

OWL

86

86

Restriction Class

- Define new classes in terms of existing classes
 - By restriction of those classes
- Example: An active faculty member is one who is teaching at least one course or has at least one research grant

87

87

Restrictions

- A **restriction** is a class that is defined by a description of its members in terms of its existing properties and classes
 - `owl:Restriction` `rdfs:subClassOf` `owl:Class`
- Restriction definition based on:
 - A property P of instances in the restriction class
 - A restriction on the range R of the property
 - $C = \{ x \mid (x P y) \text{ for } y \in R \}$

88

88

Restrictions

- $C = \{ x \mid (x P y) \text{ for } y \in R \}$
- `owl:onProperty`: specify the property that is used in the definition of the restriction class
- Forms of restrictions:
 - `owl:someValuesFrom`
 - `owl:allValuesFrom`
 - `owl:hasValue`

89

89

Restrictions (2)

- If restriction is `owl:someValuesFrom`:
 $C = \{ x \mid (x P y) \text{ for some } y \in R \}$
- If restriction is `owl:allValuesFrom`:
 $C = \{ x \mid \text{for all } y \text{ s.t. } (x P y), y \in R \}$
- If restriction is `owl:hasValue`:
 $C = \{ x \mid (x P V) \}$

90

90

Example: From Undergrad Advisees to Undergrad Advisor

- An undergrad advisor is a restriction class for which:
 - Students are advised
 - at least one advised student is undergraduate

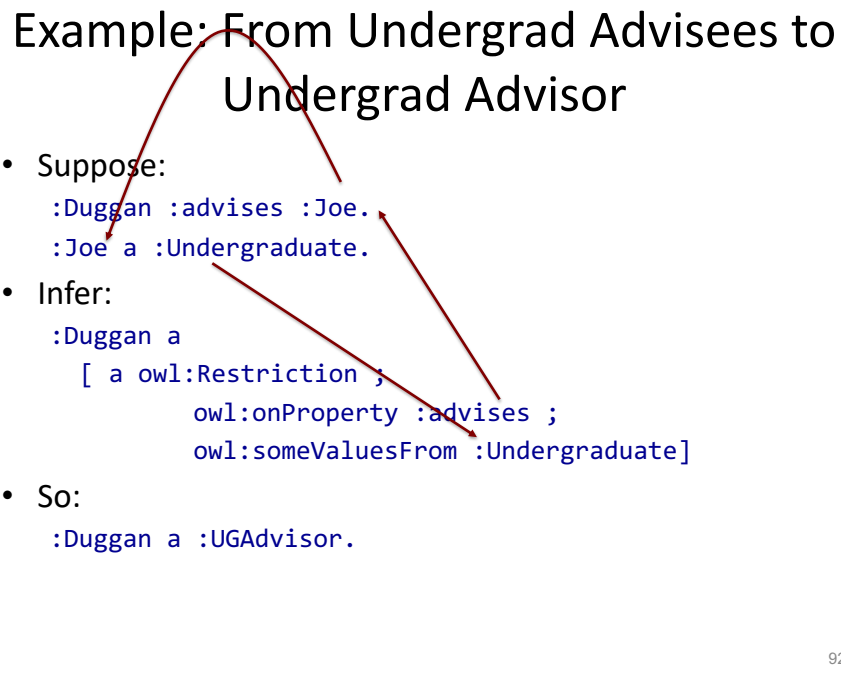
```
:UGAdvisor owl:equivalentClass  
[ a owl:Restriction ;  
  owl:onProperty :advises ;  
  owl:someValuesFrom :Undergraduate ].
```

Anonymous
class

91

91

Example: From Undergrad Advisees to Undergrad Advisor

- Suppose:
`:Duggan :advises :Joe.`
`:Joe a :Undergraduate.`
 - Infer:
`:Duggan a`
 `[a owl:Restriction ;`
 `owl:onProperty :advises ;`
 `owl:someValuesFrom :Undergraduate]`
 - So:
`:Duggan a :UGAdvisor.`
- 

92

92

Example: From Research Supervisor to Research Student

- Class of faculty members supervising **only** research students
- ```
[a owl:Restriction;
 owl:onProperty :supervises;
 owl:allValuesFrom :ResearchStudent]
```

93

93

## Example: From Research Supervisor to Research Student

- Research faculty **only** supervise research students:

```
:ResearchFaculty rdfs:subClassOf
 [a owl:Restriction;
 owl:onProperty :supervises;
 owl:allValuesFrom :ResearchStudent]
```

94

94

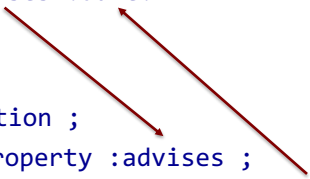
## Example: From Research Supervisor to Research Student

