Ontology Languages: Description Logics

(exercise solutions)

Oscar Romero

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya

Further Examples of DL Constructs

□ What is the meaning of these axioms?

- Disjunction: $\forall hasChild.(Doctor \sqcup Lawyer)$

- Qualified existential restriction: $\exists hasChild.Doctor$

- Full negation: $\neg(Doctor \sqcup Lawyer)$

- Number restrictions: $(\geq 2\, hasChild) \sqcap (\leq 1\, sibling)$

- Qualified number restrictions: $(\geq 2\, hasChild.\, Doctor)$

- Inverse role: $\forall hasChild^{-}.Doctor$

- Reflexive-transitive role closure: $\exists hasChild^{*}.Doctor$

Exercise

□ Represent as concept expressions the following UML diagram
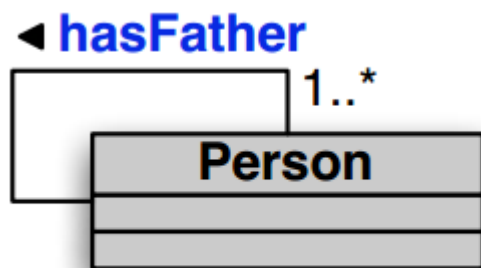
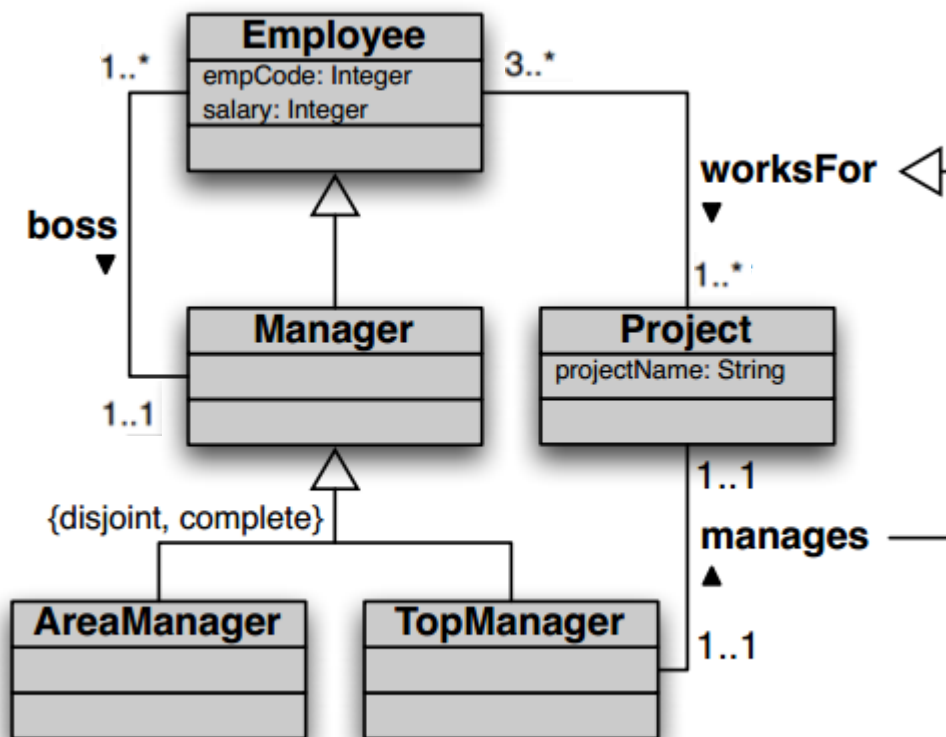$$\text{TBox } \mathcal{T}: \quad \exists \text{hasFather} \quad \sqsubseteq \quad \text{Person}$$
$$\exists \text{hasFather}^- \quad \sqsubseteq \quad \text{Person}$$
$$\text{Person} \quad \sqsubseteq \quad \exists \text{hasFather}$$

Exercise II

□ Represent as concept expressions the following UML diagram

$$Manager \sqsubseteq Employee$$
$$AreaManager \sqsubseteq Manager$$
$$TopManager \sqsubseteq Manager$$
$$Manager \sqsubseteq AreaManager \sqcup TopManager$$
$$AreaManager \sqsubseteq \neg TopManager$$
$$Employee \sqsubseteq \exists salary$$
$$\exists salary^- \sqsubseteq Integer$$
$$\exists worksFor \sqsubseteq Employee$$
$$\exists worksFor^- \sqsubseteq Project$$
$$Employee \sqsubseteq \exists worksFor$$
$$Project \sqsubseteq (\geq 3\, worksFor^-)$$
$$\cdots$$

Exercise

- TBOX:

$$Researcher \sqsubseteq \neg \, Professor$$
$$Researcher \sqsubseteq \neg \, Lecturer$$
$$\exists \, TeachesTo^- \sqsubseteq Student$$

$$Student \sqcap \neg \, Undergrad \sqsubseteq GraduateStudent$$
$$\exists \, TeachesTo.Undergrad \sqsubseteq Professor \sqcup Lecturer$$

TBOX Inferences: $Researcher \sqsubseteq \forall \, TeachesTo.GraduateStudent$

- ABOX:

$$TeachesTo(dupond, pierre)$$
$$\neg \, GraduateStudent(pierre)$$
$$\neg \, Professor(dupond)$$

ABOX Inferences: $Lecturer(dupond)$

Complex Constraints (II)

- ☐ Cardinalities on roles:

```
_:a  rdfs:subClassOf     owl:Restriction
_:a  owl:onProperty      RegisteredTo
_:a  owl:minCardinality  3
_:b  rdfs:subClassOf     owl:Restriction
_:b  owl:onProperty      RegisteredTo
_:b  owl:maxCardinality  6
:Student  rdfs:subClassOf  _:a
:Student  rdfs:subClassOf  _:b
```

- ☐ How would you express on OWL the following constraint?

$$C_1 \sqsubseteq \forall P.C$$

```
_:a  rdfs:subClassOf     owl:Restriction
_:a  owl:onProperty      P
_:a  owl:someValuesFrom  C

:C_1 rdfs:subClassOf _:a
```