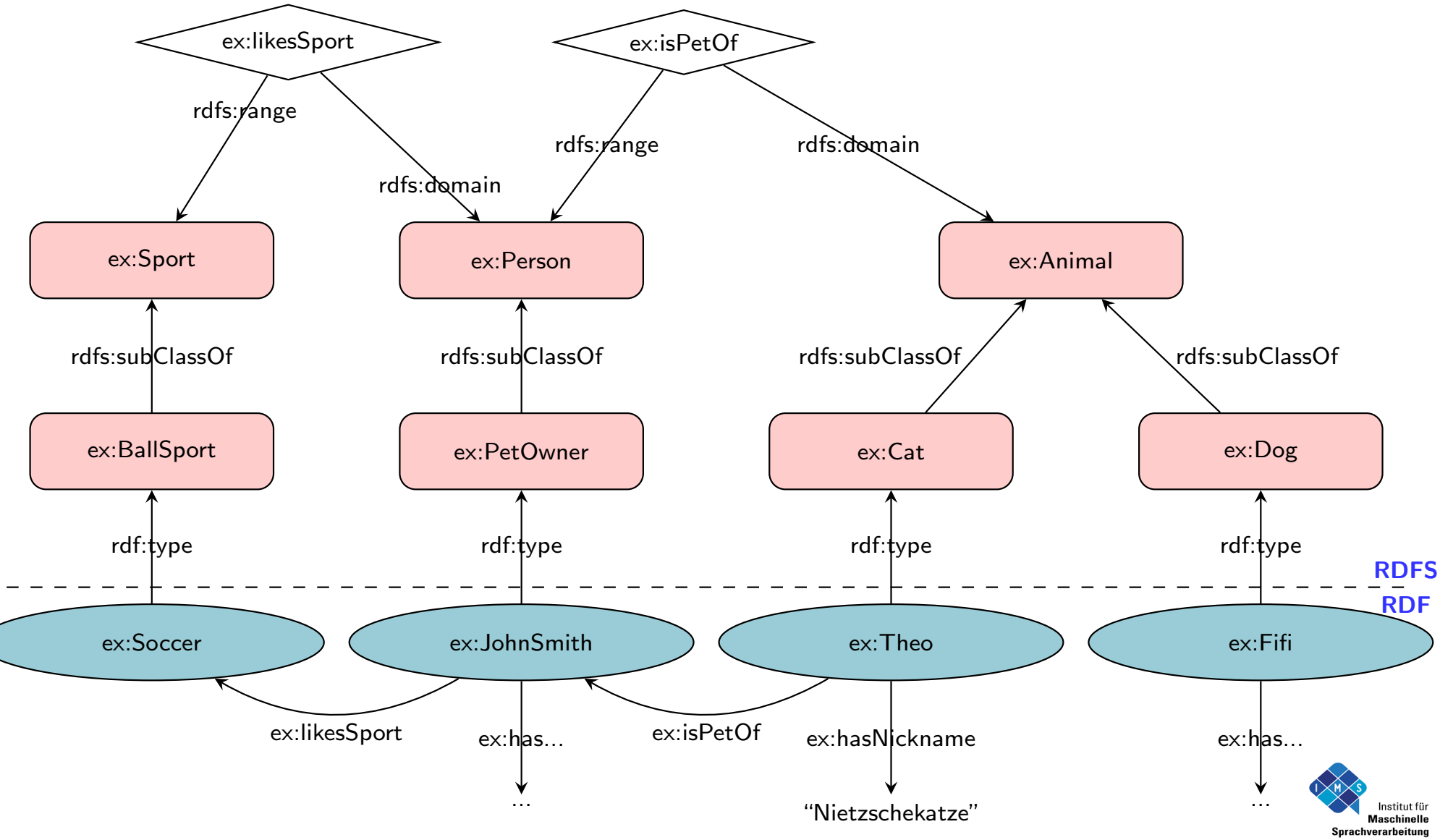
5 More

6 Summary

7 References

Institut für Maschinelle Sprachverarbeitung

The Relation of RDF and RDFS



Institut für Maschinelle Sprachverarbeitung

Quiz: Semantics

- Given the following statements:

```
ex:FruitFlyAnton rdf:type ex:FruitFly .
ex:BoxA ex:contains ex:FruitFlyAnton .
ex:contains rdfs:range ex:Fruit .
```

- What is the relation of `ex:FruitFlyAnton` and `ex:Fruit`?
- What about `ex:FruitFly` and `ex:Fruit`?
- Can you infer from this how domain and range are used for inference in RDFS in general?

Institut für Maschinelle Sprachverarbeitung

Outline

1 Recap

2 RDFS/RDF Schema

3 Classes and Instances

4 Properties

5 More

6 Summary

7 References

Institut für Maschinelle Sprachverarbeitung

Summary: Light-weight Semantics with RDFS

- RDFS lets the user define the **semantics** of the vocabulary used in RDF data models
- Distinguishes three types of resources:

instances	individual objects
classes	types of objects, templates (`rdfs:Class`)
properties	describe relations between resources `rdfs:Property`

- Introduces logical relations with a fixed semantics:

`rdf:type`	type of an instance (class membership)
`rdfs:subClassOf`	defines a class hierarchy
`rdfs:subPropertyOf`	defines a property hierarchy
`rdfs:domain`	the class of possible subjects for the property
`rdfs:range`	the class of possible objects for the property

Institut für Maschinelle Sprachverarbeitung

Outline

1 Recap

2 RDFS/RDF Schema

3 Classes and Instances

4 Properties

5 More

6 Summary

7 References

Institut für Maschinelle Sprachverarbeitung

Suggested Reading

Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph and York Sure. Semantic Web. Grundlagen. Springer textbook, 2008. (Chapter 4)

Pascal Hitzler, Markus Krötzsch and Sebastian Rudolph. Foundations of Semantic Web Technologies. Chapman & Hall/CRC, 2009. (Chapter 3)

Institut für Maschinelle Sprachverarbeitung