- Suppose:

```
:ProfBigShot a :ResearchFaculty
:ProfBigShot :advises :Jane.
```

- Infer:

```
:ProfBigShot a
 [a owl:Restriction ;
 owl:onProperty :advises ;
 owl:allValuesFrom :ResearchStudent]
```



- So:

```
:Jane a :ResearchStudent.
```

95

95

## Example: Priority Items

- Define priority levels:  
`q:PriorityLevel a owl:Class .`  
`q:High a q:PriorityLevel .`
- Property for priority level:  
`q:hasPriority rdfs:range q:PriorityLevel .`
- Define class of high-priority items:  
`q:HighPriorityItem owl:equivalentClass`  
`[ a owl:Restriction;`  
`owl:onProperty q:hasPriority;`  
`owl:hasValue q:High ] .`

96

96

## Exar

- We can filter data based on values that denote type tags

Table 7-1 Sample Tabular Data for Triples

| ID | Model Number | Division              | Product       |                      |        |           |
|----|--------------|-----------------------|---------------|----------------------|--------|-----------|
|    |              |                       | Product Line  | Manufacture Location | SKU    | Available |
| 1  | ZX-3         | Manufacturing Support | Paper Machine | Sacramento           | FB3524 | 23        |
| 2  | ZX-3P        | Manufacturing Support | Paper Machine | Sacramento           | KD5243 | 4         |
| 3  | ZX-3S        | Manufacturing Support | Paper Machine | Sacramento           | IL4028 | 34        |
| 4  | B-1430       | Control Engineering   | Feedback Line | Elizabeth            | KS4520 | 23        |
| 5  | B-1430K      | Control Engineering   | Feedback Line | Elizabeth            | CL5934 | 14        |
| 6  | B-1431       | Control Engineering   | Active Sensor | Seoul                | KK3945 | 0         |
| 7  | DBB-12       | Accessories           | Monitor       | Hong Kong            | ND5520 | 100       |
| 8  | SP-1234      | Safety                | Safety Valve  | Cleveland            | HI4554 | 4         |
| 9  | SPX-1234     | Safety                | Safety Valve  | Cleveland            | OP5333 | 14        |

`ns:Paper_Machine rdfs:type owl:Class .`

`ns:Paper_Machine owl:equivalentClass`  
`[ a owl:Restriction;`  
`owl:onProperty mfg:Product_Product_Line;`  
`owl:hasValue "Paper machine" ] .`

So e.g. `mfg:Product1 rdfs:type ns:Paper_Machine`

97

97

## Set Intersection in OWL

- Example: Undergraduate research students

```
:UndergradResearchStudent owl:equivalentClass
[a owl:Class;
 owl:intersectionOf
 (:Undergraduate :ResearchStudent)] .
```

98

98

## Set Union in OWL

- Example: Researchers are either students or faculty

```
:Researcher owl:equivalentClass
[a owl:Class;
 owl:unionOf
 (:ResearchFaculty
 :ResearchStudent)] .
```

99

99

## Enumerating Sets

- Set membership assumed open
- `owl:oneOf` allows a complete enumeration:

```
ss:SolarPlanet
 rdf:type owl:Class;
 owl:oneOf (ss:Mercury ss:Venus ss:Earth
 ss:Mars ss:Jupiter ss:Saturn
 ss:Uranus ss:Neptune) .
```

- Note: This does not state that the planets are distinct!

100

100

## Differentiating Individuals

- Non-unique naming assumption
- Instances must be explicitly distinguished:

```
ss:Earth owl:differentFrom ss:Mars .
```

- To distinguish large numbers of individuals:

```
[a owl:AllDifferent;
 owl:distinctMembers (
 ss:Mercury ss:Venus ss:Earth ss:Mars
 ss:Jupiter ss:Saturn ss:Uranus ss:Neptune)] .
```

101

101



## Example

- James Dean made 3 movies  
`:JamesDeanMovie a owl:Class;  
owl:oneOf ( :Giant :EastOfEden :Rebel )`
- We can infer e.g.  
`:Giant rdf:type :JamesDeanMovie`
- If John has a movie he likes such that  
`:JohnsMovie rdf:type :JamesDeanMovie`
- then it must be one of the above three
- If we learn:  
`:JohnsMovie owl:differentFrom :Giant  
:JohnsMovie owl:differentFrom :EastOfEden`
- then we can infer:  
`:JohnsMovie owl:sameAs :Rebel`

102

102

## Set Complement

- We have to be careful:  
`bb:MinorLeaguePlayer owl:complementOf  
bb:MajorLeaguePlayer .`
- Here is the fix:  
`bb:MinorLeaguePlayer owl:intersectionOf  
( [ a owl:Class;  
owl:complementOf bb:MajorLeaguePlayer  
]  
bb:Player ).`

103

103

## Conclusions

- Semantic Web
  - RDF: Knowledge Representation
  - RDF Schema: Classification
  - OWL: Ontologies
- From database to knowledge base
- Application: Organizing enterprise knowledge
- Application: B2B knowledge sharing

104